# AT&P JOURNAL plus

2

2009

## *Process Control*

## Riadenie procesov

*reviewed slovak professional magazine for scientific and engineering issues*

recenzované periodikum vedeckých a inžinierskych publikácií

ISSN 1336-5010

# Riadenie procesov

# *Process Control*

## Odborný garant

**prof. Ing. Miroslav Fikar, DrSc.**
Slovenská technická univerzita v Bratislave
Fakulta chemickej a potravinárskej technológie
Ústav informatizácie, automatizácie a matematiky
Oddelenie informatizácie a riadenia procesov
Radlinského 9, 812 37 Bratislava, Slovensko
Tel.: +421 2 59 32 53 66, +421 2 52 49 52 69
e-mail: miroslav.fikar@stuba.sk

## Technical guarantee

**prof. Ing. Miroslav Fikar, DrSc.**
Slovak University of Technology in Bratislava
Faculty of Chemical and Food Technology
Institute of Information Engineering,
Automation, and Mathematics
Radlinského 9, 812 37 Bratislava, Slovak Republic
Tel.: +421 2 59 32 53 66, +421 2 52 49 52 69
e-mail: miroslav.fikar@stuba.sk

## Vydavateľ Publisher

**HMH s.r.o.**
Tavarikova osada 39
841 02 Bratislava 42
IČO: 31356273

## Spoluzakladateľ Co-founder

**Katedra ASR, EF STU**
**Katedra automatizácie a regulácie, EF STU**
**Katedra automatizácie, ChtF STU**
**PPA CONTROLL, a.s.**

**www.atpjournal.sk**

## Redakčná rada Draft committee

## Redakcia Editors office

**AT&P journal**
Evidenčné číslo: EV 3242/09
Košická 37, 821 09 Bratislava 2
tel.: 02/5026 1752 – 55
fax: 02/5026 1757
e-mail: info@atpjournal.sk
http://www.atpjournal.sk

Ing. Anton Gérer
šéfredaktor – editor in chief
sefredaktor@atpjournal.sk

Ing. Ildikó Csölleová
vedúca redakcie – editorial office manager
podklady@atpjournal.sk

Mgr. Zuzana Švecová
marketingová manažérka – marketing manager
marketing@atpjournal.sk

Ing. Branislav Bložon
odborný redaktor – editor
redaktor@atpjournal.sk

Ing. Martin Karbovanec
odborný redaktor – editor
karbovanec@atpjournal.sk

Zuzana Pettingerová
technická redaktorka – DTP
podklady@atpjournal.sk

Mgr. Bronislava Chocholová
jazyková redaktorka – text corrector

Datum vydania: október 2009

ISSN 1336-5010

# Obsah

## Lineárne a nelineárne riadenie

## Prediktívne riadenie

## Procesná optimalizácia

## Robustné a adaptívne riadenie

## Inteligentné riadiace systémy

# Articles

## Linear and Non-Linear Control System Design

## Model Predictive Control

## Process Optimisation

## Robust and Adaptive Control

## Intelligent Control Systems

# Anisotropic balanced truncation — application to reduced-order controller design

Michael Tchaikovsky

**Abstract**

This paper addresses the problem of reduced-order normalized anisotropic optimal controller design by anisotropic balanced truncation. This controller is the solution to the normalized anisotropy-based stochastic $\mathcal{H}_\infty$ problem. Anisotropic balanced truncation is aimed at reducing the order of closed-loop system. Two respective Riccati equations involved are used to define a set of closed-loop input-output invariants of closed-loop system called anisotropic characteristic values. The part of controller corresponding to smaller anisotropic characteristic values is truncated to give a reduced-order one. Truncation is carried out for the closed-loop state-space realization in anisotropic balanced coordinates, when the product of two respective solutions of Riccati equations is a diagonal matrix with the squares of anisotropic characteristic values situated in descending order on its main diagonal. In anisotropic balanced coordinates, small characteristic values correspond to states which are easy to filter and control in a sense of anisotropic norm. It is shown that the reduced-order anisotropic controller is the full-order one for the reduced-order plant. An example of application to flight control in a windshear is given.

**Keywords:** stochastic norm, information, order reduction

## Introduction

The stochastic approach to $\mathcal{H}_\infty$ optimization introduced in [1], [2] is based on using the anisotropic norm of a system as performance criterion. The anisotropic norm being a special case of stochastic norm is a quantitative index of system sensitivity to random input disturbances with mean anisotropy bounded by known nonnegative parameter. In turn, the mean anisotropy of a vector random sequence produced by a stable shaping filter from vector zero-mean Gaussian white noise with scalar covariance matrix is a measure of colouredness of this sequence, that is a measure of correlation of vector components of the sequence (spatial part of the mean anisotropy), as well as a measure of correlation of different elements of this sequence (temporal part of the mean anisotropy). The latter coincides with the mutual information about an element of the sequence contained in the past history of this sequence. It has been shown that $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms of a linear discrete time-invariant system are two limiting cases of the anisotropic norm as the mean anisotropy level of input random disturbance tends to zero or infinity, respectively. Therefore, this approach combines the attractive features of robust control and information theories holding an intermediate position between $\mathcal{H}_2$ /LQG and $\mathcal{H}_\infty$ problems. Given a standard plant model and input mean anisotropy level, the anisotropy-based stochastic $\mathcal{H}_\infty$ problem consists in finding an output-feedback dynamic controller to internally stabilize the closed-loop system and minimize its anisotropic norm. The solution to this problem presented in [2] yields to the full-order controller and results in solving a cross-coupled nonlinear algebraic equation system defining the controller state-space realization matrices. However, we are interested in obtaining a reduced-order anisotropic controller.

The approximative approach to model reduction according to minimum anisotropic norm performance was introduced in [3]. A reduced-order model obtained by this method approximates the behaviour of a full-order system in steady-state mode, but it does not reflect the full-order system dynamics, since does not take into account pole locations of full- and reduced-order systems at all. Besides that, this method is intended for an open-loop system, therefore it accounts for neither controller properties nor even controller presence. This paper addresses the problem of reduced-order normalized anisotropic optimal controller design by means of anisotropic balanced truncation, which is close to LQG and $\mathcal{H}_\infty$ balanced truncation techniques developed in [4], [5], correspondingly, and aimed at reducing the order of a closed-loop system. Two respective Riccati equations involved are used to define a set of closed-loop input-output system invariants called the *anisotropic characteristic values*. The part of the plant or controller corresponding to smaller anisotropic characteristic values is truncated to give a reduced-order plant or controller. Truncation is carried out for the closed-loop realization in anisotropic balanced coordinates, when the product of two respective solutions of the Riccati equations is a diagonal matrix with the squares of anisotropic characteristic values situated in descending order on its main diagonal. In anisotropic balanced coordinates, small characteristic values correspond to states which are easy to filter and control in a sense of anisotropic norm. It will be shown that the reduced-order controller is the full-order one for the reduced-order plant.

The paper structure is as follows. In Section 1 we consider the normalized anisotropy-based $\mathcal{H}_\infty$ problem. Subsection 1.1 introduces the problem statement together with some necessary background. The state-space structure of the worst-case input shaping filter together with a sufficient saddle-point type condition for optimality of a controller in the problem are given in Subsection 1.2. The algebraic equation system defining the state-estimating optimal controller is introduced in Subsection 1.3. The technique of controller order reduction by anisotropic balanced truncation is considered in Section 2. The notion of anisotropic characteristic values is introduced in Subsection 2.1 together with a nonsingular similarity transformation putting the system realization into the anisotropic balanced coordinates. Subsection 2.2 represents the expressions for state-space realizations of reduced-order plant and anisotropic controller. Section 3 is devoted to an example of reduced-order anisotropic controller design for longitudinal flight control in a windshear.

## 1. Normalized anisotropy-based stochastic $\mathcal{H}_\infty$ optimization problem

The normalized anisotropy-based stochastic $\mathcal{H}_\infty$ problem is characterized by some features making it different from the general-case problem considered in [2]. To disclose these important distinctions, it is preferable to consider in details the statement and solution of the normalized problem.

### 1.1 Problem statement

All the encountered random elements are assumed to be defined on a complete probability space $(\mathbf{\Omega}, \mathcal{F}, \mathbf{P})$ with the set $\mathbf{\Omega}$ of primary outcomes, the $\sigma$-algebra $\mathcal{F}$ of random events, and the probability measure $\mathbf{P}$ with the corresponding expectation functional $\mathbf{E}$.

Consider a linear discrete time-invariant causal plant $P(z)$ with $n$-dimensional internal state $X$, $m_1$-dimensional disturbance input $W$, $m_2$-dimensional control input $U$, $p_1$-dimensional controlled output $Z$, and $p_2$-dimensional measured output $Y$. All these signals are double-sided discrete-time sequences related to each other by the equations

$$\begin{bmatrix} x_{k+1} \\ z_k \\ y_k \end{bmatrix} = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right] \begin{bmatrix} x_k \\ w_k \\ u_k \end{bmatrix}, \tag{1}$$

where all matrices have appropriate dimensions, $p_1 = m_1 = p_2 + m_2$, and the matrices

$$B_1 = \begin{bmatrix} B_2 & 0 \end{bmatrix}, D_{21} = \begin{bmatrix} 0 & I_{p_2} \end{bmatrix}, C_1 = \begin{bmatrix} C_2 \\ 0 \end{bmatrix}, D_{12} = \begin{bmatrix} 0 \\ I_{m_2} \end{bmatrix}. \tag{2}$$

The state-space realization $(A, B_2, C_2)$ is assumed to be minimal (i.e. $(A, B_2)$ is controllable, $(A, C_2)$ is observable). Plant (1) is called the normalized standard plant. It is easily seen that the normalized standard plant has the controlled output

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = \begin{bmatrix} C_2 X \\ U \end{bmatrix} \tag{3}$$

and the disturbance input

$$W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} \tag{4}$$

partitioned into $m_2$-dimensional block $W_1$ and $p_2$-dimensional block $W_2$ such that $W_1$ enters the system together with the control signal (actuator noise) while $W_2$ affects upon the measured output $Y$ (measurement noise).

The only prior information on the probability distribution of the random external disturbance $W$ consists in the following: $W$ is a stationary Gaussian sequence with mean anisotropy bounded by a known nonnegative parameter $\alpha$. Specifically, the latter means that $W$ is produced from $m_1$-dimensional Gaussian white noise $V$ with zero mean and identity covariance matrix: $\mathbf{E}(v_k) = 0$, $\mathbf{E}(v_k v_k^{\mathrm{T}}) = I_{m_1}$, $-\infty < k < +\infty$, by an unknown shaping filter $G$ in the family

$$\mathcal{G}_\alpha \triangleq \left\{ G \in \mathcal{H}_2^{m_1 \times m_1} : \overline{\mathbf{A}}(G) \leqslant \alpha \right\},$$

where

$$\overline{\mathbf{A}}(G) = -\frac{1}{4\pi} \int_{-\pi}^{\pi} \ln\det \left\{ \frac{m_1}{\|G\|_2^2} \widehat{G}(\omega) \widehat{G}^*(\omega) \right\} d\omega$$

is the mean anisotropy functional introduced in [1] (also called the mean anisotropy), the angular boundary value

$$\widehat{G}(\omega) \triangleq \lim_{r \to 1-0} G(re^{i\omega}),$$

and

$$\|G\|_2 \doteq \left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathrm{tr} \left\{ \widehat{G}(\omega) \widehat{G}^*(\omega) \right\} d\omega \right\}^{1/2}.$$

At that, no assumption is made in respect of cross-correlation of blocks $W_1$ and $W_2$.

The normalized anisotropy-based stochastic $\mathcal{H}_\infty$ optimization problem is formulated as follows.

*Problem 1*. Given normalized standard plant (1) and input mean anisotropy level $\alpha \geqslant 0$, find a strictly causal controller $K$ to internally stabilize the closed-loop system $F(z)$ given by the lower linear-fractional transformation of the pair $(P, K)$:

$$F(z) = \mathcal{F}_l(P, K) = P_{11} + P_{12} K (I_{p_2} - P_{22} K)^{-1} P_{21}, \tag{5}$$

where

$$P_{ij}(z) \sim \left[\begin{array}{c|c} A & B_j \\ \hline C_i & D_{ij} \end{array}\right], \; i, j = \overline{1,2}, \tag{6}$$

and minimize its $\alpha$-anisotropic norm:

$$|\!|\!| F |\!|\!|_\alpha \doteq \sup_{G \in \mathcal{G}_\alpha} \frac{\|FG\|_2}{\|G\|_2} \to \inf_K, \; K \in \mathcal{K}. \tag{7}$$

The formulated problem (just as any of minimax problems) can be considered as an antagonistic game of two opponents, control system designer and nature. The set of designer's strategies in this game is the set $\mathcal{K}$ of internally stabilizing controllers, and the set of nature's strategies is the family $\mathcal{G}_\alpha$ of filters generating random sequences with mean anisotropy bounded by known parameter $\alpha \geqslant 0$.

Denote that in the case of zero mean anisotropy level $\alpha = 0$ the formulated above problem coincides with the normalized LQG problem considered in [4], [5].

### 1.2 Worst-case shaping filter for closed-loop system

Since problem (7) is a minimax problem, one can use the results of differential game theory to formulate a saddle-point type condition of optimality. For any shaping filter $G \in \mathcal{G}_\alpha$ and any internally stabilizing controller $K \in \mathcal{K}$, let us introduce the following sets

$$\mathcal{K}^\lozenge(G) \triangleq \operatorname*{Arg\,min}_{K \in \mathcal{K}} \|FG\|_2, \qquad G \in \mathcal{G}_\alpha, \tag{8}$$

$$\mathcal{G}_\alpha^\lozenge(K) \triangleq \operatorname*{Arg\,max}_{G \in \mathcal{G}_\alpha} \frac{\|FG\|_2}{\|G\|_2}, \qquad K \in \mathcal{K}. \tag{9}$$

These sets are assumed to be nonempty. Set (5) consists of the controllers being solutions to the weighted LQG problem under the assumption that the closed-loop system input is fed with coloured noise $W = GV$. Any such controller $K \in \mathcal{K}^\lozenge(G)$ minimizes variance of the output random sequence $Z$ (LQG-cost)

$$J_{\mathrm{LQG}}(FG) \triangleq \mathbf{E}(z_k^{\mathrm{T}} z_k) = \mathbf{E}\left(\begin{bmatrix} x_k \\ u_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} C_2^{\mathrm{T}} C_2 & 0 \\ 0 & I_{m_2} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}\right) \tag{10}$$

of weighted closed-loop system $FG$ with input disturbance $V$.

In turn, set (9) is formed by the filters generating Gaussian random sequences $W$ with spectral densities

$$\widehat{S}_{WW}(\omega) = \widehat{G}(\omega)\widehat{G}^*(\omega), \qquad \omega \in [-\pi, \pi],$$

which are the worst (i.e. the most adverse) for the closed-loop system $F(z) = \mathcal{F}_l(P,K)$. Although the set $\mathcal{G}_\alpha^\lozenge(K)$ is invariant under right-hand multiplication by an all-pass system, and hence, consists of infinite number of filters, all of them generate the sequences with the unique up to scalar multiplier worst-case spectral density [1]. Such filters are called the worst-case input shaping filters.

Thus the relation

$$(\mathcal{K}^\lozenge \circ \mathcal{G}_\alpha^\lozenge)(K) \triangleq \bigcup_{G \in \mathcal{G}_\alpha^\lozenge(K)} \mathcal{K}^\lozenge(G), \qquad K \in \mathcal{K}$$

defines the (generally set-valued) composition $\mathcal{K}^\lozenge \circ \mathcal{G}_\alpha^\lozenge : \mathcal{K} \to 2^{\mathcal{K}}$ of the mappings $\mathcal{K}^\lozenge : \mathcal{G}_\alpha^\lozenge \to 2^{\mathcal{K}}$ and $\mathcal{G}_\alpha^\lozenge : \mathcal{K}^\lozenge \to 2^{\mathcal{G}_\alpha^\lozenge}$. The following lemma that can be proved similar to Lemma 1 in [2] establishes a sufficient saddle-point type condition of optimality for problem (7).

*Lemma 1*. If a controller $K$ is a stationary point of the mapping $\mathcal{K}^\lozenge \circ \mathcal{G}_\alpha^\lozenge$, that is, if there exists a shaping filter $G$ such that

$$K \in \mathcal{K}^\lozenge(G), \qquad G \in \mathcal{G}_\alpha^\lozenge(K),$$

then the controller $K$ is a solution to problem (7).

Let $K \in \mathcal{K}$ be an admissible controller with $n$-dimensional internal state $H$ related with the measurement $Y$ and control sequence $U$ by the equations

$$\begin{bmatrix} h_{k+1} \\ \hline u_k \end{bmatrix} = \left[\begin{array}{c|c} A_{\mathrm{c}} & B_{\mathrm{c}} \\ \hline C_{\mathrm{c}} & 0 \end{array}\right] \begin{bmatrix} h_k \\ \hline y_k \end{bmatrix}, \tag{11}$$

where $A_{\mathrm{c}}$, $B_{\mathrm{c}}$, $C_{\mathrm{c}}$ are constant matrices of appropriate dimensions. Then the state-space realization of closed-loop system (5) is given by

$$F(z) \sim \left[\begin{array}{c|c} A_{\mathrm{cl}} & B_{\mathrm{cl}} \\ \hline C_{\mathrm{cl}} & 0 \end{array}\right] = \left[\begin{array}{cc|c} A & B_2 C_{\mathrm{c}} & B_1 \\ B_{\mathrm{c}} C_2 & A_{\mathrm{c}} & B_{\mathrm{c}} D_{21} \\ \hline C_1 & D_{12} C_{\mathrm{c}} & 0 \end{array}\right] \tag{12}$$

with $A_{\mathrm{cl}}$ stable, that is, taking into account (2)–(4), input $W$ and output $Z$ of the closed-loop system $F$ are related by the equations

$$\begin{bmatrix} x_{k+1} \\ h_{k+1} \\ \hline z_{1k} \\ z_{2k} \end{bmatrix} = \left[\begin{array}{cc|cc} A & B_2 C_{\mathrm{c}} & B_2 & 0 \\ B_{\mathrm{c}} C_2 & A_{\mathrm{c}} & 0 & B_c \\ \hline C_2 & 0 & 0 & 0 \\ 0 & C_{\mathrm{c}} & 0 & 0 \end{array}\right] \begin{bmatrix} x_k \\ h_k \\ \hline w_{1k} \\ w_{2k} \end{bmatrix}.$$

It is assumed that the system $F(z)$ satisfies strict inequality

$$\frac{1}{\sqrt{m_1}} \|F\|_2 < \|F\|_\infty, \tag{13}$$

otherwise, its anisotropic norm coincides trivially with the scaled $\mathcal{H}_2$ norm. It should be noted that inequality (13) is violated iff the closed-loop system $F$ is inner up to a nonzero constant factor, in which case there exists a number $\lambda > 0$ such that $\widehat{F}^*(\omega)\widehat{F}(\omega) = \lambda I_{m1}$ for almost all $\omega \in [-\pi, \pi]$ [1].

*Lemma 2*. Let the realization $(A, B_2, C_2)$ of plant (1) be minimal, and let the closed-loop system $F$ not be inner. Then, for any controller $K \in \mathcal{K}$ and given level of input mean anisotropy $\lambda \geqslant 0$, there exists a unique pair $(q, R)$ of the scalar parameter $q \in [0, \|F\|_\infty^{-2})$ and stabilizing solution $R = R^{\mathrm{T}} \geqslant 0$ of the algebraic Riccati equation

$$\left.\begin{aligned} & R = A_{\mathrm{cl}}^{\mathrm{T}} R A_{\mathrm{cl}} + q C_{\mathrm{cl}}^{\mathrm{T}} C_{\mathrm{cl}} + L^{\mathrm{T}} \Sigma^{-1} L \\ & \Sigma \triangleq (I_{m_1} - B_{\mathrm{cl}}^{\mathrm{T}} R B_{\mathrm{cl}})^{-1} \\ & L = \begin{bmatrix} L_1 & L_2 \end{bmatrix} \triangleq \Sigma B_{\mathrm{cl}}^{\mathrm{T}} R A_{\mathrm{cl}} \end{aligned}\right\} \tag{14}$$

such that

$$-\frac{1}{2} \ln\det\left\{\frac{m_1 \Sigma}{\operatorname{tr}(L P_{\mathrm{c}} L^{\mathrm{T}} + \Sigma)}\right\} = \alpha, \tag{15}$$

where $P_{\mathrm{c}} = P_{\mathrm{c}}^{\mathrm{T}} > 0$ is the controllability gramian of the shaping filter

$$G(z) \sim \left[\begin{array}{c|c} A_{\mathrm{cl}} + B_{\mathrm{cl}} L & B_{\mathrm{cl}} \Sigma^{1/2} \\ \hline L & \Sigma^{1/2} \end{array}\right] \tag{16}$$

satisfying the Lyapunov equation

$$P_{\mathrm{c}} = (A_{\mathrm{cl}} + B_{\mathrm{cl}} L) P_{\mathrm{c}} (A_{\mathrm{cl}} + B_{\mathrm{cl}} L)^{\mathrm{T}} + B_{\mathrm{cl}} \Sigma B_{\mathrm{cl}}^{\mathrm{T}}. \tag{17}$$

At that, filter (16) is a representative of set (9) of the worst-case input shaping filters satisfying factorization

$$\widehat{G}(\omega)\widehat{G}^*(\omega) = (I_{m_1} - q\widehat{F}^*(\omega)\widehat{F}(\omega))^{-1}.$$

Proof of this lemma follows immediately from Theorem 2 in [1] applied to closed-loop system (12).

*Remark 1*. Recall that a solution $R = R^{\mathrm{T}} \in \mathbb{R}^{2n \times 2n}$ of algebraic Riccati equation (14) is called stabilizing one if the matrix $A_{\mathrm{cl}} + B_{\mathrm{cl}} L$ is stable and $\Sigma > 0$. For any controller $K \in \mathcal{K}$ and $q \in [0, \|F\|_\infty^{-2})$ equation (14) has a unique stabilizing solution and this solution is a positive semidefinite matrix [1].

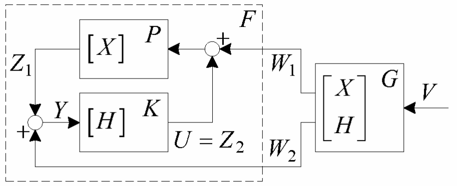


**Fig.1 Block diagram of weighted closed-loop system $FG$**

*Remark 2*. The internal state of the worst-case shaping filter $G$ actually is a copy of the internal state of the closed-loop system $F$ (see block diagram at Fig. 1). Thus, equations (1) and (11) combined with

$$w_k = L_1 x_k + L_2 h_k + \Sigma^{1/2} v_k$$

relate the input $V$, output $W = GV$, and internal state $(X,H)$ of worst-case shaping filter (16). Taking into account partitioning (4) of filter output $W$, one can put down the following equations

$$\begin{bmatrix} x_{k+1} \\ h_{k+1} \\ \hline w_{1k} \\ w_{2k} \end{bmatrix} = \left[\begin{array}{cc|c} A + B_2 L_{11} & B_2(C_\mathrm{c} + L_{12}) & B_2 \tilde{\Sigma}_1 \\ B_\mathrm{c}(C_2 + L_{21}) & A_\mathrm{c} + B_\mathrm{c} L_{22} & B_\mathrm{c} \tilde{\Sigma}_2 \\ \hline L_{11} & L_{12} & \tilde{\Sigma}_1 \\ L_{21} & L_{22} & \tilde{\Sigma}_2 \end{array}\right] \begin{bmatrix} x_k \\ h_k \\ \hline v_k \end{bmatrix}$$

defining the dynamics of the worst-case shaping filter $G(z)$, where

$$\begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} = \begin{bmatrix} L_1 & L_2 \end{bmatrix} = L, \qquad (18)$$

$$\begin{bmatrix} \tilde{\Sigma}_1 \\ \tilde{\Sigma}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\Sigma}_{11} & \tilde{\Sigma}_{12} \\ \tilde{\Sigma}_{21} & \tilde{\Sigma}_{22} \end{bmatrix} = \Sigma^{1/2}. \qquad (19)$$

For weighted closed-loop system $FG$ we have

$$\begin{bmatrix} x_{k+1} \\ h_{k+1} \\ \hline z_{1k} \\ z_{2k} \end{bmatrix} = \left[\begin{array}{cc|c} A + B_2 L_{11} & B_2(C_\mathrm{c} + L_{12}) & B_2 \tilde{\Sigma}_1 \\ B_\mathrm{c}(C_2 + L_{21}) & A_\mathrm{c} + B_\mathrm{c} L_{22} & B_\mathrm{c} \tilde{\Sigma}_2 \\ \hline C_2 & 0 & 0 \\ 0 & C_\mathrm{c} & 0 \end{array}\right] \begin{bmatrix} x_k \\ h_k \\ \hline v_k \end{bmatrix}. \qquad (20)$$

### 1.3 Optimal state-estimating controller for weighted LQG problem

Let us fix the worst-case input shaping filter $G^\Diamond \in \mathcal{G}_\alpha^\Diamond$ defined by Lemma 2 and consider the weighted plant

$$P_G \triangleq \begin{bmatrix} P_{11} G^\Diamond & P_{12} \\ P_{21} G^\Diamond & P_{22} \end{bmatrix}$$

$$\sim \left[\begin{array}{ccc|cc} A & B_2 L_{11} & B_2 L_{12} & B_2 \tilde{\Sigma}_1 & B_2 \\ 0 & A + B_2 L_{11} & B_2(C_c + L_{12}) & B_2 \tilde{\Sigma}_1 & 0 \\ 0 & B_c(C_2 + L_{21}) & A_c + B_c L_{22} & B_2 \tilde{\Sigma}_2 & 0 \\ \hline C_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{m_2} \\ C_2 & L_{21} & L_{22} & \tilde{\Sigma}_2 & 0 \end{array}\right], \qquad (21)$$

where $P_{ij}$ are defined by (6), with two inputs $V$ (Gaussian white noise) and $U$, two outputs $Z = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}$, $Y$, and $3n$-dimensional internal state $(X, X^0, H^0)$, where $(X^0, H^0)$ is the internal state of the worst-case shaping filter $G^\Diamond$ (see Fig. 2).

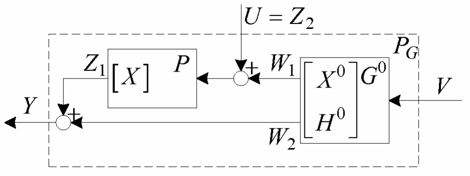


**Fig.2 Block diagram of plant $P_G$ for weighted LQG problem**

With the fixed worst-case input shaping filter, Problem 1 is equivalent to the following weighted LQG problem for plant (21).

*Problem 2*. Given the weighted plant $P_G$, find a strictly causal controller $K$ to internally stabilize closed-loop system $F_G = \mathcal{F}_l(P_G, K)$ and minimize LQG cost (10):

$$J_{\mathrm{LQG}}(F_G) \to \inf_{K \in \mathcal{K}}. \qquad (22)$$

Since in general the internal state $X$ of plant (1) is not measurable, and the measurement $Y$ includes additive noise $W_2$, the desired controller $K$ can be only the state-estimating output-feedback controller with the internal state $H$ being the optimal mean-square estimate of the internal state $X$ of plant (1).

Let $\mathcal{F}_k^Y$ denote the $\sigma$-algebra of random events induced by the history $(y_j)_{j \leqslant k}$ of the measurement signal $Y$ at instant $k$, and $(\mathcal{F}_k^Y)_{-\infty<k<+\infty}$ be the flow of $\sigma$-algebras in $\mathcal{F}$ generated by the sequence $Y$.

Recall that stabilizing controller (11) is called state-estimating one if its $n$-dimensional internal state $H$ coincides with the sequence of one-step predictors for the internal state $X$ of plant (1) by the measurement signal $Y$ under the worst-case input disturbance $W$, i.e. if

$$h_k = \mathbf{E}(x_k \mid \mathcal{F}_{k-1}^Y), \qquad -\infty < k < +\infty,$$

when $W = GV$ with the worst-case input shaping filter $G \in \mathcal{G}_\alpha^\Diamond(K)$ [2].

Let us consider the weighted closed-loop system $FG$ introduced by state-space equations (20) (see Fig. 2). In this system, the following relations between the flows of $\sigma$-algebras generated by the stationary Gaussian sequences are valid:

$$\mathcal{F}_k^H \subset \mathcal{F}_{k-1}^Y \subset \mathcal{F}_k^V \supset \mathcal{F}_k^X, \qquad -\infty < k < +\infty. \qquad (23)$$

Denote by

$$\begin{aligned} \hat{x}_k &\triangleq \mathbf{E}(x_k \mid \mathcal{F}_{k-1}^Y), \\ \hat{\hat{x}}_k &\triangleq \mathbf{E}(x_{k+1} \mid \mathcal{F}_{k-1}^Y), \\ \hat{y}_k &\triangleq \mathbf{E}(y_k \mid \mathcal{F}_{k-1}^Y) \end{aligned} \qquad (24)$$

the one-step and two-step predictors of the state $X$ by observation $Y$, as well as the one-step self-predictor of the measurement $Y$. Predictors (24) are $\mathcal{F}_{k-1}^{Y}$-measurable, and the following prediction errors correspond to them:

$$\tilde{x}_k \triangleq x_k - \hat{x}_k, \quad \tilde{\tilde{x}}_k \triangleq x_{k+1} - \hat{\hat{x}}_k, \quad \tilde{y}_k \triangleq y_k - \hat{y}_k. \tag{25}$$

As it was noted in [2], the sequence of measurement prediction errors $(\tilde{y}_k)_{-\infty<k<+\infty}$ is the martingale-difference [6] with respect to the flow $(\mathcal{F}_k^Y)_{-\infty<k<+\infty}$ and, hence, the zero-mean Gaussian white noise.

From equations (1), (2), (20) and inclusions (23), we have the following expressions for predictors (24) and prediction errors (25):

$$\left.\begin{aligned} \hat{\hat{x}}_k &= (A+B_2L_{11})\hat{x}_k + B_2(C_\mathrm{c}+L_{12})h_k \\ \hat{y}_k &= (C_2+L_{21})\hat{x}_k + L_{22}h_k \end{aligned}\right\}, \tag{26}$$

$$\left.\begin{aligned} \tilde{\tilde{x}}_k &= (A+B_2L_{11})\tilde{x}_k + B_2\tilde{\Sigma}_1 v_k \\ \tilde{y}_k &= (C_2+L_{21})\tilde{x}_k + \tilde{\Sigma}_2 v_k \end{aligned}\right\}. \tag{27}$$

By virtue of Normal Correlation Lemma [6], the predictor $\hat{x}_{k+1}$ is given by

$$\hat{x}_{k+1} = \hat{\hat{x}}_k + \mathbf{E}(\tilde{\tilde{x}}_k\tilde{y}_k^{\mathrm{T}})[\mathbf{E}(\tilde{y}_k\tilde{y}_k^{\mathrm{T}})]^{-1}\tilde{y}_k, \tag{28}$$

whereas the covariance matrix of prediction error $\tilde{x}_k$ is

$$\mathbf{E}(\tilde{x}_k\tilde{x}_k^{\mathrm{T}}) = \mathbf{E}(\tilde{\tilde{x}}_k\tilde{\tilde{x}}_k^{\mathrm{T}}) - \mathbf{E}(\tilde{\tilde{x}}_k\tilde{y}_k^{\mathrm{T}})[\mathbf{E}(\tilde{y}_k\tilde{y}_k^{\mathrm{T}})]^{-1}\mathbf{E}(\tilde{y}_k\tilde{\tilde{x}}_k^{\mathrm{T}}). \tag{29}$$

Denoting

$$S \triangleq \mathbf{E}(\tilde{x}_k\tilde{x}_k^{\mathrm{T}}), \tag{30}$$

let us express the covariance matrices in (29) from equation (27) as follows:

$$\begin{aligned} \mathbf{E}(\tilde{\tilde{x}}_k\tilde{\tilde{x}}_k^{\mathrm{T}}) &= (A+B_2L_{11})S(A+B_2L_{11})^{\mathrm{T}} + B_2\Sigma_{11}B_2^{\mathrm{T}}, \\ \mathbf{E}(\tilde{y}_k\tilde{y}_k^{\mathrm{T}}) &= (C_2+L_{21})S(C_2+L_{21})^{\mathrm{T}} + \Sigma_{22}, \\ \mathbf{E}(\tilde{\tilde{x}}_k\tilde{y}_k^{\mathrm{T}}) &= (A+B_2L_{11})S(C_2+L_{21})^{\mathrm{T}} + B_2\Sigma_{12}, \end{aligned} \tag{31}$$

where

$$\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} = \begin{bmatrix} \tilde{\Sigma}_1 \\ \tilde{\Sigma}_2 \end{bmatrix}\begin{bmatrix} \tilde{\Sigma}_1^{\mathrm{T}} & \tilde{\Sigma}_2^{\mathrm{T}} \end{bmatrix} = \Sigma. \tag{32}$$

Introducing the notation

$$\mathbf{E}(\tilde{y}_k\tilde{y}_k^{\mathrm{T}}) \triangleq \Theta, \qquad \mathbf{E}(\tilde{\tilde{x}}_k\tilde{y}_k^{\mathrm{T}}) \triangleq \Lambda\Theta, \tag{33}$$

from (29)–(31) we obtain the filtering algebraic Riccati equation

$$\left.\begin{aligned} &S = (A+B_2L_{11})S(A+B_2L_{11})^{\mathrm{T}} + B_2\Sigma_{11}B_2^{\mathrm{T}} - \Lambda\Theta\Lambda^{\mathrm{T}} \\ &\Theta \triangleq (C_2+L_{21})S(C_2+L_{21})^{\mathrm{T}} + \Sigma_{22} \\ &\Lambda \triangleq ((A+B_2L_{11})S(C_2+L_{21})^{\mathrm{T}} + B_2\Sigma_{12})\Theta^{-1} \end{aligned}\right\} \tag{34}$$

in the prediction error covariance matrix $S$.

*Remark 3*. Denote that Riccati equation (34) has a unique stabilizing positive definite solution $S = S^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ such that the matrix $A + B_2L_{11} - \Lambda(C_2+L_{21})$ is stable [7].

Substituting (26) and (33) to (28) with

$$\tilde{y}_k = y_k - (C_2+L_{21})\hat{x}_k - L_{22}h_k$$

in mind, we obtain

$$\begin{aligned} \hat{x}_{k+1} = {} & (A+B_2L_{11} - \Lambda(C_2+L_{21}))\hat{x}_k \\ & + (B_2(C_\mathrm{c}+L_{12}) - \Lambda L_{22})h_k + \Lambda y_k. \end{aligned}$$

The last equation together with controller equations (11) shows that the sequence $\widehat{X} = (\hat{x}_k)_{-\infty<k<+\infty}$ is produced from the measurement $Y$ by the system $E(z)$ (i.e. $\widehat{X} = EY$ ) with $2n$-dimensional internal state $(\widehat{X}, H)$ and the state-space realization

$$E(z) \sim \left[\begin{array}{cc|c} A+B_2L_{11}-\Lambda(C_2+L_{21}) & B_2L_{12}+B_2C_\mathrm{c}-\Lambda L_{22} & \Lambda \\ 0 & A_\mathrm{c} & B_\mathrm{c} \\ \hline I_n & 0 & 0 \end{array}\right]. \tag{35}$$

The following lemma defines the state-space realization matrices of the state-estimating controller.

*Lemma 3*. Let the state-space realization matrices of stabilizing controller (11) be given by

$$\begin{aligned} &A_\mathrm{c} = A + B_2(C_\mathrm{c}+M_1) - \Lambda(C_2+M_2), \\ &B_\mathrm{c} = \Lambda, \end{aligned} \tag{36}$$

where

$$M_1 \triangleq L_{11} + L_{12}, \qquad M_2 \triangleq L_{21} + L_{22}, \tag{37}$$

and the matrix $\Lambda$ is expressed via the stabilizing solution of filtering algebraic Riccati equation (34). Then controller (11) is the state-estimating one.

*Proof*. Substituting (36) to (35) and applying Lemma 8 from Appendix, we obtain

$$E(z) \sim \left[\begin{array}{c|c} A+B_2(C_\mathrm{c}+M_1)-\Lambda(C_2+M_2) & \Lambda \\ \hline I_n & 0 \end{array}\right] = \left[\begin{array}{c|c} A_\mathrm{c} & B_\mathrm{c} \\ \hline I_n & 0 \end{array}\right],$$

i.e. the controller is the state-estimating one.

After designing the optimal state estimator, let us construct optimal estimate-feedback loop. By the state-estimating property of the controller $K$, the copy of its internal state $H^0$ coincides with the one-step predictor $\widehat{X}^0$ of the sequence $X^0$:

$$h_k^0 \equiv \hat{x}_k^0 \triangleq \mathbf{E}(x_k^0 \mid \mathcal{F}_{k-1}^Y), \qquad -\infty < k < +\infty. \tag{38}$$

Then the sequence $\widehat{X}^0$ defined by (38) and the sequence of the one-step predictors $\widehat{X}$ defined by (24) are governed by the equations

$$\left.\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + B_2M_1\hat{x}_k^0 + B_2u_k + \Lambda\tilde{y}_k \\ \hat{x}_{k+1}^0 &= (A+B_2(M_1+C_\mathrm{c}))\hat{x}_k^0 + \Lambda\tilde{y}_k \end{aligned}\right\}, \tag{39}$$

where

$$\tilde{y}_k = y_k - C_2\hat{x}_k - M_2\hat{x}_k^0 \tag{40}$$

is the zero-mean Gaussian white noise with the covariance matrix $\Theta$ defined by (34), the matrices $M_1$ and $M_2$ are given by (37). The one-step predictor of the output $Z$ by the measurement $Y$ is given by

$$\hat{z}_k \triangleq \mathbf{E}(z_k \mid \mathcal{F}_{k-1}^Y) = \begin{bmatrix} C_2\hat{x}_k \\ u_k \end{bmatrix}.$$

The covariance matrix of the corresponding prediction error $\tilde{z}_k \triangleq z_k - \hat{z}_k$ is

$$\mathbf{E}(\tilde{z}_k\tilde{z}_k^{\mathrm{T}})=\begin{bmatrix}C_2SC_2^{\mathrm{T}} & 0\\ 0 & 0\end{bmatrix},$$

where $S$ is the stabilizing solution of filtering Riccati equation (34), and also does not depend on the controller matrices. It follows that problem (22) reduces to the state-feedback LQ problem of minimizing the LQ-cost

$$\begin{aligned}J_{LQ}&\triangleq\mathbf{E}(\hat{z}_k^{\mathrm{T}}\hat{z}_k)=\mathbf{E}\left(\begin{bmatrix}\hat{x}_k\\ u_k\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}C_2^{\mathrm{T}}C_2 & 0\\ 0 & I_{m_2}\end{bmatrix}\begin{bmatrix}\hat{x}_k\\ u_k\end{bmatrix}\right)\\ &=\mathbf{E}\left(\begin{bmatrix}\hat{x}_k\\ \hat{x}_k^0\\ u_k\end{bmatrix}^{\mathrm{T}}\begin{bmatrix}C_2^{\mathrm{T}}C_2 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & I_{m_2}\end{bmatrix}\begin{bmatrix}\hat{x}_k\\ \hat{x}_k^0\\ u_k\end{bmatrix}\right)\end{aligned}\qquad(41)$$

in the framework of dynamics (39) over stabilizing controllers $K\in\mathcal{K}$ which is solved standardly [8].

The optimal control law minimizing LQ-cost (41) is given by

$$u_k^{\mathrm{o}}=N_1\hat{x}_k+N_2\hat{x}_k^0,$$

where the matrices $N_1$ and $N_2$ are expressed from the stabilizing solution $T_\star$ of the control algebraic Riccati equation

$$\left.\begin{aligned}T_\star&=A_\star^{\mathrm{T}}T_\star A_\star+C_\star^{\mathrm{T}}C_\star-N_\star^{\mathrm{T}}\Pi_\star N_\star\\ \Pi_\star&\triangleq B_\star^{\mathrm{T}}T_\star B_\star+I_{m_2}\\ N_\star&=[N_1\quad N_2]\triangleq-\Pi_\star^{-1}B_\star^{\mathrm{T}}T_\star A_\star\end{aligned}\right\},\qquad(42)$$

where

$$\left[\begin{array}{c|c}A_\star & B_\star\\ \hline C_\star & 0\end{array}\right]\triangleq\left[\begin{array}{cc|c}A & B_2M_1 & B_2\\ 0 & A+B_2(M_1+C_{\mathrm{c}}) & 0\\ \hline C_2 & 0 & 0\\ 0 & 0 & 0\end{array}\right].\qquad(43)$$

*Remark 4*. Equation (42) has a unique stabilizing positive definite solution $T_\star=T_\star^{\mathrm{T}}\in\mathbb{R}^{2n\times 2n}$ such that the matrix $A_\star+B_\star N_\star$ is stable [7].

*Remark 5*. Denote that for $U\equiv U^{\mathrm{o}}$ and $V\equiv 0$

$$J_{\mathrm{LQ}}^{\mathrm{o}}\triangleq\min J_{\mathrm{LQ}}=\mathbf{E}\left(\begin{bmatrix}\hat{x}_0\\ \hat{x}_0^0\end{bmatrix}^{\mathrm{T}}T_\star\begin{bmatrix}\hat{x}_0\\ \hat{x}_0^0\end{bmatrix}\right).$$

The following lemma defines the solution to Problems 1 and 2.

*Lemma 4*. Let the state-space realization matrices of stabilizing controller (11) be given by relations (36) of Lemma 3 together with

$$C_{\mathrm{c}}=N_1+N_2,\qquad(44)$$

where the matrices $N_1$ and $N_2$ are expressed via the stabilizing solution of control algebraic Riccati equation (42). Then controller (11) is a solution to Problems 1 and 2.

*Proof.* Let us substitute (44) and (40) to (39) and apply Lemma 8 from Appendix. Taking into account (36) and (44), we obtain

$$K(z)\sim\left[\begin{array}{c|c}A+B_2(C_{\mathrm{c}}+M_1)-\Lambda(C_2+M_2) & \Lambda\\ \hline N_1+N_2 & 0\end{array}\right]=\left[\begin{array}{c|c}A_{\mathrm{c}} & B_{\mathrm{c}}\\ \hline C_{\mathrm{c}} & 0\end{array}\right].$$

Lemmas 2, 3, and 4 establish a system of matrix algebraic nonlinear equations for finding the state-space realization matrices of optimal controller (11) solving normalized anisotropy-based $\mathcal{H}_\infty$ problem (7) for $n$-dimensional plant (1). This system includes the following cross-coupled equations: $(2n\times 2n)$ Riccati equation (14) for the worst-case shaping filter, $(2n\times 2n)$ Lyapunov equation (17), mean anisotropy equation (15), $(n\times n)$ filtering Riccati equation (34), $(2n\times 2n)$ control Riccati equation (42), expressions (36) and (44) for the controller matrices, as well as notational relations (12), (18), (19), (32), (37), and (43). Satisfying this equation system is sufficient for optimality of the obtained $n$-dimensional controller. Denote that in the case of zero mean anisotropy level $\alpha=0$ the solution to Problem 1 reduces to the solution to normalized LQG problem considered in [4], [5], and the above equation system reduces to the well-known two independent $(n\times n)$ filtering and control Riccati equations that can be solved separately. But in general case $\alpha>0$ the full cross-coupled equation system is solved numerically by means of specifically designed homotopy-based algorithm (see, for example, [9]) with the normalized LQG controller state-space realization matrices as an initial point.

## 2. Controller order reduction by anisotropic balanced truncation

### 2.1 Anisotropic characteristic values and anisotropic balanced coordinates

To introduce a new set of invariants for the anisotropic optimal closed-loop system, let us consider filtering and control algebraic Riccati equations (34) and (42) with the respective stabilizing solutions $S$ and $T_\star$.

Define the block partitioning

$$T_\star=\begin{bmatrix}T_{11} & T_{12}\\ T_{12}^{\mathrm{T}} & T_{22}\end{bmatrix},\qquad T_{ij}\in\mathbb{R}^{n\times n},\qquad(45)$$

of the stabilizing solution of control Riccati equation (42). Taking into account partitioning (45) and notation (43), equation (42) can be rewritten as

$$\left.\begin{aligned}T_{11}&=A^{\mathrm{T}}T_{11}A+C_2^{\mathrm{T}}C_2-N_1^{\mathrm{T}}\Pi N_1\\ T_{12}&=A^{\mathrm{T}}T_{11}B_2M_1+A^{\mathrm{T}}T_{12}(A+B_2M_1+B_2C_{\mathrm{c}})-N_1^{\mathrm{T}}\Pi N_2\\ T_{22}&=(A+B_2M_1+B_2C_{\mathrm{c}})^{\mathrm{T}}T_{22}(A+B_2M_1+B_2C_{\mathrm{c}})\\ &\quad+(B_2M_1)^{\mathrm{T}}T_{11}(B_2M_1)-N_2^{\mathrm{T}}\Pi N_2\\ &\quad+(B_2M_1)^{\mathrm{T}}T_{12}(A+B_2M_1+B_2C_{\mathrm{c}})\\ &\quad+(A+B_2M_1+B_2C_{\mathrm{c}})^{\mathrm{T}}T_{12}^{\mathrm{T}}(B_2M_1)\\ \Pi&\triangleq B_2^{\mathrm{T}}T_{11}B_2+I_{m_2}\\ N_1&\triangleq-\Pi^{-1}B_2^{\mathrm{T}}T_{11}A\\ N_2&\triangleq-\Pi^{-1}B_2^{\mathrm{T}}(T_{11}B_2M_1+T_{12}(A+B_2M_1+B_2C_{\mathrm{c}}))\end{aligned}\right\}\qquad(46)$$

with the matrix $M_1$ expressed by (37) through the stabilizing solution $R$ of algebraic Riccati equation (14).

*Remark 6*. It has been noted in [10] that if the state-space realization matrices of stabilizing controller (11) are given by relations (36) of Lemma 3 together with

$$\begin{aligned}C_{\mathrm{c}}&=N_1+N_2\\ &=-(B_2^{\mathrm{T}}(T_{11}+T_{12})B_2+I_{m_2})^{-1}B_2^{\mathrm{T}}(T_{11}+T_{12})(A+B_2M_1),\end{aligned}\qquad(47)$$

where the matrices $T_{11}, T_{12} \in \mathbb{R}^{n\times n}$ satisfy the first and second equations in system (46), then controller (11) is a solution to Problems 1 and 2.

In terms of block partitioning (45), for $U \equiv U^{\mathrm{o}}$ and $V \equiv 0$ we have

$$J_{LQ}^{\mathrm{o}} = E(\hat{x}_0^{\mathrm{T}} T_{11}\hat{x}_0 + \hat{x}_0^{\mathrm{T}} T_{12}\hat{x}_0^0 + (\hat{x}_0^0)^{\mathrm{T}} T_{12}^{\mathrm{T}}\hat{x}_0 + (\hat{x}_0^0)^{\mathrm{T}} T_{22}\hat{x}_0^0).$$

Since actually in the system closed by the state-estimating optimal controller $\hat{x}_0 \equiv \hat{x}_0^0$, the above expression gives

$$\begin{aligned}\mathbf{E}(\hat{x}_0^{\mathrm{T}} T\hat{x}_0) &= \min J_{\mathrm{LQ}} \\ &= \min \mathbf{E}(\hat{x}_k^{\mathrm{T}}(C_2+N_1+N_2)^{\mathrm{T}}(C_2+N_1+N_2)\hat{x}_k),\end{aligned} \qquad (48)$$

where

$$T \triangleq T_{11} + T_{12} + T_{12}^{\mathrm{T}} + T_{22}. \qquad (49)$$

From (46), (47) and (49) it follows that the matrix $T = T^{\mathrm{T}} > 0$ is the stabilizing solution of the following control Riccati equation

$$\left.\begin{aligned} &T = (A+B_2M_1)^{\mathrm{T}} T(A+B_2M_1) + C_2^{\mathrm{T}}C_2 - N^{\mathrm{T}}\Pi N \\ &\Pi \triangleq B_2^{\mathrm{T}} T B_2 + I_{m_2} \\ &N \triangleq -\Pi^{-1}B_2^{\mathrm{T}} T(A+B_2M_1)\end{aligned}\right\}. \qquad (50)$$

Since the matrix $A + B_2M_1$ is stable, equation (50) has a unique stabilizing solution [7].

Recall now that the stabilizing solution $S = S^{\mathrm{T}} > 0$ of the filtering algebraic Riccati equation

$$\left.\begin{aligned} &S = (A+B_2L_{11})S(A+B_2L_{11})^{\mathrm{T}} + B_2\Sigma_{11}B_2^{\mathrm{T}} - \Lambda\Theta\Lambda^{\mathrm{T}} \\ &\Theta \triangleq (C_2+L_{21})S(C_2+L_{21})^{\mathrm{T}} + \Sigma_{22} \\ &\Lambda \triangleq ((A+B_2L_{11})S(C_2+L_{21})^{\mathrm{T}} + B_2\Sigma_{12})\Theta^{-1}\end{aligned}\right\} \qquad (51)$$

is the covariance matrix of the prediction error $\tilde{x}_k = x_k - \hat{x}_k$ :

$$S = \mathbf{E}((x_k - \hat{x}_k)(x_k - \hat{x}_k)^{\mathrm{T}}). \qquad (52)$$

Now let us introduce a new set of invariants for the anisotropic optimal closed-loop system that will play a central role in reducing the order of the normalized anisotropic controller.

*Theorem 5*. Let the realization $(A, B_2, C_2)$ of plant (1) be minimal and let $T = T^{\mathrm{T}} > 0$ and $S = S^{\mathrm{T}} > 0$ be the stabilizing solutions of control and filtering algebraic Riccati equations (50) and (51), respectively. Then the eigenvalues of the matrix $TS$ are similarity invariants. Further, these eigenvalues are real and strictly positive. Let

$$\phi_1^2 \geqslant \phi_2^2 \geqslant \cdots \geqslant \phi_n^2 > 0$$

denote the $n$ eigenvalues of the matrix $TS$ arranged in decreasing order, then there exists a similarity transformation

$$(A, B_2, C_2) \to (Q^{-1}AQ, Q^{-1}B_2, C_2Q) \qquad (53)$$

with the matrix $Q$ nonsingular that transforms both $T$ and $S$ to the form

$$Q^{-1}TQ^{-\mathrm{T}} = Q^{\mathrm{T}}SQ = \Phi, \qquad (54)$$

where

$$\Phi \triangleq \begin{bmatrix} \phi_1 & 0 & \cdots & 0 \\ 0 & \phi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \phi_n \end{bmatrix}. \qquad (55)$$

*Proof*. Let us consider a nonsingular state transformation

$$x_k = Q\tilde{x}_k, \quad h_k = Q\tilde{h}_k, \quad Q \in \mathbb{R}^{n\times n}. \qquad (56)$$

Substituting (56) to plant and controller state-space equations (1), (11), one can verify the correctness of (53). Further substitution of the realization $(Q^{-1}AQ, Q^{-1}B, C_2Q)$ to equations (14), (17), (15), (50), and (51), as well as to expressions (36) and (44) for the controller matrices and notational relations (12), (18), (19), (32), (37), and (43) shows that under transformation (56)

$$\begin{aligned} &T \to \tilde{T} \triangleq Q^{-1}TQ^{-\mathrm{T}}, \\ &S \to \tilde{S} \triangleq Q^{\mathrm{T}}SQ,\end{aligned}$$

that is

$$TS \to \tilde{T}\tilde{S} = Q^{-1}TSQ.$$

Since $TS$ and $Q^{-1}TSQ$ are similar matrices, the eigenvalues of the matrix $TS$ are similarity invariants. The existence of a similarity transformation resulting in equality (54) follows from the positive definiteness of $T$ and $S$ (see Corollary 8.3.3 in [11]), which also implies that the eigenvalues of the matrix $TS$ are real and strictly positive.

*Definition 1*. The real positive values

$$\phi_1 \geqslant \phi_2 \geqslant \cdots \geqslant \phi_n > 0 \qquad (57)$$

defined in Theorem 5 are called *anisotropic characteristic values*.

*Definition 2*. When the stabilizing solutions $T$ and $S$ of respective control and filtering Riccati equations (50) and (51) are in form (54), (55), the system is said to be in *anisotropic balanced coordinates*, and the realization

$$(\tilde{A}, \tilde{B}_2, \tilde{C}_2) \triangleq (Q^{-1}AQ, Q^{-1}B_2, C_2Q)$$

is called *anisotropic balanced realization*.

Writing control and filtering algebraic Riccati equations (50) and (51) in the anisotropic balanced state-space representation $(\tilde{A}, \tilde{B}_2, \tilde{C}_2)$ yields, respectively,

$$\left.\begin{aligned} &\Phi = (\tilde{A}+\tilde{B}_2\tilde{M}_1)^{\mathrm{T}}\Phi(\tilde{A}+\tilde{B}_2\tilde{M}_1) + \tilde{C}_2^{\mathrm{T}}\tilde{C}_2 - \tilde{N}^{\mathrm{T}}\tilde{\Pi}\tilde{N} \\ &\tilde{\Pi} \triangleq \tilde{B}_2^{\mathrm{T}}\Phi\tilde{B}_2 + I_{m_2} \\ &\tilde{N} \triangleq -\tilde{\Pi}^{-1}\tilde{B}_2^{\mathrm{T}}\Phi(\tilde{A}+\tilde{B}_2\tilde{M}_1)\end{aligned}\right\}, \qquad (58)$$

$$\left.\begin{aligned} &\Phi = (\tilde{A}+\tilde{B}_2\tilde{L}_{11})\Phi(\tilde{A}+\tilde{B}_2\tilde{L}_{11})^{\mathrm{T}} + \tilde{B}_2\Sigma_{11}\tilde{B}_2^{\mathrm{T}} - \tilde{\Lambda}\tilde{\Theta}\tilde{\Lambda}^{\mathrm{T}} \\ &\tilde{\Theta} \triangleq (\tilde{C}_2+\tilde{L}_{21})\Phi(\tilde{C}_2+\tilde{L}_{21})^{\mathrm{T}} + \Sigma_{22} \\ &\tilde{\Lambda} \triangleq ((\tilde{A}+\tilde{B}_2\tilde{L}_{11})\Phi(\tilde{C}_2+\tilde{L}_{21})^{\mathrm{T}} + \tilde{B}_2\Sigma_{12})\tilde{\Theta}^{-1}\end{aligned}\right\} \qquad (59)$$

with

$$\tilde{M}_1 \triangleq M_1Q, \quad \tilde{L}_{11} \triangleq L_{11}Q, \quad \tilde{L}_{21} \triangleq L_{21}Q. \qquad (60)$$

The matrix $\Phi$ defined by (55) is a *unique* positive definite stabilizing solution to *both* of these algebraic Riccati equations, and due to this uniqueness all the relevant information related to the anisotropic characteristic values and the anisotropic balanced realization are concentrated in these two Riccati equations.

*Remark 7*. It must be understood that the anisotropic characteristic values just as the system anisotropic norm are functions of the external disturbance mean anisotropy level $\alpha \geqslant 0$. Strictly speaking, we should use the notation $\phi_i(\alpha)$ and $\Phi(\alpha)$, but for the sake of simplicity we apply notations $\phi_i$ and $\Phi$.

Let $T_2 = T_2^{\mathrm{T}} > 0$ and $S_2 = S_2^{\mathrm{T}} > 0$ be the respective stabilizing solutions to dual control and filtering algebraic Riccati equations for the discrete-time LQG problem

$$T_2 = A^{\mathrm{T}} T_2 A + C_2^{\mathrm{T}} C_2 - A^{\mathrm{T}} T_2 B_2 (B_2^{\mathrm{T}} T_2 B_2 + I_{m_2})^{-1} B_2^{\mathrm{T}} TA, \qquad (61)$$

$$S_2 = AS_2 A^{\mathrm{T}} + B_2 B_2^{\mathrm{T}} - AS_2 C_2^{\mathrm{T}} (C_2 S_2 C_2^{\mathrm{T}} + I_{m_2})^{-1} C_2 S_2 A^{\mathrm{T}}. \qquad (62)$$

Then one can define the LQG characteristic values for the discrete-time case similarly to [4] as

$$\begin{aligned} &\psi_1 \geqslant \psi_2 \geqslant \cdots \geqslant \psi_n > 0, \\ &\psi_i^2 = \lambda_i\{T_2 S_2\}, \quad i = \overline{1,n}. \end{aligned} \qquad (63)$$

The following theorem establishes some properties of the anisotropic characteristic values.

*Theorem 6*. Let the realization $(A, B_2, C_2)$ of plant (1) be minimal, and let the LQG characteristic values for this realization be defined by (63). For anisotropic characteristic values (57), the following statements hold true:

(1) $\phi_i \geqslant \psi_i$ and $\phi_i = \psi_i$ iff $\alpha = 0$;

(2) each anisotropic characteristic value $\phi_i$ is a monotonically increasing function of the parameter $\alpha$;

(3) if the anisotropic characteristic values $\phi_i$ are different, then $\dfrac{d\phi_i}{d\alpha} \geqslant 0$;

(4) each anisotropic characteristic value $\phi_i$ is a continuous function of the parameter $\alpha$.

*Proof*. (1) Applying the results of [12], [13] to equations (50), (61) and (34), (62), we obtain that

$$T \geqslant T_2 \quad \text{and} \quad S \geqslant S_2.$$

It follows that [11]

$$\begin{aligned} &T^{1/2} S T^{1/2} \geqslant T^{1/2} S_2 T^{1/2}, \\ &S_2^{1/2} T S_2^{1/2} \geqslant S_2^{1/2} T_2 S_2^{1/2}, \end{aligned}$$

and

$$\lambda_i\{TS\} \geqslant \lambda_i\{TS_2\} \geqslant \lambda_i\{T_2 S_2\},$$

which implies $\phi_i \geqslant \psi_i$. Equality is attained with $\alpha = 0$ since in this case $T = T_2$ and $S = S_2$ that completes the proof of the first assertion.

(2) It is known from [14] that the anisotropic norm of a system is a monotonically increasing differentiable function of the parameter $\alpha$. It means that $\alpha_2 \geqslant \alpha_1 \geqslant 0$ always implies

$$\| \mathcal{F}_l(P, K(\alpha_2)) \|_{\alpha_2} \geqslant \| \mathcal{F}_l(P, K(\alpha_2)) \|_{\alpha_1}$$

with obvious notations. From the other hand, the anisotropic controller $K(\alpha_1)$ minimizes $\alpha_1$-anisotropic norm of the closed-loop system that yields

$$\| \mathcal{F}_l(P, K(\alpha_2)) \|_{\alpha_1} \geqslant \| \mathcal{F}_l(P, K(\alpha_1)) \|_{\alpha_1}.$$

The resulting chain of inequalities

$$\| \mathcal{F}_l(P, K(\alpha_2)) \|_{\alpha_2} \geqslant \| \mathcal{F}_l(P, K(\alpha_2)) \|_{\alpha_1} \geqslant \| \mathcal{F}_l(P, K(\alpha_1)) \|_{\alpha_1}$$

means that the anisotropic norm of the closed-loop system increases monotonically as a function of the parameter $\alpha$. Applying again the results of [12], [13] to equations (50) and (51) obtained for the different mean anisotropy levels $\alpha_1$ and $\alpha_2$, it can be shown that

$$T(\alpha_2) \geqslant T(\alpha_1) \quad \text{and} \quad S(\alpha_2) \geqslant S(\alpha_1).$$

Using the same argument as in the proof of the first assertion, this implies that $\phi_i(\alpha_2) \geqslant \phi_i(\alpha_1)$. In fact, $T$ and $S$ are differentiable functions of $\alpha$, so we have

$$\frac{dT}{d\alpha} \geqslant 0 \quad \text{and} \quad \frac{dS}{d\alpha} \geqslant 0.$$

(3) Since $T$ and $S$ are differentiable functions of $\alpha$, hence $TS$ is also a differentiable function of $\alpha$. Since by definition $\phi_i^2 = \lambda_i\{TS\}$ and by assumption $\phi_i$ are different, then each anisotropic characteristic value $\phi_i$ is a differentiable function of $\alpha$ too. But from the second assertion, each $\phi_i$ is a monotonically increasing function of $\alpha$. Therefore, $\dfrac{d\phi_i}{d\alpha} \geqslant 0$ holds for each $\phi_i$.

(4) From the proof of the third assertion, $TS$ is a differentiable, hence, continuous function of $\alpha$. It is well known (see e.g. [11]) that the eigenvalues of a matrix are continuous functions of the matrix elements that completes the proof.

Denote that the transformation matrix $Q$ putting the closed-loop system realization into the anisotropic balanced coordinates can be found in the following quite standard way (see, for example, [15]). Let us find an upper triangular nonsingular matrices $\mathcal{T}, \mathcal{S} \in \mathbb{R}^{n \times n}$ from the respective Cholesky factorizations of the stabilizing solution $T$ of control Riccati equation (50)

$$T = \mathcal{T}^{\mathrm{T}} \mathcal{T}$$

and the stabilizing solution $S$ of filtering Riccati equation (51)

$$S = \mathcal{S}^{\mathrm{T}} \mathcal{S}.$$

Then, find the singular value decomposition of the matrix

$$\mathcal{S} \mathcal{T}^{\mathrm{T}} = \mathcal{U} \Phi \mathcal{V}^{\mathrm{T}},$$

where $\mathcal{U}\mathcal{U}^{\mathrm{T}} = I$, $\mathcal{V}\mathcal{V}^{\mathrm{T}} = I$. Then the transformation matrix is given by

$$Q = \mathcal{T}^{\mathrm{T}} \mathcal{V} \Phi^{-1/2}.$$

### 2.2 Reduced-order plant and controller

Before proceeding to the controller order reduction, let us consider some motivation for the anisotropic balancing of Theorem 5 and summarize the essence of the results. First, the matrices $\tilde{T}$ and $\tilde{S}$ are diagonal. Therefore, taking into account expressions (48) for minimum value of the LQ-cost and (52) for the covariance matrix of prediction error, one could say that the similarity transformation $Q$ defined by (53) decouples the state components in both the control and filtering problems. Second, since $\tilde{T}$ and $\tilde{S}$ are equal, this transformation also weights all of the state components equally between the control and the filtering problems. This weight or importance of the state component $\tilde{X}_i$ in the normalized anisotropy-based stochastic $\mathcal{H}_\infty$ problem is the anisotropic characteristic value $\phi_i$, since $\phi_i$ is the filtering error covariance for the component $\tilde{X}_i$ and, at the same time, the cost induced by an initial condition aligned with $\tilde{X}_i$. This fact carries over into the controller design problem, as the anisotropic optimal controller is the cascade of optimal estimator (35) and control gain (44). Thus, the anisotropic characteristic value $\phi_i$ specifies how much the state component $\tilde{X}_i$ participates in the closed-loop behaviour of the system in the following sense. If $\phi_i$ is large, then the component $\tilde{X}_i$ is difficult to filter (see (52) and (54)) and difficult to control (see (48) and (54)), hence, $\tilde{X}_i$ is an important state component that must be taken into consideration in the controller design. Vice versa, if the anisotropic characteristic value is small, then $\tilde{X}_i$ is easy to filter and easy to control, hence, the component $\tilde{X}_i$ is not of the great essence that can be discarded for designing a reduced-order controller.

Let the state-space realization $(\tilde{A},\tilde{B}_2,\tilde{C}_2)$ be minimal with $n$ states and in anisotropic balanced coordinates with anisotropic characteristic values

$$\phi_1 \geqslant \phi_2 \geqslant \cdots \geqslant \phi_n > 0.$$

That is, $\Phi = \operatorname{diag}\{\phi_1,\ldots,\phi_n\} = \tilde{T} = \tilde{S}$ is the stabilizing solution of control and filtering Riccati equations (58) and (59) associated with the realization $(\tilde{A},\tilde{B}_2,\tilde{C}_2)$. Fix $r < n$ such that $\phi_r > \phi_{r+1}$ and partition the matrix $\Phi$ accordingly into

$$\Phi = \begin{bmatrix} \Phi_1 & 0 \\ 0 & \Phi_2 \end{bmatrix} \tag{64}$$

with

$$\Phi_1 = \operatorname{diag}\{\phi_1,\ldots,\phi_r\}, \quad \Phi_2 = \operatorname{diag}\{\phi_{r+1},\ldots,\phi_n\}. \tag{65}$$

Partition the matrices $\tilde{A}$, $\tilde{B}_2$, and $\tilde{C}_2$ conformably with the partitioning (64) as follows:

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix}, \quad \tilde{B}_2 = \begin{bmatrix} \tilde{B}_{21} \\ \tilde{B}_{22} \end{bmatrix}, \quad \tilde{C}_2 = \begin{bmatrix} \tilde{C}_{21} & \tilde{C}_{22} \end{bmatrix}.$$

Then the reduced-order realization with $r$-dimensional state is $(\tilde{A}_{11},\tilde{B}_{21},\tilde{C}_{21})$.

Let $K(z) = (\tilde{A}_{\mathrm{c}},\tilde{B}_{\mathrm{c}},\tilde{C}_{\mathrm{c}})$ be the balanced realization of the normalized anisotropic controller for plant (1) as defined in Lemma 4. Partition the matrices $\tilde{A}_{\mathrm{c}}$, $\tilde{B}_{\mathrm{c}}$, and $\tilde{C}_{\mathrm{c}}$ conformably with the partitioning (64) as follows:

$$\tilde{A}_{\mathrm{c}} = \begin{bmatrix} \tilde{A}_{\mathrm{c}11} & \tilde{A}_{\mathrm{c}12} \\ \tilde{A}_{\mathrm{c}21} & \tilde{A}_{\mathrm{c}22} \end{bmatrix}, \quad \tilde{B}_{\mathrm{c}} = \begin{bmatrix} \tilde{B}_{\mathrm{c}1} \\ \tilde{B}_{\mathrm{c}2} \end{bmatrix}, \quad \tilde{C}_{\mathrm{c}} = \begin{bmatrix} \tilde{C}_{\mathrm{c}1} & \tilde{C}_{\mathrm{c}2} \end{bmatrix}.$$

Then the reduced-order controller with $r$-dimensional state is

$$K_r(z) \sim \left[\begin{array}{c|c} \tilde{A}_{\mathrm{c}11} & \tilde{B}_{\mathrm{c}1} \\ \hline \tilde{C}_{\mathrm{c}1} & 0 \end{array}\right]. \tag{66}$$

As it can be easily shown by substitution, the matrix $\Phi_1$ given by (65) is the stabilizing solution to the corresponding control and filtering Riccati equations

$$\left.\begin{aligned} \Phi_1 &= (\tilde{A}_{11} + \tilde{B}_{21}\tilde{M}_{11})^{\mathrm{T}}\Phi_1(\tilde{A}_{11} + \tilde{B}_{21}\tilde{M}_{11}) \\ &\quad + \tilde{C}_{21}^{\mathrm{T}}\tilde{C}_{21} - \tilde{N}_1^{\mathrm{T}}\tilde{\Pi}_{11}\tilde{N}_1 \\ \tilde{\Pi}_{11} &\triangleq \tilde{B}_{21}^{\mathrm{T}}\Phi_1\tilde{B}_{21} + I_{m_2} \\ \tilde{N}_1 &\triangleq -\tilde{\Pi}_{11}^{-1}\tilde{B}_{21}^{\mathrm{T}}\Phi_1(\tilde{A}_{11} + \tilde{B}_{21}\tilde{M}_{11}) \end{aligned}\right\},$$

$$\left.\begin{aligned} \Phi_1 &= (\tilde{A}_{11} + \tilde{B}_{21}\tilde{L}_{11})\Phi_1(\tilde{A}_{11} + \tilde{B}_{21}\tilde{L}_{11})^{\mathrm{T}} \\ &\quad + \tilde{B}_{21}\Sigma_{11}\tilde{B}_{21}^{\mathrm{T}} - \tilde{\Lambda}_1\tilde{\Theta}_{11}\tilde{\Lambda}_1^{\mathrm{T}} \\ \tilde{\Theta}_{11} &\triangleq (\tilde{C}_{21} + \tilde{L}_{21})\Phi_1(\tilde{C}_{21} + \tilde{L}_{21})^{\mathrm{T}} + \Sigma_{22} \\ \tilde{\Lambda}_1 &\triangleq ((\tilde{A}_{11} + \tilde{B}_{21}\tilde{L}_{11})\Phi_1(\tilde{C}_{21} + \tilde{L}_{21})^{\mathrm{T}} + \tilde{B}_{21}\Sigma_{12})\tilde{\Theta}_{11}^{-1} \end{aligned}\right\},$$

for the reduced-order realization $(\tilde{A}_{11},\tilde{B}_{21},\tilde{C}_{21})$. From this fact it immediately follows that the reduced-order controller $K_r(z)$ is the full-order normalized anisotropic optimal controller for the reduced-order plant $P_r(z)$ with the realization $(\tilde{A}_{11},\tilde{B}_{21},\tilde{C}_{21})$.

Of course, there are two important questions of stability and performance of the closed-loop system when reduced-order controller (66) is connected to full-order plant (1). Development of apriori conditions for closed-loop stability is now in progress. The lack of performance in terms of the anisotropic norm can be expressed as

$$|||F|||_\alpha = |||F - F_r|||_\alpha, \tag{67}$$

where

$$F_e(z) \sim \left[\begin{array}{c|c} A_e & B_e \\ \hline C_e & 0 \end{array}\right] = \left[\begin{array}{cc|c} A_{\mathrm{cl}} & 0 & B_{\mathrm{cl}} \\ 0 & A_{\mathrm{cl}r} & B_{\mathrm{cl}r} \\ \hline C_{\mathrm{cl}} & -C_{\mathrm{cl}r} & 0 \end{array}\right] \tag{68}$$

is the error model, $F$ is the closed-loop system defined by (12), and

$$F_r(z) \sim \left[\begin{array}{c|c} A_{\mathrm{cl}r} & B_{\mathrm{cl}r} \\ \hline C_{\mathrm{cl}r} & 0 \end{array}\right] = \left[\begin{array}{cc|cc} A & B_2C_{\mathrm{c}1} & B_2 & 0 \\ B_{\mathrm{c}1}C_2 & A_{\mathrm{c}11} & 0 & B_{\mathrm{c}1} \\ \hline C_2 & 0 & 0 & 0 \\ 0 & C_{\mathrm{c}1} & 0 & 0 \end{array}\right] \tag{69}$$

is the closed-loop system with a reduced-order controller.

The following theorem defines the value of performance error (67).

*Theorem 7*. Let the full-order closed-loop system $F(z)$ be given by (12), and let the reduced-order closed-loop system $F_r(z)$ represented by (69) be stable. Then the anisotropic norm of the error model $F_e(z)$ with realization (68) is given by

$$|||F_e(z)|||_\alpha = \left\{\frac{1}{q_e}\left(1 - \frac{m_1}{\operatorname{tr}(L_eP_eL_e^{\mathrm{T}} + \Sigma_e)}\right)\right\}^{1/2},$$

where $q_e \in [0, \|F_e\|_\infty^{-2})$, $L_e$, $\Sigma_e = \Sigma_e^{\mathrm{T}} > 0$, and $P_e = P_e^{\mathrm{T}} > 0$ satisfy the equation system

$$\left.\begin{array}{l} R_e = A_e^{\mathrm{T}} R_e A_e + q_e C_e^{\mathrm{T}} C_e + L_e^{\mathrm{T}} \Sigma_e^{-1} L_e \\ \Sigma_e \triangleq (I_{m_1} - B_e^{\mathrm{T}} R_e B_e)^{-1} \\ L_e \triangleq \Sigma_e B_e^{\mathrm{T}} R_e A_e \end{array}\right\}, \qquad (70)$$

$$-\frac{1}{2}\ln\det\left\{\frac{m_1 \Sigma_e}{\operatorname{tr}(L_e P_e L_e^{\mathrm{T}} + \Sigma_e)}\right\} = \alpha, \qquad (71)$$

$$P_e = (A_e + B_e L_e) P_e (A_e + B_e L_e)^{\mathrm{T}} + B_e \Sigma_e B_e^{\mathrm{T}}. \qquad (72)$$

At that, the solution $(q_e, R_e = R_e^{\mathrm{T}} > 0, P_e)$ to equation system (70)–(72) is a unique one.

Proof of this theorem immediately follows from Theorem 2 in [1] applied to error model (68).

## 3. Application example: longitudinal flight control

As an application example, let us briefly consider the problem of longitudinal flight control aimed at wind disturbance attenuation for aircraft in landing approach along glidepath with prescribed relative slope angle in presence of coloured random noises by means of reduced-order anisotropic controller. More detailed problem statement and aircraft model can be found in [16]. The obtained control law minimizes the influence of actuator and measurement noises, as well as wind disturbance on deviations of airspeed $\Delta V$ and altitude $\Delta h$ from prescribed values (controlled variables). Deviation of generalized elevators $\Delta\vartheta_{cy}$ and throttle lever $\Delta\delta_t$ are considered as aircraft control.

The anisotropic, LQG, and $\mathcal{H}_\infty$ controllers were designed for aircraft TU-154 in landing approach along glidepath with fixed relative slope angle $\theta_0 = -2.7$ deg. Nonlinear equations describing an aircraft longitudinal motion (see [16]) were linearized in the trajectory point with airspeed $V_0 = 71.375$ m/sec and altitude $h_0 = 600$ m. The resulted standard plant model (1) has order $n = 6$.

The full-order normalized anisotropic optimal controllers was found for two different prescribed levels of mean anisotropy of random disturbances $\alpha_1 = 0.01$ and $\alpha_2 = 0.6$; the suboptimal $\mathcal{H}_\infty$ controller was obtained for $\gamma = 2.6523$. Then, the closed-loop systems with anisotropic, LQG, and $\mathcal{H}_\infty$ controllers were put into the balanced coordinates via the respective nonsingular transformations. Anisotropic, LQG, and $\mathcal{H}_\infty$ characteristic values for this problems are given in descending order in Table 1. Denote that the $\mathcal{H}_\infty$ reduced-order controller does not provide stability of the closed-loop system for $r < 5$. The reduced anisotropic and LQG controllers retain the stable closed-loop up to $r = 3$.

The results of computer simulation for systems closed by the reduced-order controllers are presented in Fig. 3 through 9. The deterministic horizontal and vertical components of wind disturbance are presented by the model in the form of vortex ring considered in [16] (see Fig. 3). The worst-case random coloured noises are presented in Fig. 4 and 7. The transients and control signals in the closed-loop systems with various controllers are given in Fig. 5, 6, and 8, 9. Black-coloured curves in the plots correspond to the closed-loop system with anisotropic controllers, blue and red colours — to the systems closed by LQG and $\mathcal{H}_\infty$ controllers, respectively.

| $i$ | LQG | $\mathcal{H}_\infty$ | Anisotropic: <br> $\alpha = 0.01$ | <br> $\alpha = 0.6$ |
|---|---|---|---|---|
| 1 | 2.5102 | 2.6369 | 6.5624 | 37.3321 |
| 2 | 0.8492 | 0.8925 | 1.3688 | 7.1399 |
| 3 | 0.5362 | 0.5611 | 0.7050 | 2.5277 |
| 4 | 0.0879 | 0.0900 | 0.0905 | 0.0993 |
| 5 | 0.0681 | 0.0693 | 0.0694 | 0.0712 |
| 6 | 0.0119 | 0.012430 | 0.012432 | 0.0127 |

**Tab.1 LQG, $\mathcal{H}_\infty$, and anisotropic characteristic values**

| Controller: | LQG | Anisotropic | $\mathcal{H}_\infty$ |
|---|---|---|---|
| $\max\lvert\Delta V\rvert$, m/sec | 9.625 | 11.11 | 12.56 |
| $\max\lvert\Delta h\rvert$, m | 60.94 | 45.61 | 29.15 |
| $\max\lvert\Delta\vartheta_{cy}\rvert$, deg | 14.12 | 18.49 | 29.66 |
| $\max\lvert\Delta\delta_t\rvert$, deg | 4.407 | 4.605 | 3.683 |

**Tab.2 Comparison for reduced-order controllers, $\alpha = 0.01$, $r = 5$**

| Controller: | LQG | Anisotropic |
|---|---|---|
| $\max\lvert\Delta V\rvert$, m/sec | 9.53 | 11.87 |
| $\max\lvert\Delta h\rvert$, m | 60.79 | 46.84 |
| $\max\lvert\Delta\vartheta_{cy}\rvert$, deg | 16.85 | 21.91 |
| $\max\lvert\Delta\delta_t\rvert$, deg | 8.689 | 9.108 |

**Tab.3 Comparison for reduced-order controllers, $\alpha = 0.01$, $r = 3$**

| Controller: | LQG | Anisotropic | $\mathcal{H}_\infty$ |
|---|---|---|---|
| $\max\lvert\Delta V\rvert$, m/sec | 9.652 | 11.69 | 12.84 |
| $\max\lvert\Delta h\rvert$, m | 63.39 | 40.96 | 29.64 |
| $\max\lvert\Delta\vartheta_{cy}\rvert$, deg | 14.5 | 23.1 | 31.13 |
| $\max\lvert\Delta\delta_t\rvert$, deg | 5.089 | 4.636 | 3.832 |

**Tab.4 Comparison for reduced-order controllers, $\alpha = 0.6$, $r = 5$**

| Controller: | LQG | Anisotropic |
|---|---|---|
| $\max\lvert\Delta V\rvert$, m/sec | 9.509 | 12.74 |
| $\max\lvert\Delta h\rvert$, m | 56.73 | 38.39 |
| $\max\lvert\Delta\vartheta_{cy}\rvert$, deg | 16.01 | 24.73 |
| $\max\lvert\Delta\delta_t\rvert$, deg | 8.227 | 6.981 |

**Tab.5 Comparison for reduced-order controllers, $\alpha = 0.01$, $r = 5$**

The comparison results in numbers are brought together in Tables 2 through 5. The comparison shows that the maximum absolute deviation of airspeed $\Delta V$ is lesser for system with the LQG controllers, whereas the maximum absolute deviation of the prescribed altitude $\Delta h$ is lesser for system closed by the $\mathcal{H}_\infty$ controller with $r = 5$. The maximum absolute value of the control signal $\Delta\vartheta_{cy}$ for $r = 5$

was shown by the $\mathcal{H}_\infty$ controller, and for $r=3$ by the anisotropic one, whereas the minimum absolute amplitude was demonstrated by LQG controller in both cases. As for the maximum absolute value of $\Delta\delta_t$, for $r=5$ it was given by the anisotropic controller, and for $r=3$ by the LQG one.

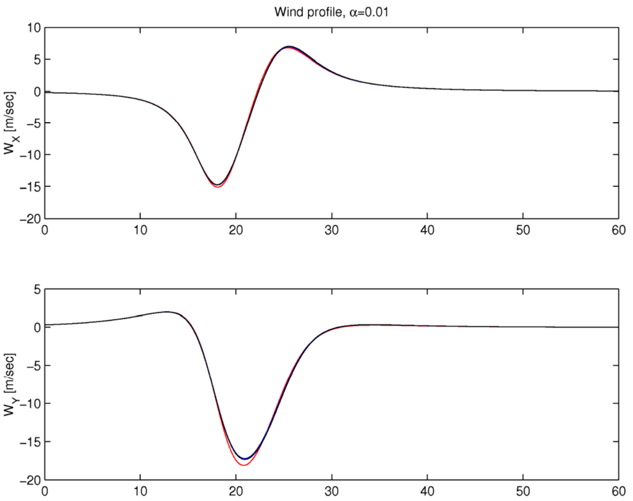


**Fig.3 Horizontal and vertical components $W_x$ and $W_y$ of wind profile**

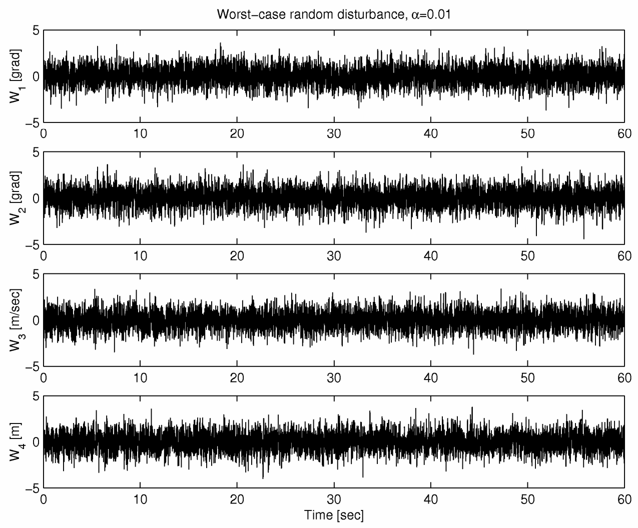


**Fig.4 Worst-case random disturbance, $\alpha=0.01$**

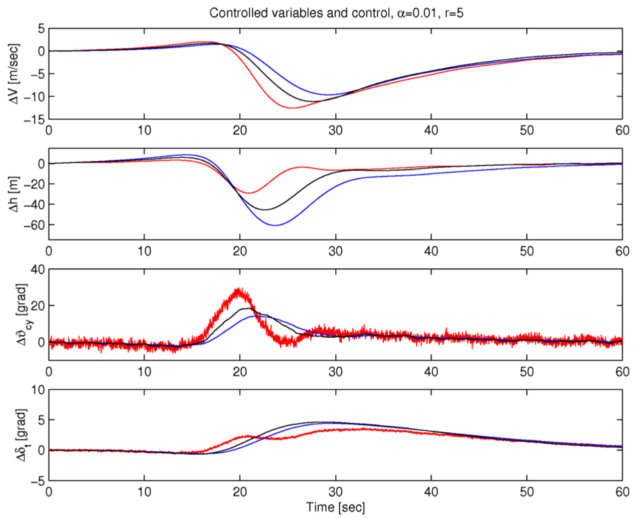


**Fig.5 Controlled variables $\Delta V$, $\Delta h$, and control $\Delta\vartheta_{cy}$, $\Delta\delta_t$ for $\alpha=0.01$, $r=5$**

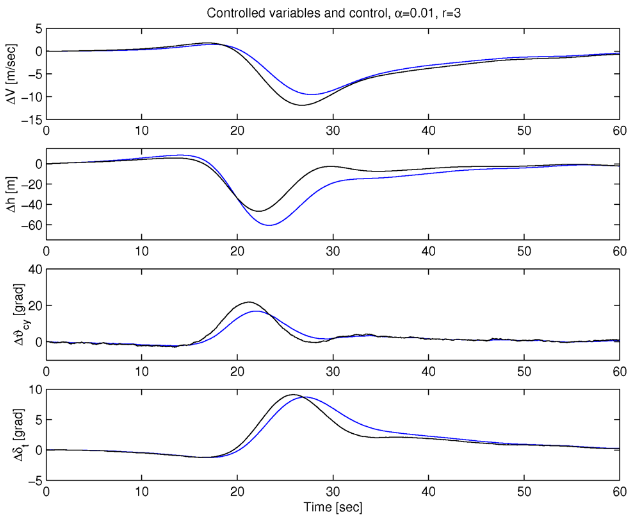


**Fig.6 Controlled variables $\Delta V$, $\Delta h$, and control $\Delta\vartheta_{cy}$, $\Delta\delta_t$ for $\alpha=0.01$, $r=3$**

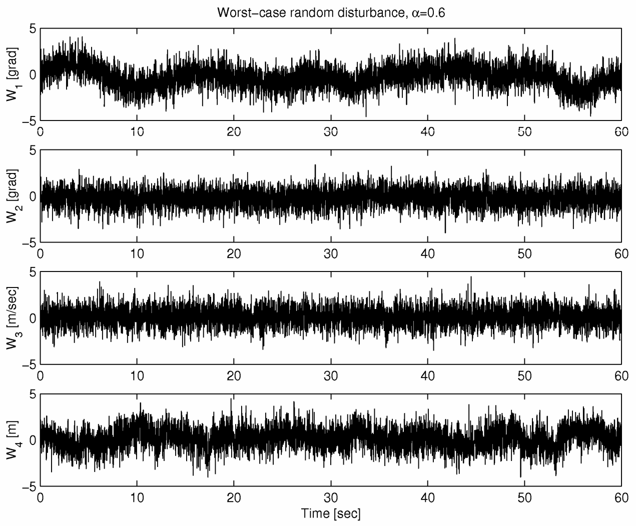


**Fig.7 Worst-case random disturbance, $\alpha=0.6$**

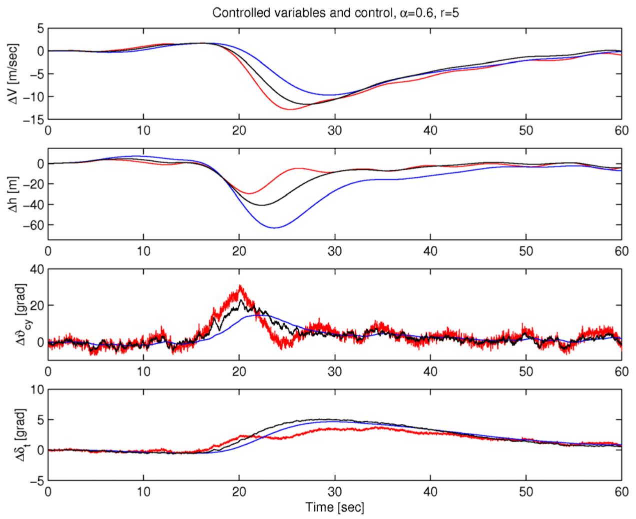


**Fig.8 Controlled variables $\Delta V$, $\Delta h$, and control $\Delta\vartheta_{cy}$, $\Delta\delta_t$ for $\alpha=0.6$, $r=5$**

From Fig. 5 and 8 it can be seen that the control signals generated by the anisotropic controller are more smooth than that of the $\mathcal{H}_\infty$ controller, whereas the latter, owing to its conservatism, strives to counteract each element of the noise random sequence, interpreting it as a deterministic signal. In practice, such the control would not likely be physically realizable or would require using of additional smoothing filters.

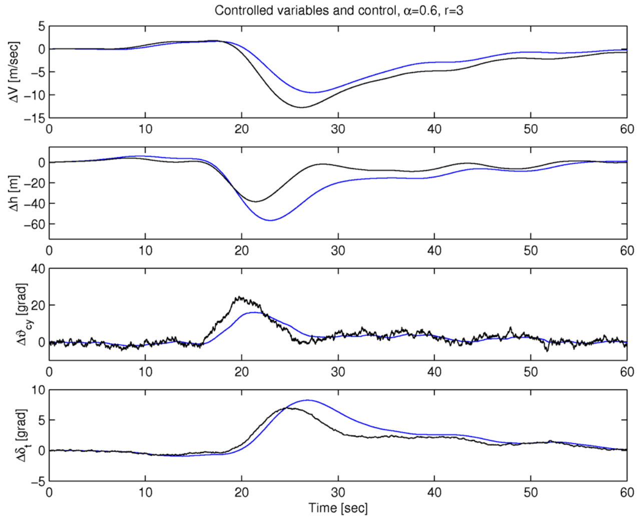


**Fig.9 Controlled variables $\Delta V$, $\Delta h$, and control $\Delta\vartheta_{cy}$, $\Delta\delta_t$ for $\alpha = 0.6$, $r = 3$**

## Conclusion

This paper presents the truncation technique for reducing order of normalized anisotropic optimal closed-loop system aimed at reduced-order controller design. Truncation is carried out for the closed-loop realization in anisotropic balanced coordinates, when the product of respective filtering and control Riccati equation solutions is a diagonal matrix with the squares of anisotropic characteristic values situated in descending order on its main diagonal. In anisotropic balanced coordinates, small characteristic values correspond to the states which are easy to filter and control in a sense of anisotropic norm. The part of the plant or controller corresponding to smaller anisotropic characteristic values is truncated to obtain a reduced-order plant or controller. It was shown that the reduced-order controller is the full-order optimal one for the reduced-order plant. Development of apriori stability conditions for the closed-loop system consisting of full-order plant and reduced-order controller is now in progress. As an application example, we consider the longitudinal aircraft control problem aimed at random disturbance attenuation by means of the reduced-order anisotropic controller. Simulation for aircraft in landing approach along glidepath with fixed relative slope angle shows that the reduced-order anisotropic controller retains the inherent properties of the full-order one. Comparison between reduced-order anisotropic, LQG, and $\mathcal{H}_\infty$ controllers in presence of the worst-case random and deterministic disturbances demonstrates the main advantages of the anisotropic controller, namely, smoothness and physical realizability of control signals together with sufficiently good attenuation of random and deterministic disturbances. In this problem, the reduced-order anisotropic controllers also show lesser order preserving closed-loop system stability than the reduced-order $\mathcal{H}_\infty$ controller.

### Acknowledgements

This work was supported by Russian Foundation for Basic Research (grant 08-08-00567-a) and Program #15 of EMMCD RAS.

## References

[1] VLADIMIROV, I.G., KURDYKOV, A.P., SEMYONOV, A.V.: On Computing the Anisotropic Norm of Linear Discrete-Time-Invariant Systems. *Proc. 13th IFAC World Congress*, San-Francisco, CA, pages 179–184, 1996.

[2] VLADIMIROV, I.G., KURDYKOV, A.P., SEMYONOV, A.V.: State-Space Solution to Anisotropy-Based Stochastic $\mathcal{H}_\infty$-Optimization Problem. *Proc. of 13th IFAC World Congress*, San-Francisco, CA, pages 427–432, 1996.

[3] KURDYUKOV, A.P., TCHAIKOVSKY, M.M.: Model Reduction According to Minimum Anisotropic Norm Performance. Stability and Oscillations of Nonlinear Control Systems: *Book of Abstracts of E.S. Pyatnitskiy X International Workshop*, Moscow, pages 166–167, 2008.

[4] JONCKHERE, A.E., SILVERMAN, L.M.: A New Set of Invariants for Linear Systems — Application to Reduced Order Compensator Design. *IEEE Trans. AC*, 28:953–964, 1983.

[5] MUSTAFA, D,., GLOVER, K.: Controller Reduction by $\mathcal{H}_\infty$-Balanced Truncation. *IEEE Trans. AC*, 36:668–682, 1991.

[6] LIPTSER, R.S., SHIRYAEV, A.N.: *Statistics of Random Processes*. Springer-Verlag, New-York, 1977.

[7] MOLINARI, B.P.: The Stabilizing Solution of the Discrete Algebraic Riccati Equation. *IEEE Trans. AC*, AC-20:396–399, 1975.

[8] DORATO, P., LEVIS, A.H.: Optimal Linear Regulators: The Discrete-Time Case. *IEEE Trans. AC*, AC-16:613–620, 1971.

[9] KURDYUKOV, A.P., MAXIMOV, E.A., TCHAIKOVSKY, M.M.: Homotopy Method for Aolving Anisotropy-Based Stochastic $\mathcal{H}_\infty$-Optimization Problem with Uncertainty. *Proc. 5th IFAC Symposium on Robust Control Design*, Toulouse, France, 2006.

[10] TCHAIKOVSKY, M.M., KURDYUKOV, A.P.: On Simplifying Solution to Normalized Anisotropy-Based Stochastic $\mathcal{H}_\infty$ Problem. *Proc. 6th IFAC Symposium on Robust Control Design*, Haifa, Israel, 2009.

[11] BERNSTEIN, D.S.: *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory*. Princeton University Press, Princeton, NJ, 2005.

[12] WIMMER, H.K.: Monotonicity and Maximality of Solutions of Discrete-Time Algebraic Riccati Equations. *J. Math. Systems Estimat. Control*, 2:219–235, 1992.

[13] CLEMENTS, D.J., WIMMER, H.K.: Monotonicity of the Optimal Cost in the Discrete-Time Regulator Problem and Schur Complements. *Automatica*, 37:1779–1786, 2001.

[14] DIAMOND, P., VLADIMIROV, I.G., KURDYUKOV, A.P., SEMYONOV, A.V.: Anisotropy-Based Performance Analysis of Linear Discrete Time-Invariant Control Systems. *Int. J. of Control*, 74:28–42, 2001.

[15] DATTA, B.N.: *Numerical Methods for Linear Control Systems Design and Analysis*. Elseveir Academic Press, San Diego, CA, 2004.

[16] KURDYUKOV, A.P., PAVLOV, B.V., TIMIN, V.N., VLADIMIROV, I.G.: Longitudinal Anisotropy-Based Flight Control in a Wind Shear. *Proc. 16th IFAC Symposium on Automatic Control in Aerospace*, Saint-Petersburg, Russia, 2004.

obsah
articles

## Appendix: An equivalence lemma

*Lemma 8*. Let $A_1, A_2 \in \mathbb{R}^{n\times n}$. Then

$$\left[\begin{array}{cc|c} A_1 & A_2 - A_1 & B \\ 0 & A_2 & B \\ \hline C_1 & C_2 & D \end{array}\right] \sim \left[\begin{array}{c|c} A_2 & B \\ \hline C_1 + C_2 & D \end{array}\right], \tag{A.1}$$

i.e. these realizations result in the same state-space input-output operator.

*Proof*. Let $(X, \Xi)$, $U$, and $Y$ be the internal state, input, and output of the system with realization at the left-hand side of (A.1). All these signals are related by the equations

$$\left[\begin{array}{c} x_{k+1} \\ \xi_{k+1} \\ \hline y_k \end{array}\right] = \left[\begin{array}{cc|c} A_1 & A_2 - A_1 & B \\ 0 & A_2 & B \\ \hline C_1 & C_2 & D \end{array}\right] \left[\begin{array}{c} x_k \\ \xi_k \\ \hline u_k \end{array}\right]. \tag{A.2}$$

Subtracting $\xi_{k+1}$ from $x_{k+1}$ in (A.2), we obtain

$$x_{k+1} - \xi_{k+1} = A_1(x_k - \xi_k)$$

that yields $x_k = \xi_k \quad \forall k \in \mathbb{Z}$ for any coinciding initial condition $x_{-\infty} = \xi_{-\infty}$, i.e. the second and third equations in (A.2) give the state-space realization at the right-hand side of (A.1).

**Michael Tchaikovsky, Ph.D. (Candidate of Sciences)**

Institute of Control Sciences, Russian Academy of Sciences
Laboratory 1 “Dynamics of control systems”
65 Profsoyuznaya str.
117997, Moscow, Russia
Tel.: +79151399341
E-mail mmtchaikovsky@hotmail.com

# Probabilistically Tuned LQ Control for Mechatronic Applications

Květoslav Belda

**Abstract**
Mechatronic applications are integral part of production machines and industrial robots. The key task is a design of their suitable control, which should ensure safe control actions in spite of sudden changes of working conditions. The paper presents specific probabilistic interpretation of well-known Linear Quadratic control.This interpretation employs complex information on system behavior and gives physical meaning for fine-tuning of control parameters. The principles of fully probabilistic design with emphasis on on-line tuning are demonstrated on physical model of gearbox mechatronic system representing flexible mechanism occurring in rolling machines.

**Keywords:** adaptive control, state-space realization, mechatronic applications

## Introduction

Mechatronic systems comprise elemental part of production machines and industrial robots. They consist of beams, wheels, joints and drives with power electronics. The systems have to be precisely controlled to provide safe motion and elimination of undesired vibrations causing drive wear and damage.

In this paper, the gearbox mechatronic system is used as a representative system. It represents flexible mechanism (Fig.1) occurring in rolling mill machines [3] and also in geared robot arms [8] of serial industrial robots - manipulators. Considered system consists of electric drive, solid wheels and elastic belts or elastic shafts respectively.

The aim is to tune suitably designed control, which should adapt itself for sudden changes of working conditions (load changes, external signal disturbances etc.) making control process stochastic.

The most general formulation of the control design is based on the minimization of expected value of a suitably chosen loss function. The loss function is defined as a function of system inputs, outputs and desired behavior with respect to feedback control strategies. The control strategy has to be chosen in correspondence to the purpose of control. One of well known powerful strategy is LQ (Linear Quadratic) control employing linear system model and quadratic criterion [2]. Its more general probabilistic interpretation [5] with emphasis on on-line parameter fine-tuning is presented here. The on-line tuning protects drives of controlled system from sharp actions induced by unpredicted change of working conditions.

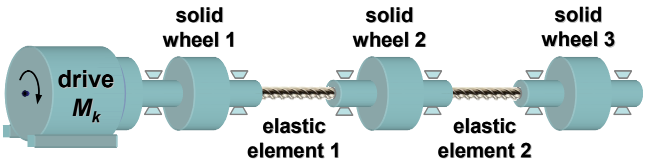


Fig.1 Scheme of gearbox mechatronic system

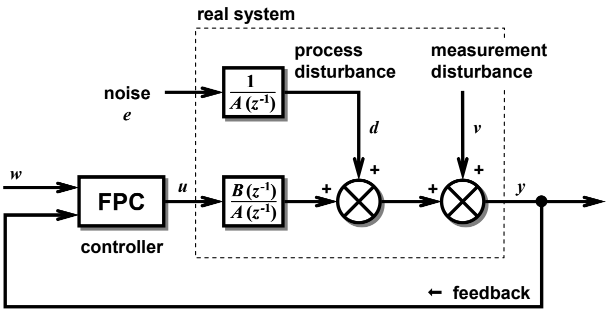


**Fig.2 Block diagram of the closed control loop of probabilistic controller and controlled system**

The proposed approach considers more complex information on controlled system behavior using probabilistic description of whole closed-loop, block diagram of which is shown in Fig.2. The diagram represents the structure of the closed-loop of considered mechatronic system.

In fully probabilistic approach, all available aspects of the closed-loop including expected and desired inputs and outputs, are defined as probability density functions. Consequently, the probabilistic interpretation may use more of available information contrary to standard design, which may have an insufficient number of representative parameters or interpretations for the information available.

In mechatronic systems (e.g. manipulators - robots [7], [8], the fully probabilistic approach offers to express stochastic inaccuracies of the mechanical elements (e.g. backlashes, friction, wear, elasticity etc.), actuators generating control actions and inaccuracies of measurement sensors and appropriate wiring (signal disturbances). Mechatronic systems represent a chain of different elements, which cause different inaccuracies, combination of which causes stochastic system behavior.

This paper is focused on probabilistic interpretation LQ control design as a promising approach regarding its tuning and application to mentioned mechatronic systems. It starts section, where the basic principles of fully probabilistic control design is briefly outlined. The following sections deal with definition of suitable models describing controlled systems and implementation issues. Then, the princip of on-line fully probabilistic control tuning is explained. At the end, proposed approach is demonstrated on physical model of flexible gearbox mechanism.

## Probabilistic design principles

The fully probabilistic control design determines admissible control strategy, which forces the joint distribution of all closed-loop variables as close as possible to the desired (ideal) distribution.linebreak To measure level of proximity of these distributions, the Kullback-Leibler divergence (*KL-divergence*) $D(f\,||\,{}^{I}f)$ is used as follows [4], [5].

$$D(f\,||\,{}^{I}f) \equiv E\left\{\ln\frac{f(X)}{{}^{I}f(X)}\right\} = \int f(X)\ln\frac{f(X)}{{}^{I}f(X)}dX \qquad (1)$$

where the pair of probability density functions (*pdf*s) $f$ and ${}^{I}f$ is considered to be acting on their domains i.e. on a set of all values $X^*$.

From control point of view, the *KL-divergence* represents the loss function or optimality criterion. By its minimization, the optimal control law is obtained. The following lines outline the mini-linebreak mization process. Due to necessity to consider time for computation of control law, the discrete design within finite time interval is considered.

### General assumptions

Let us start from explanation of pair of *pdf*s mentioned in (1), which are evaluated within some specific discrete-time interval. In control design, they represent joint *pdf*s of real and ideal closed-loop behavior:

- joint *pdf* of the real closed-loop behavior

$$f(X) = f_N \equiv f(\mathbf{x}_{k+N}, u_{k+N-1}, \cdots, u_k, \mathbf{x}_k) \qquad (2)$$

- joint *pdf* of the ideal closed-loop behavior

$$f(X) = f_N \equiv f(\mathbf{x}_{k+N}, u_{k+N-1}, \cdots, u_k, \mathbf{x}_k) \qquad (3)$$

These *pdf*s are considered to be defined for values in given time and their parameters to be valid within specific finite horizon $N$ called control horizon. The label $N$ represents the number of discrete time instants $j$ from instant $k$ within the horizon; i.e. $j = k+1, \cdots, k+N$; $u_{(\cdot)}$ are control actions.

Due to practical consequences, the *pdf*s are based on the assumption that succeeding system state $\mathbf{x}_j$ arises from previous system state $\mathbf{x}_{j-1}$ and system input $u_{j-1}$ only. Thus, $\mathbf{x}_j$ is independent of past system states and system inputs. This assumption is formulated as follows:

$$f(\mathbf{x}_j\,|\,\mathbf{x}_{j-1}, u_{j-1}, \cdots, \mathbf{x}_0, u_0) = f(\mathbf{x}_j\,|\,\mathbf{x}_{j-1}, u_{j-1}) \qquad (4)$$

$$f(u_j\,|\,\mathbf{x}_j, u_{j-1}, \cdots, \mathbf{x}_0, u_0) = f(u_j\,|\,\mathbf{x}_j) \qquad (5)$$

$${}^{I}f(\mathbf{x}_j\,|\,\mathbf{x}_{j-1}, u_{j-1}, \cdots, \mathbf{x}_0, u_0) = {}^{I}f(\mathbf{x}_j\,|\,\mathbf{x}_{j-1}, u_{j-1}) \qquad (6)$$

$${}^{I}f(u_j\,|\,\mathbf{x}_j, u_{j-1}, \cdots, \mathbf{x}_0, u_0) = {}^{I}f(u_j\,|\,\mathbf{x}_j) \qquad (7)$$

where *pdf* labeled by superscript $^{I}$ denote user requirements, i.e. user ideals.

Thus, the *pdf*s (2) and (3), or *pdf*s (4) to (7) respectively, describe real and ideal behavior of individual parts of given closed-loop i.e. behavior of the system and controller; e.g. in instant $j = k+1$, the real and ideal system behavior is modeled by *pdf*s $f(\mathbf{x}_{k+1}\,|\,\mathbf{x}_k, u_k)$ and ${}^{I}f(\mathbf{x}_{k+1}\,|\,\mathbf{x}_k, u_k)$; and real and ideal controller behavior is modeled by *pdf*s $f(u_k\,|\,\mathbf{x}_k)$ and ${}^{I}f(u_k\,|\,\mathbf{x}_k)$, respectively.

The suitable specification of individual *pdf*s will be described in implementation section.

### Task specification

The task of fully probabilistic control design is to determine optimal control law - optimal *pdf* ${}^{O}f(u_k\,|\,\mathbf{x}_k)$ of the *pdf* $f(u_k\,|\,\mathbf{x}_k)$:

$$\{{}^{O}f(u_{j-1}\,|\,x_{j-1})\}_{j=k+1}^{k+N} \in \underset{\{f(u_{j-1}\,|\,x_{j-1})\}_{j=k+1}^{k+N}}{\arg\min} D(f_N\,||\,{}^{I}f_N) \qquad (8)$$

As indicated in (8), the task of design consists in minimization of *KL-divergence*. The following subsection outlines the minimization procedure, which leads to the optimal *pdf* of controller and the optimal control law respectively.

### Outline of minimization procedure

This subsection presents a brief outline of minimization procedure only, detail derivation is described in [5]. Optimal *pdf* of the controller can be obtained using (8).

From control theory point of view, considering the assumptions from subsection of general asumptions, the equation (8) can be interpreted as expression of specific dynamic programming procedure [1].

$$\begin{aligned} \min_{\{f(u_{j-1}\,|\,x_{j-1})\}_{j=k+1}^{k+N}} D(f_N\,||\,{}^{I}f_N) &= \min_{\{f(u_{j-1}\,|\,x_{j-1})\}_{j=k+1}^{k+N}} E\left\{\sum_{j=k+1}^{k+N} z_j\right\} \\ = \min_{\{f(u_k\,|\,x_k)\}} E\left\{\sum_{j=k+1}^{k+N} z_j\right\} \cdots &= \min_{\{f(u_k\,|\,x_k)\}} \Big\{ E(z_{k+1}) + \cdots \\ \min_{\{f(u_{k+N-2}\,|\,x_{k+N-2})\}} \Big\{ E(z_{k+N-1}) &+ \min_{\{f(u_{k+N-1}\,|\,x_{k+N-1})\}} \{E(z_{k+N-1})\} \cdots \Big\} \end{aligned} \qquad (9)$$

where $z_j = \ln\dfrac{f_j(\mathbf{x}_j\,|\,\mathbf{x}_{j-1}, u_{j-1})}{{}^{I}f_j(\mathbf{x}_j\,|\,\mathbf{x}_{j-1}, u_{j-1})}$ is $j^{th}$ partial loss. The expression (9) leads to the following *pdf* of optimal control:

$${}^{O}f(u_k\,|\,\mathbf{x}_{kj}) = \frac{1}{\gamma(\mathbf{x}_k)}\,{}^{I}f(u_k\,|\,\mathbf{x}_k)\,e^{-\delta(u_k, \mathbf{x}_k)} \qquad (10)$$

where $\delta(u_k, \mathbf{x}_k)$ and $\gamma(\mathbf{x}_k)$ are suitably formed artificial quantities defined as follows

$$\delta(u_k, \mathbf{x}_k) = f(\mathbf{x}_{k+1}\,|\,u_k, \mathbf{x}_k)\,\ln\frac{f_j(\mathbf{x}_{k+1}\,|\,\mathbf{x}_k, u_k)}{{}^{I}f_j(\mathbf{x}_{k+1}\,|\,\mathbf{x}_k, u_k)}\,d\mathbf{x}_{k+1} \qquad (11)$$

$$\gamma(\mathbf{x}_k) = \int {}^{I}f(u_k\,|\,\mathbf{x}_k)\,e^{-\delta(u_k, \mathbf{x}_k)}\,du_k \qquad (12)$$

## Probabilistic model

As formerly mentioned, the system behavior can be described by probability density function (*pdf*). If the system behavior is normally distributed, then its *pdf* denoted by $f(y)$ is defined as follows

$$N(\mu_y, r_y) = f(y) = \frac{1}{\sqrt{2\pi r_y}} e^{-\frac{(y-\mu_y)^2}{2r_y}} \qquad (13)$$

where $\mu_y$ represents mean value, i.e. expected value of system output $y$ ($\mu_y = E\{y\}$), $\sigma_y^2 = r_y$ denotes a dispersion (variance; $r_y = E\{(y-\mu_y)^2\}$). In control design, these parameters are considered to be continuous in values and discrete in time. Their continuity follows from the system character. The discreteness in time is given by discrete realization of control, which naturally respects the time for its computation. Internal structure of parameters mentioned above can be specified in more detail either as ARX model or as state-space model. The ARX model [6] with normally distributed noise is defined as:

$$y_k = \sum_{i=1}^{n} b_i u_{k-i} - \sum_{i=1}^{n} a_i y_{k-i} + e_{y_k}, \quad e_{y_k} \sim N(0, r_y) \qquad (14)$$

where $n$ is an order and $e_{y_k}$ is a model noise, which has a dispersion $r_y$. The state-space model is defined as:

$$\mathbf{x}_{k+1} = \underbrace{\mathbf{A}\mathbf{x}_k + \mathbf{B}u_k}_{\mu_{\mathbf{x}}} + e_{\mathbf{x}_k}, \quad e_{\mathbf{x}_k} \sim N(0, \mathbf{R}) \qquad (15)$$

$$y_k = \mathbf{C}\mathbf{x}_k + \tilde{e}_{y_k}, \quad \tilde{e}_{y_k} \sim N(0, \tilde{r}_y) \qquad (16)$$

Equations (15) and (16) represent general state-space notation, in which the state $\mathbf{x}_k$ may belinebreak available or not; e.g. it has not a physical interpretation and for the control purposes it has to be estimated.

To avoid mentioned drawback, it is possible to use so-called pseudo state-space model [2], which is a direct reinterpretation of ARX model (14). Such reinterpretation means state-space model with non-minimal state, which contains only delayed values of inputs and outputs. An internal structure of the reinterpretation is de-fined as follows:

$$\mathbf{x}_k = \begin{bmatrix} u_{k-1} \\ \vdots \\ u_{k-n+1} \\ y_k \\ \vdots \\ y_{k-n+1} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & \cdots & 0 & 0 & \cdots & 0 \\ 1 & \ddots & \vdots & \vdots & & \vdots \\ 0 & \ddots & 0 & 0 & \cdots & 0 \\ b_2 & \cdots & b_n & -a_1 & \cdots & -a_n \\ 0 & \cdots & 0 & 1 & \cdots & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \end{bmatrix}, \qquad (17)$$

$$\mathbf{B} = \begin{bmatrix} 1 & \cdots & 0 & b_1 & \cdots & 0 \end{bmatrix}^T, \qquad (18)$$

$$\mathbf{C} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T, \quad \tilde{r}_y = 0 \qquad (19)$$

Relation of the pseudo-state space model to ARX model is obvious from the following corollary:

$$y_k = \underbrace{\mathbf{CAx}_{k-1} + \mathbf{CB}u_{k-1}}_{\mu_y} + \underbrace{\mathbf{Ce}_{\mathbf{x}_k}}_{e_{yk}}, \quad \mathbf{e}_{yk} \sim N(0, r_y), \qquad (20)$$

Models (14); or (15) and (16); or (15) to (19) are used as models for implementation of fully probabilistic control design described below.

## Implementation of control

Let us start from general expression (10) representing optimal *pdf* (section on principles). To compute real parameters of this *pdf*, individual *pdf*s from assumptions (4), (6) and (7) have to be defined. These emph{pdf}s represent both real and ideal behavior of closed-loop (Fig.2). Assuming model given by (15) to (19), i.e. finite memory and known parameters of appropriate distributions, then emph *pdf*s are defined as follows:

- *pdf* of the real controlled system output

$$N(\mu_y, r_y): f(y_{k+1} \mid u_k, \mathbf{x}_k) = \frac{1}{\sqrt{2\pi\, r_y}} e^{-\frac{1}{2}(y_{k+1}-\mu_y)^T r_y^{-1} (y_{k+1}-\mu_y)} \qquad (21)$$

- *pdf* of the ideal controlled system output

$$N({}^I\mu_y, {}^Ir_y): {}^If(y_{k+1} \mid u_k, \mathbf{x}_k) = \\ = \frac{1}{\sqrt{2\pi\, {}^Ir_y}} e^{-\frac{1}{2}(y_{k+1}-{}^I\mu_y)^T\, {}^Ir_y^{-1} (y_{k+1}-{}^I\mu_y)} \qquad (22)$$

where ideal ${}^I\mu_y$ is the desired output value $w_{k+1}$;

- *pdf* of the ideal controlled system input

$$N({}^I\mu_u, {}^Ir_u): {}^If(u_k \mid \mathbf{x}_k) = \\ = \frac{1}{\sqrt{2\pi\, {}^Ir_u}} e^{-\frac{1}{2}(u_{k+1}-{}^I\mu_u)^T\, {}^Ir_u^{-1} (u_{k+1}-{}^I\mu_u)} \qquad (23)$$

where ${}^I\mu_u$ is assumed to be the previous action $u_{k-1}$ and the dispersion ${}^Ir_u$ can be viewed as a tuning parameter of the controller. For *pdf*s defined like that, the computation of *pdf* (10) leads to the following expressions:

$${}^Of(u_k \mid \mathbf{x}_k, u_{k-1}) = \frac{1}{\sqrt{2\pi\, {}^Or_u}} e^{-\frac{1}{2}(u_k - {}^Or_u\, b)^T\, {}^Or_u^{-1}(u_k - {}^Or_u\, b)} \\ = \frac{1}{\sqrt{2\pi\, {}^Or_u}} e^{-\frac{1}{2}\left\{u_k + \mathbf{k}_k\mathbf{x}_k - \sum_{j=k+1}^{k+N+1} k_{wj} w_j - k_u u_{k-1}\right\}^2} \qquad (24)$$

$${}^Ou_k = -\mathbf{k}_k\mathbf{x}_k + \sum_{j=k+1}^{k+N+1} k_{wj} w_j + k_u u_{k-1} \qquad (25)$$

where ${}^Ou_k$ is the optimal control law.

## On-line probabilistic tuning

This section focuses on tuning of control parameters. In general, the parameters of the controllers determine the character of the control actions responding on changes of working conditions and user requirements. Usually, the parameters - their values - are selected according to user experiences or according to some simple empirical rule. The values are constant for whole control process or sometimes they are discontinuously reset. It is not suitable for dynamic systems within changeable environment.

Presented probabilistic formulation of LQ control is suitable for on-line tuning or fine-tuning. Partly, it can use local consecutively-changed models (model adaptation) and partly, can use different slightly-changed control parameters (controller adaptation). The former can be characterized as some change of system properties i.e. model parameters and the latter can represent the quality of the description i.e. quality of the model parameters. Thus, good reliable model gives more accurate and brisk controller and vice versa.

The both mentioned ways of adaptation can be covered in the control law (25), which represents standard form of LQ control. The gains $\mathbf{k}_k$, $k_{wj}$ and $k_u$ contain parameters of the system model (e.g. model (20)) and simultaneously control parameters, which are presented by dispersions ${}^{I}r_u$ and ${}^{I}r_y$. These dispersions are very important, because they are determining factors for the gains $\mathbf{k}_k$, $k_{wj}$ and $k_u$ in (24) and (25).

In comparison with non-probabilistic LQ control design, reciprocal values of the dispersions represent input and output penalization factors ($q_u = {}^{I}r_u$, $q_y = {}^{I}r_y$), which together adjust individual terms in quadratic loss-function.

As was already mentioned, their choice is based on experience or on experimental tuning. In fully probabilistic control design, interpretation of these quantities is more straightforward. The equations (22) and (23) imply that ${}^{I}r_u$ and ${}^{I}r_y$ represent noise dispersions for ideal distribution of the system and controller.

The algorithm proposed in this paper is intended for systems (e.g. mechatronic one), where the mathematical model together with the noise can change substantially, possibly due to additional interference, that may occur randomly during the control. Inadequate choice of input and output penalizations or ${}^{I}r_u$ with ${}^{I}r_y$ respectively, can cause serious device failures, e.g. system actuators (drives) might not be able to achieve designed control or may be damaged by them.

Unexpected system noise increase may force the controller to generate inputs out of any reasonable physical range of the device. In such undesired cases, it would usually be acceptable to decrease control quality in order to achieve at least some reasonable control actions. Probabilistic control interpretation of penalization factors as dispersions can achieve indicated strategy via on-line control tuning.

The tuning is based on the idea of changing of dispersion ${}^{I}r_y$ so that its amplitude is propor-tional to the output dispersion $r_y$ or practically to its estimate

$$\hat{r}_{yi} = e_{yi}\, e_{yi}^{T} = (y_i - \hat{\mu}_{yi})\,(y_i - \hat{\mu}_{yi})^{T} \tag{26}$$

calculated from current data $y_i$ and model. The effect is that during periods of increased output noise, output ideal is set to be less strict. It causes the output to be tracked less closely. This allows the input to stay in its reasonable constraints.

However, current output dispersionlinebreak can change very quickly causing big changes in dispersion ${}^{I}r_y$. In order to avoid this, $\hat{r}_{yi}$ has to be filtrated. As a suitable filter, exponential forgetting is used. It can be defined as follows:

$$\tilde{r}_{y1} = (1-\lambda)\,\hat{r}_{y1} \tag{27}$$

$$\tilde{r}_{yi} = \lambda\,\tilde{r}_{yi-1} + (1-\lambda)\,\hat{r}_{yi}, \quad i = 2,\cdots,k \tag{28}$$

where $\lambda$ is a forgetting factor influencing quickness of weight decrease of individual contributions $\hat{r}_{yi}$. The equations (27) and (28) can form one general expression:

$$\tilde{r}_{yk} = (1-\lambda)\sum_{i=1}^{k}\lambda^{k-i}\,\hat{r}_{yi} \tag{29}$$

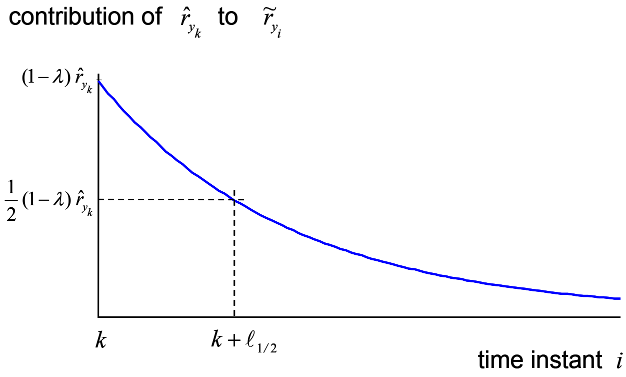


**Fig.3 Trend of contribution of $\hat{r}_{yk}$ to $\tilde{r}_{yi}$**

In order to find reasonable value for parameter $\lambda$ the suitable number of time instants $\ell$ has to be defined in correspondence to the characterlinebreak of control process During these $\ell$ time instants, the contribution of $\hat{r}_{yk}$ to $\hat{r}_{yi}$ drops to the given level.

Standard choice is to select number of instants (denoted by $\ell_{1/2}$) that cause dropping the contribution of $\hat{r}_{yk}$ to one half of the original value. It implies that $\ell_{1/2}$ satisfies the equation:

$$\lambda^{\ell_{1/2}}(1-\lambda)\,\hat{r}_{yk} = \frac{1}{2}(1-\lambda)\,\hat{r}_{yk} \tag{30}$$

See Fig.3 for illustration of this effect. Producing `half-time' $\ell_{1/2}$ is user-friendly way to find a suitable value for constant $\lambda$, because user can easily imagine what is the time needed for a contribution of $\hat{r}_{yk}$ to drop to one half. Consequently, suitable $\lambda$ can be found like this:

$$\lambda = \left(\frac{1}{2}\right)^{1/\ell_{1/2}} \tag{31}$$

where $\ell_{1/2}$ is provided by the user.

## On-line tuned LQ control of gearbox mechatronic system

This section demonstrates the presented fully probabilistic interpretation of LQ control design including the on-line parameter tuning. The aim is to illustrate improvements of control process that follow from consequences of previous section.

As was mentioned in introduction, the gearbox system (see Fig.4(e)), consists of three wheels, which are mutually connected by two elastic belts. Position of the wheel 1 is controlled by servo-motor, and the position of the wheel 3 is measured.

From control design point of view, the mechatronic system is modelled by ARX model (14) of order $n=6$, which is determined by the fackt, that each solid wheel represents approximatelly 2 orders. Real control of the system is provided by adaptive LQ controller.

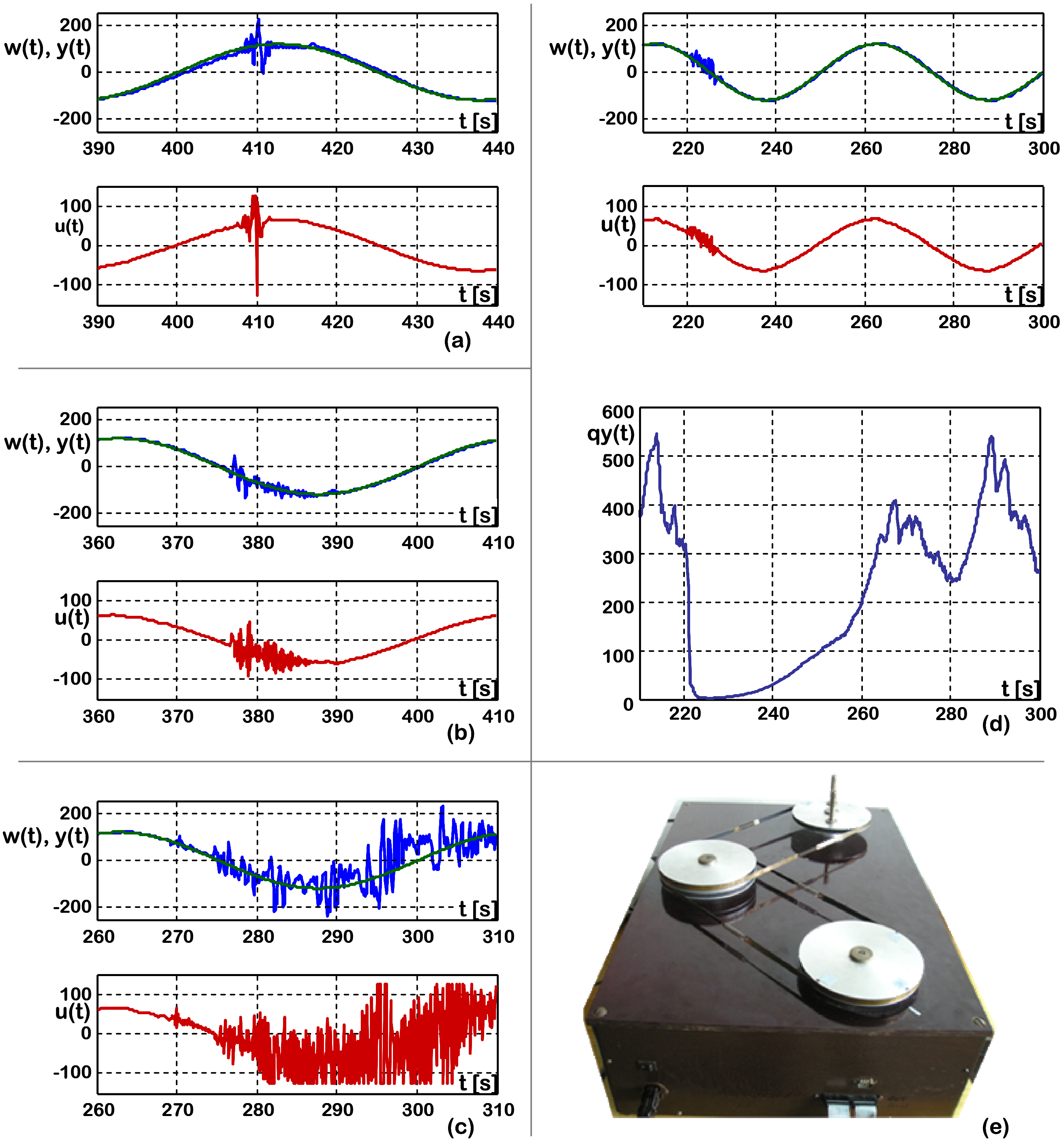


**Fig.4 Real experiment: (a), (b) and (c) comparison of standard LQ control $q_y = 1$, $q_y = 100$ and $q_y = 200$ respectively; (d) control generated by probabilistic design with tuning (desired and real system output $w(t)$ and $y(t)$; input $u(t)$; penalization $q_y(t)$; (e) gearbox system**

During control process, the discrepancy between model estimated and the real system occurs. This causes sharp changes of control actions, which do not follow from desired profile of system output but just from temporal discrepancy of estimated model from reality. In ideal conditions, this undesirable state damp out shortly.

However, in real conditions, it can cause unpredictable behaviour damaging drives and even it can damage other structural elements of the system. This phenomenon is being suppressed by tuning algorithm proposed in this paper (see Fig.4(d)).

Fig.4 specifically, demonstrates four runs of real control process. The individual sub-figures (a), (b), and (c) show control runs with differentlinebreak but constant output penalization ($q_y$). In all caseslinebreak of constant $q_y$, the input magnitude startslinebreak to change rapidly due to sudden disturbance. The process eventually stabilizes, however, in case (c) the controller have not stabilized at all.

With adaptive tuning proposed in this paper (sub-figure (d) of Fig.4) the changes in input are reasonably small, moreover, the output matches desired value much better.

## Conclusion

The paper outlines the principles and practical aspects of fully probabilistic interpretation of LQ control. Consequently, the on-line tuning was introduced. This way of design forms sound physical interpretation for tunable controller parameters. The design with tuning was applied and demonstrated on real gearbox mechatronic system occuring frequenty in production machines (e.g. rolling mills) and in industrial robots (geared robotic arms). The representative results are discussed in section dealing with real time control of gearbox mechatronic system.

## References

[1] BERTSEKAS, D.: Dynamic programming and optimal control, second ed. Athena Scientific, Nashua, US, New York, 2001.

[2] BOBÁL, V., BÖHM, J., FESSL, J., MACHÁČEK, J.: Digital Self-tuning Controllers. Springer, 2005. ISBN 1-85233-980-2.

[3] ETTLER, P., KÁRNÝ, M., GUY, T.: Bayes for rolling mills: From parameter estimation to decision support. In Proc. of the 16th World Congress of the IFAC, pages 1–6. IFAC, Prague, July 2005.

[4] KÁRNÝ, M.: Towards fully probabilistic control design. Automatica, 32(12):1719–1722, 1996.

[5] KÁRNÝ, M., BÖHM, J., GUY, T., JIRSA, L., NAGY, I., NEDOMA, P., TESAŘ, L.: Optimized Bayesian Dynamic Advising, Theory and Algorithms. Springer, 2006. ISBN 978-1-85233-928-9.

[6] PETERKA, V.: Trends and Progress in System Identification, chapter Bayesian Approach to System Identification. Pergamon Press, Oxford, 1981.

[7] TSAI, L.-W.: Robot Analysis: The Mechanics of Serial and Parallel Manipulators. John Wiley Inc., New York, 1999. ISBN 0-471-32593-7.

[8] WERNHOLT, E., GUNNARSSON, S.: Nonlinear identification of a physically parameterized robot model. In Proceedings of the 14th IFAC Symposium on System Identification, pages 143–148. Newcastle, Australia, March 2006.

**Ing. Květoslav Belda, Ph.D.**

Institute of Information Theory and Automation
Department of Adaptive Systems
Pod Vodárenskou věží 4
182 08 Praha 8
Czech Republic
E-mail: belda@utia.cas.cz

# Exponential Stability of Networked Control Systems with Random Delays

Dušan Krokavec

**Abstract**
In this paper, the problem of exponential stability for the standard form of state control, realized in a networked control system structure, is studied. To deal with the problem of exponential stability analysis of the event-time-driven modes in the networked control systems the delayed-dependent exponential stability condition are proven and actualized. Based on the delay-time dependent Lyapunov-Krasovskii functional the linear matrix inequalities for stability conditions are new formulated. Since presented method can use bilinear matrix inequality techniques it is computationally enough efficient and extended.

**Keywords:** Networked systems, stability analysis, time-delay systems, linear matrix inequality, state feedback.

## Introduction

Recent advances in the communication technology lead to an increased use of the networked control. The networked control systems (NCS) are control loops closed through a shared communication network, where the network between control system components is used to exchange the information and control signals. The advantages of such structure are most of all simple installation, maintenance and system volume, as well as increased system agility. However, due to communication channel insertion, induced delay and packet dropout may seriously deteriorate the performance of the system, especially its stability.

During the previous decade, the stability problem of the networked control systems with random delays has attracted a lot of attention. The main approach for the stability analysis relies on Lyapunov--Krasovskii functional and on LMI approaches for constructing common Lyapunov function. For the reason of the network-induced delays it is often assumed that the actuator and the controller are event driven, but once the large delay bound appears, the system may become unstable. The usual approach ignores in the controller design stage the network delays [13] and in the next design step is analyzed stability, performance and robustness with respect to the effects and the characteristics of the network delays and the scheduling policy. Progress review in this research field one can find e.g. in [3], [10], [20], and the references therein.

This paper is concerned with the problem of the event-time-driven mode in the networked control systems. Under this mode in a critical event a switched delay system structure is occasioned, which may include an unstable subsystem. The paper actualizes, completes and extends the basic idea presented in [16] in coincidence with [19] to obtain conditions for the exponential stability of such structure. Possibly time-varying delay is considered and main attention is focused on the linear matrix inequalities (LMIs) which have to hold to obtain the control process exponentially stable. The presented LMI approach is computationally efficient as it can be solved numerically (see e.g. [1], [8]), and is based on Lyapunov-Krasovskii functional (see e.g. [4], [6]) and Leibniz-Newton formula to eliminate some dead-time dependent terms [12].

## 1. Problem Description

Through this paper the task is concerned with the stability analysis of NCS, where a linear dynamic system given by the set of equations

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{q}(t) \tag{2}$$

is controlled by the delayed state feedback. Here $\boldsymbol{q}(t) \in \mathbf{R}^{n}$, $\boldsymbol{u}(t) \in \mathbf{R}^{r}$ and $\boldsymbol{y}(t) \in \mathbf{R}^{m}$ are vectors of the state, input and measurable output variables, respectively, and the system parameters $\boldsymbol{A} \in \mathbf{R}^{n \times n}$, $\boldsymbol{B} \in \mathbf{R}^{n \times r}$ and $\boldsymbol{C} \in \mathbf{R}^{m \times n}$ are real matrices. Problem of the interest is to design the stable NCS with the memory less linear state feedback controller of the form

$$\boldsymbol{u}(t) = \boldsymbol{K}\boldsymbol{q}(t) \tag{3}$$

where matrix $\boldsymbol{K} \in \mathbf{R}^{r \times n}$ is the controller gain matrix. Accepting a network delay-time, the event-time-driven closed-loop system (1), (2) takes the form

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{K}\boldsymbol{q}(i_k \Delta t), \quad t \in \langle i_k \Delta t + \tau_k, j_k) \tag{4}$$

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t), \quad t \in \langle j_k, i_{k+1}\Delta t + \tau_{k+1}) \tag{5}$$

where $i_k$: $k$ = 1,2,... are some integers, $\Delta t$ is the sampling period, and $\tau_k \geq 0$ is the time delay, which denotes the time interval from the instant time $i_k \Delta t$, where the sensors notes the sample data from the plat to the instant time when the actuators transfer data to the plant.

In the next it is supposed that this condition is satisfied

$$j_k = \begin{cases} i_{k+1}\Delta t + \tau_{k+1}, & (i_{k+1} - i_k)\Delta t + \tau_{k+1} \leq h \\ i_k \Delta t + h, & (i_{k+1} - i_k)\Delta t + \tau_{k+1} > h \end{cases} \tag{6}$$

Event-time-driven mode means, that the controller and the actuators data will be updated once a new packet comes, and these new data can be held during the intervening time less then $h$. If at the end of this time interval the new data packet has not yet come, acting data will be set to zero and they will stay zero until the new data will come. By this rule obtained the switched delay system may include an unstable subsystem (see [16]).

## 2. Basic Preliminaries

### 2.1 Schur Complement

The nonlinear convex inequalities can be converted to LMI form using Schur complements. Let a linear matrix inequality takes form

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & -\boldsymbol{R} \end{bmatrix} < 0, \quad \boldsymbol{Q} = \boldsymbol{Q}^T > 0, \boldsymbol{R} = \boldsymbol{R}^T > 0 \tag{7}$$

Using Gauss elimination it yields

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{SR}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & -\boldsymbol{R} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{R}^{-1}\boldsymbol{S}^T & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}+\boldsymbol{SR}^{-1}\boldsymbol{S}^T & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{R} \end{bmatrix} \tag{8}$$

Since

$$\det \begin{bmatrix} \boldsymbol{I} & \boldsymbol{SR}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = 1 \tag{9}$$

and $\boldsymbol{I}$ is the identity matrix of appropriate dimension, with this transform the negativity of (7) is not changed, i.e. this follows as a consequence

$$\begin{gathered} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & -\boldsymbol{R} \end{bmatrix} < 0 \Leftrightarrow \begin{bmatrix} \boldsymbol{Q}+\boldsymbol{SR}^{-1}\boldsymbol{S}^T & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{R} \end{bmatrix} < 0 \\ \Updownarrow \\ \boldsymbol{Q}+\boldsymbol{SR}^{-1}\boldsymbol{S}^T < 0, \boldsymbol{R} > 0 \end{gathered} \tag{10}$$

respectively. As one can see, this complement offer possibility to rewrite the nonlinear inequalities in a closed matrix LMI form (see e.g. [1], [8]).

### 2.2 Zero Complement

Since Leibniz-Newton formula

$$\int_{t-\tau}^{t} \dot{\boldsymbol{x}}(r)\,\mathrm{d}r = \boldsymbol{x}(t) - \boldsymbol{x}(t-\tau) \tag{11}$$

implies

$$\boldsymbol{x}(t) - \boldsymbol{x}(t-\tau) - \int_{t-\tau}^{t} \dot{\boldsymbol{x}}(r)\,\mathrm{d}r = \boldsymbol{0} \tag{12}$$

it is evident that for any nonzero matrix $\boldsymbol{W}$ of appropriate dimension it is true

$$\boldsymbol{z}^T(t)\boldsymbol{W}[\boldsymbol{x}(t) - \boldsymbol{x}(t-\tau) - \int_{t-\tau}^{t} \dot{\boldsymbol{x}}(r)\,\mathrm{d}r] = 0 \tag{13}$$

where $\boldsymbol{z}(t)$ is an arbitrary vector (see [7], [16]).

### 2.3 Symmetric Upper-bounds Inequality

Let $f(\boldsymbol{x}(r), v),\ \boldsymbol{x}(r) \in \mathbf{R}^n,\ \boldsymbol{X} > 0,\ \boldsymbol{X} \in \mathbf{R}^{n \times n},\ a > 0,\ a \in \mathbf{R},$ is any real positive definite and integrable vector function of the form

$$f(\boldsymbol{x}(r), v) = \boldsymbol{x}^T(r)\,\mathrm{e}^{av}\,\boldsymbol{Xx}(r) \tag{14}$$

such, that there exists well defined integration as following

$$\int_{-b}^{0}\int_{t+v}^{t} f(\boldsymbol{x}(r), v)\,\mathrm{d}r\,\mathrm{d}v > 0 \tag{15}$$

where $b \geq 0, b \in \mathbf{R}, t \in \langle 0, \infty)$.

Since for (14) one can write

$$\boldsymbol{x}^T(r)\,\mathrm{e}^{av}\,\boldsymbol{Xx}(r) - \boldsymbol{x}^T(r)\,\mathrm{e}^{av}\,\boldsymbol{Xx}(r) = 0 \tag{16}$$

thus, by Schur complement, it is true, that

$$\begin{bmatrix} \boldsymbol{x}^T(r)\,\mathrm{e}^{av}\,\boldsymbol{Xx}(r) & \boldsymbol{x}^T(r) \\ \boldsymbol{x}(r) & \mathrm{e}^{-av}\,\boldsymbol{X}^{-1} \end{bmatrix} \geq 0 \tag{17}$$

and the double integration of (17) leads to

$$\begin{bmatrix} \int_{-b}^{0}\int_{t+v}^{t} \boldsymbol{x}^T(r)\,\mathrm{e}^{av}\,\boldsymbol{Xx}(r)\,\mathrm{d}r\,\mathrm{d}v & \int_{-b}^{0}\int_{t+v}^{t} \boldsymbol{x}^T(r)\,\mathrm{d}r\,\mathrm{d}v \\ \int_{-b}^{0}\int_{t+v}^{t} \boldsymbol{x}(r)\,\mathrm{d}r\,\mathrm{d}v & \int_{-b}^{0}\int_{t+v}^{t} \mathrm{e}^{-av}\,\boldsymbol{X}^{-1}\,\mathrm{d}r\,\mathrm{d}v \end{bmatrix} \geq 0 \tag{18}$$

Then with

$$\int_{t+v}^{t} \mathrm{e}^{-av}\,\boldsymbol{X}^{-1}\,\mathrm{d}r = -v\,\mathrm{e}^{-av}\,\boldsymbol{X}^{-1} \tag{19}$$

as well as with

$$\begin{aligned} \int_{-b}^{0} -v\,\mathrm{e}^{-av}\,\boldsymbol{X}^{-1}\,\mathrm{d}v = \tfrac{v}{a}\mathrm{e}^{-av}\,\boldsymbol{X}^{-1}\Big|_{-b}^{0} - \int_{-b}^{0} \tfrac{1}{a} v\,\mathrm{e}^{-av}\,\boldsymbol{X}^{-1}\,\mathrm{d}v = \\ = \tfrac{1}{a^2}(va+1)\,\mathrm{e}^{-av}\,\boldsymbol{X}^{-1}\Big|_{-b}^{0} = c^{-1}\boldsymbol{X}^{-1} \end{aligned} \tag{20}$$

for

$$c^{-1} = \tfrac{1}{a^2}(1 + ab\,\mathrm{e}^{ab} - \mathrm{e}^{-ab}) \tag{21}$$

inequality (18) can be rewritten as

$$\begin{bmatrix} \int_{-b}^{0}\int_{t+v}^{t} \boldsymbol{x}^T(r)\,\mathrm{e}^{av}\,\boldsymbol{Xx}(r)\,\mathrm{d}r\,\mathrm{d}v & \int_{-b}^{0}\int_{t+v}^{t} \boldsymbol{x}^T(r)\,\mathrm{d}r\,\mathrm{d}v \\ * & c^{-1}\boldsymbol{X}^{-1} \end{bmatrix} \geq 0 \tag{22}$$

It is evident, that (22) implies

$$\begin{aligned} \int_{-b}^{0}\int_{t+v}^{t} \boldsymbol{x}^T(r)\,\mathrm{e}^{av}\,\boldsymbol{Xx}(r)\,\mathrm{d}r\,\mathrm{d}v \geq \\ \geq \left[\int_{-b}^{0}\int_{t+v}^{t} \boldsymbol{x}^T(r)\,\mathrm{d}r\,\mathrm{d}v\right]^T c\boldsymbol{X}\left[\int_{-b}^{0}\int_{t+v}^{t} \boldsymbol{x}(r)\,\mathrm{d}r\,\mathrm{d}v\right] \end{aligned} \tag{23}$$

(see e.g. [16]). Hereafter $*$ denotes the symmetric item in a symmetric matrix.

## 3. Exponential Stability of the Autonomous System

Defining the delay-dependent Lyapunov-Krasovskii functional as follows

$$v(\boldsymbol{q}(t)) = \boldsymbol{q}^T(t)\boldsymbol{Pq}(t) + \int_{-h}^{0}\int_{t+v}^{t} \dot{\boldsymbol{q}}^T(r)\,\mathrm{e}^{\alpha_1(r-t)}\,\boldsymbol{R}\dot{\boldsymbol{q}}(t)\,\mathrm{d}r\,\mathrm{d}v > 0 \tag{24}$$

where $\boldsymbol{P} = \boldsymbol{P}^T > 0, \boldsymbol{R} = \boldsymbol{R}^T > 0$ and evaluating derivative of this functional one obtains

$$\begin{aligned} \dot{\mathrm{v}}(\boldsymbol{q}(t)) = \dot{\boldsymbol{q}}^T(t)\boldsymbol{Pq}(t) + \boldsymbol{q}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t) + \\ + h\int_{t}^{t} \dot{\boldsymbol{q}}^T(r)\,\mathrm{e}^{\alpha_1 0}\,\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r - \int_{t-h}^{t} \dot{\boldsymbol{q}}^T(r)\,\mathrm{e}^{-\alpha_1 h}\,\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r - \\ -\alpha_1 \int_{-h}^{0}\int_{t+v}^{t} \dot{\boldsymbol{q}}^T(r)\,\mathrm{e}^{\alpha_1(r-t)}\,\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r\,\mathrm{d}v < 0 \end{aligned} \tag{25}$$

$$\begin{aligned}\dot{\mathrm{v}}(\boldsymbol{q}(t)) &= \dot{\boldsymbol{q}}^T(t)\boldsymbol{P}\boldsymbol{q}(t)+\boldsymbol{q}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t)+h\dot{\boldsymbol{q}}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t)-\\ &-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r-\\ &-\alpha_1\int_{-h}^{0}\int_{t+v}^{t}\dot{\boldsymbol{q}}^T(r)\mathrm{e}^{\alpha_1(r-t)}\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r\,\mathrm{d}v<0\end{aligned} \tag{26}$$

respectively. Therefore, this follows as a consequence

$$\begin{aligned}&\dot{\mathrm{v}}(\boldsymbol{q}(t))-\alpha_2\,\mathrm{v}(\boldsymbol{q}(t))\le\\ &\boldsymbol{q}^T(t)(\boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}+h\boldsymbol{A}^T\boldsymbol{R}\boldsymbol{A}-\alpha_2\boldsymbol{P})\boldsymbol{q}(t)-\\ &-(\alpha_1+\alpha_2)\dot{\mathrm{v}}^{\circ}(\boldsymbol{q}(t))-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r<0\end{aligned} \tag{27}$$

where

$$\dot{\mathrm{v}}^{\circ}(\boldsymbol{q}(t))=\left[\int_{-h}^{0}\int_{t+v}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}r\,\mathrm{d}v\right]^T c\boldsymbol{R}\left[\int_{-h}^{0}\int_{t+v}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r\,\mathrm{d}v\right] \tag{28}$$

$$c^{-1}=\tfrac{1}{\alpha_1^2}(1+\alpha_1 h\,\mathrm{e}^{\alpha_1 h}-\mathrm{e}^{-\alpha_1 h}) \tag{29}$$

Since it can be written

$$\begin{aligned}&\int_{-h}^{0}\int_{t+v}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r\,\mathrm{d}v=\int_{-h}^{0}(\boldsymbol{q}(t)-\boldsymbol{q}(t+v))\,\mathrm{d}v=\\ &=h\boldsymbol{q}(t)-\int_{-h}^{0}\boldsymbol{q}(t+v)\,\mathrm{d}v=h\boldsymbol{q}(t)-\int_{t-h}^{t}\boldsymbol{q}(r)\,\mathrm{d}r\end{aligned} \tag{30}$$

then

$$\dot{\mathrm{v}}^{\circ}(\boldsymbol{q}(t))=\left[h\boldsymbol{q}(t)-\int_{t-h}^{t}\boldsymbol{q}(r)\,\mathrm{d}r\right]^T c\boldsymbol{R}\left[h\boldsymbol{q}(t)-\int_{t-h}^{t}\boldsymbol{q}(r)\,\mathrm{d}r\right] \tag{31}$$

and using notation

$$\boldsymbol{p}^T(t)=[\boldsymbol{q}^T(t)\;\;\boldsymbol{q}^T(t-h)\;\;\int_{t-h}^{t}\boldsymbol{q}^T(r)\,\mathrm{d}r] \tag{32}$$

equality (31) can be rewritten as

$$\dot{\mathrm{v}}^{\circ}(\boldsymbol{q}(t))=\boldsymbol{p}^T(t)\begin{bmatrix}h\\0\\-1\end{bmatrix}c\boldsymbol{R}\begin{bmatrix}h\,0\,{-1}\end{bmatrix}\boldsymbol{p}(t) \tag{33}$$

Using (32) by the same way constraint (13) can be adapted for solution in the next form

$$\begin{aligned}&\boldsymbol{p}^T(t)\boldsymbol{W}[\boldsymbol{q}^T(t)-\boldsymbol{q}^T(t-h)-\int_{t-h}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r]+\\ &[\boldsymbol{q}^T(t)-\boldsymbol{q}^T(t-h)-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}r]\boldsymbol{W}^T\boldsymbol{p}(t)=0\end{aligned} \tag{34}$$

$$\begin{aligned}&\boldsymbol{p}^T(t)\boldsymbol{W}\begin{bmatrix}1\,{-1}\,0\end{bmatrix}\boldsymbol{p}(t)+\boldsymbol{p}^T(t)\begin{bmatrix}1\,{-1}\,0\end{bmatrix}^T\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{p}(t)-\\ &-\boldsymbol{p}^T(t)\boldsymbol{W}\int_{t-h}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}r]\boldsymbol{W}^T\boldsymbol{p}(t)=0\end{aligned} \tag{35}$$

respectively, where

$$\boldsymbol{W}^T=\begin{bmatrix}\boldsymbol{W}_1^T\,\boldsymbol{W}_2^T\,\boldsymbol{W}_3^T\end{bmatrix} \tag{36}$$

Therefore, with (33) and (36) inequality (27) takes form

$$\begin{aligned}&\dot{\mathrm{v}}(\boldsymbol{q}(t))-\alpha_2\,\mathrm{v}(\boldsymbol{q}(t))\le\boldsymbol{q}^T(t)\boldsymbol{S}^{\circ}\boldsymbol{q}(t)+\boldsymbol{p}^T(t)\boldsymbol{T}^{\circ}\boldsymbol{p}(t)\\ &\quad-\boldsymbol{p}^T(t)\boldsymbol{W}\int_{t-h}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}r]\boldsymbol{W}^T\boldsymbol{p}(t)-\\ &\qquad-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r<0\end{aligned} \tag{37}$$

where

$$\boldsymbol{S}^{\circ}=\boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}+h\boldsymbol{A}^T\boldsymbol{R}\boldsymbol{A}-\alpha_2\boldsymbol{P} \tag{38}$$

$$\boldsymbol{T}^{\circ}=-(\alpha_1+\alpha_2)\begin{bmatrix}h\\0\\-1\end{bmatrix}c\boldsymbol{R}\begin{bmatrix}h\,0\,{-1}\end{bmatrix}+\boldsymbol{W}\begin{bmatrix}1\,{-1}\,0\end{bmatrix}+\begin{bmatrix}1\\-1\\0\end{bmatrix}\boldsymbol{W}^{\boldsymbol{T}} \tag{39}$$

Since

$$\boldsymbol{q}^T(t)\boldsymbol{S}^{\circ}\boldsymbol{q}(t)=\boldsymbol{p}^T(t)\boldsymbol{T}_1^{\circ}\boldsymbol{p}(t) \tag{40}$$

with

$$\boldsymbol{T}_1^{\circ}=\begin{bmatrix}\boldsymbol{S}^{\circ} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} \tag{41}$$

the extended weighting matrix takes structure

$$\boldsymbol{T}^{\circ}=\begin{bmatrix}-h^2d\boldsymbol{R} & \boldsymbol{0} & hd\boldsymbol{R}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}\\ hd\boldsymbol{R} & \boldsymbol{0} & -d\boldsymbol{R}\end{bmatrix}+\begin{bmatrix}\boldsymbol{W}_1+\boldsymbol{W}_1^T & -\boldsymbol{W}_1+\boldsymbol{W}_2^T & \boldsymbol{W}_3^T\\ \boldsymbol{W}_2-\boldsymbol{W}_1^T & -\boldsymbol{W}_2-\boldsymbol{W}_2^T & -\boldsymbol{W}_3^T\\ \boldsymbol{W}_3 & \boldsymbol{W}_3 & \boldsymbol{0}\end{bmatrix} \tag{42}$$

where

$$d=(\alpha_1+\alpha_2)c \tag{43}$$

Thus, it is evident that

$$\begin{aligned}&\dot{\mathrm{v}}(\boldsymbol{q}(t))-\alpha_2\,\mathrm{v}(\boldsymbol{q}(t))\le\boldsymbol{p}^T(t)\boldsymbol{T}^{\bullet}\boldsymbol{p}(t)-\boldsymbol{p}^T(t)\boldsymbol{W}\int_{t-h}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r-\\ &\quad-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}r\,\boldsymbol{W}^T\boldsymbol{p}(t)-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r<0\end{aligned} \tag{44}$$

where

$$\boldsymbol{T}^{\bullet}=\boldsymbol{T}^{\circ}+\boldsymbol{T}_1^{\circ}=\begin{bmatrix}\boldsymbol{S} & \boldsymbol{W}_2^T-\boldsymbol{W}_1 & \boldsymbol{W}_3^T-hd\boldsymbol{R}\\ * & -\boldsymbol{W}_2-\boldsymbol{W}_2^T & -\boldsymbol{W}_3^T\\ * & * & -d\boldsymbol{R}\end{bmatrix} \tag{45}$$

$$\boldsymbol{S}=\boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}+h\boldsymbol{A}^T\boldsymbol{R}\boldsymbol{A}-\alpha_2\boldsymbol{P}-h^2d\boldsymbol{R}+\boldsymbol{W}_1+\boldsymbol{W}_1^T \tag{46}$$

Analogously to (17), it yields

$$\begin{bmatrix}\dot{\boldsymbol{q}}^T(r)\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r) & \dot{\boldsymbol{q}}^T(r)\\ \dot{\boldsymbol{q}}(r) & \mathrm{e}^{\alpha_1 h}\boldsymbol{R}^{-1}\end{bmatrix}\ge 0 \tag{47}$$

and since

$$\int_{t-h}^{t}\mathrm{e}^{\alpha_1 h}\boldsymbol{R}^{-1}\,\mathrm{d}r=h\,\mathrm{e}^{\alpha_1 h}\boldsymbol{R}^{-1} \tag{48}$$

the following is obtained

$$\begin{bmatrix}\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r & \int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}r\\ \int_{t-h}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r & h\,\mathrm{e}^{\alpha_1 h}\boldsymbol{R}^{-1}\end{bmatrix}\ge 0 \tag{49}$$

$$\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r\ge\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}r\,\mathrm{e}^{-\alpha_1 h}h^{-1}\boldsymbol{R}\int_{t-h}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r \tag{50}$$

Thus, using (50) it is possible to note

$$\begin{aligned}&-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\mathrm{e}^{\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r-\\ &-\boldsymbol{p}^T(t)\boldsymbol{W}\int_{t-h}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}r-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}r\boldsymbol{W}^T\boldsymbol{p}(t)\le\\ &\dot{\boldsymbol{p}}^T(t)\begin{bmatrix}0\\0\\-1\end{bmatrix}(-h^{-1}\mathrm{e}^{-\alpha_1 h}\boldsymbol{R})\begin{bmatrix}0\,0\,{-1}\end{bmatrix}\dot{\boldsymbol{p}}(t)+\\ &+\boldsymbol{p}^T(t)\boldsymbol{W}\begin{bmatrix}0\,0\,{-1}\end{bmatrix}\dot{\boldsymbol{p}}(t)+\dot{\boldsymbol{p}}^T(t)\begin{bmatrix}0\\0\\-1\end{bmatrix}\boldsymbol{W}^T\boldsymbol{p}(t)=\dot{\mathrm{v}}^{\circ}(t)\end{aligned} \tag{51}$$

Completing (51) to square with notation

$$\boldsymbol{Z}=-h^{-1}\mathrm{e}^{-\alpha_1 h}\boldsymbol{R} \tag{52}$$

gives

$$\begin{gathered}\dot{\mathrm{v}}^{\circ}(t)=-\boldsymbol{p}^T(t)\boldsymbol{W}\boldsymbol{Z}^{-1}\boldsymbol{W}^T\boldsymbol{p}(t)+\\ [\dot{\boldsymbol{p}}^T(t)\begin{bmatrix}0\\0\\-1\end{bmatrix}\boldsymbol{Z}+\boldsymbol{p}^T(t)\boldsymbol{W}]\boldsymbol{Z}^{-1}[\boldsymbol{W}^T\boldsymbol{p}(t)+\boldsymbol{Z}\begin{bmatrix}0\\0\\-1\end{bmatrix}^T\dot{\boldsymbol{p}}(t)]=\\ -\boldsymbol{p}^T(t)\boldsymbol{W}\boldsymbol{Z}^{-1}\boldsymbol{W}^T\boldsymbol{p}(t)+\theta(t)\end{gathered}\tag{53}$$

Since $\boldsymbol{Z} < 0$ implies $\theta(t) < 0$, it is obvious that

$$\begin{gathered}\dot{\mathrm{v}}(\boldsymbol{q}(t))-\alpha_2\,\mathrm{v}(\boldsymbol{q}(t))\le\\ \le\boldsymbol{p}^T(t)\boldsymbol{T}^{\bullet}\boldsymbol{p}(t)+\theta(t)-\boldsymbol{p}^T(t)\boldsymbol{W}\boldsymbol{Z}^{-1}\boldsymbol{W}^T\boldsymbol{p}(t)<0\end{gathered}\tag{54}$$

if and only if

$$\boldsymbol{T}=\boldsymbol{T}^{\bullet}-\boldsymbol{W}\boldsymbol{Z}^{-1}\boldsymbol{W}^T<0\tag{55}$$

Combining Schur complement property with (36) and (45), inequality (55) can now be rewritten as follows

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{S} & \boldsymbol{W}_2^T-\boldsymbol{W}_1 & \boldsymbol{W}_3^T-hd\boldsymbol{R} & h\boldsymbol{W}_1\\ * & -\boldsymbol{W}_2-\boldsymbol{W}_2^T & -\boldsymbol{W}_3^T & h\boldsymbol{W}_2\\ * & * & -d\boldsymbol{R} & h\boldsymbol{W}_3\\ * & * & * & -\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\end{bmatrix}<0\tag{56}$$

$$\boldsymbol{S}=\boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}+h\boldsymbol{A}^T\boldsymbol{R}\boldsymbol{A}-\alpha_2\boldsymbol{P}-h^2d\boldsymbol{R}+\boldsymbol{W}_1+\boldsymbol{W}_1^T\tag{57}$$

Given matrices $\boldsymbol{P} > 0$, $\boldsymbol{R} > 0$, and scalars $\alpha_1 > 0$, $h > 0$, then (54) is negative, if there exist scalar $\alpha_2 > 0$ and matrices $\boldsymbol{W}_i,\ i=1,2,3$ such that (56) holds. Therefore it also holds

$$\mathrm{e}^{-\alpha_2 t}\,\dot{\mathrm{v}}(\boldsymbol{q}(t))-\mathrm{e}^{-\alpha_2 t}\,\alpha_2\,\mathrm{v}(\boldsymbol{q}(t))<0\tag{58}$$

(compare with [18]). Integrating (58) from 0 to $t$ results in the formulas

$$\begin{gathered}\int_0^t(\mathrm{e}^{-\alpha_2 r}\,\dot{\mathrm{v}}(\boldsymbol{q}(r))-\mathrm{e}^{-\alpha_2 r}\,\alpha_2\,\mathrm{v}(\boldsymbol{q}(r)))\,\mathrm{d}\,r=\\ =\mathrm{e}^{-\alpha_2 r}\mathrm{v}(\boldsymbol{q}(r))\Big|_0^t=\mathrm{e}^{-\alpha_2 t}\,\mathrm{v}(\boldsymbol{q}(t))-\mathrm{v}(\boldsymbol{q}(0))<0\end{gathered}\tag{59}$$

$$\mathrm{v}(\boldsymbol{q}(t))<\mathrm{e}^{\alpha_2 t}\,\mathrm{v}(\boldsymbol{q}(0))\tag{60}$$

respectively. It is obvious, that with these conditions the trajectories of an autonomous system are stable.

## 4. Exponential Stability of the Controlled System

Since in this case the derivative of Lyapunov-Krasovskii functional also takes form (26), it implies

$$\begin{gathered}\dot{\mathrm{v}}(\boldsymbol{q}(t))+\alpha_1\,\mathrm{v}(\boldsymbol{q}(t))=\alpha_1\boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{q}(t)+\boldsymbol{q}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t)+\\ +\dot{\boldsymbol{q}}^T(t)\boldsymbol{P}\boldsymbol{q}(t)+h\dot{\boldsymbol{q}}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t)-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r)\,\mathrm{d}\,r<0\end{gathered}\tag{61}$$

With the known matrix $\boldsymbol{K}$ of the control law (4) the system state equation is

$$\dot{\boldsymbol{q}}(t)=\boldsymbol{A}\boldsymbol{q}(t)+\boldsymbol{B}\boldsymbol{K}\boldsymbol{q}(t-h)\tag{62}$$

Defining

$$\boldsymbol{v}^T(t)=\left[\boldsymbol{q}^T(t)\ \ \boldsymbol{q}^T(t-h)\right]\tag{63}$$

$$\boldsymbol{s}^T(t,r)=\left[\boldsymbol{q}^T(t)\ \ \boldsymbol{q}^T(t-h)\ \ \dot{\boldsymbol{q}}^T(r)\right]\tag{64}$$

then, using (13) it holds

$$\begin{gathered}\boldsymbol{v}^T(t)\boldsymbol{V}[\boldsymbol{q}^T(t)-\boldsymbol{q}^T(t-h)-\int_{t-h}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}\,r]+\\ [\boldsymbol{q}^T(t)-\boldsymbol{q}^T(t-h)-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}\,r]\boldsymbol{V}^T\boldsymbol{v}(t)=0\end{gathered}\tag{65}$$

$$\begin{gathered}\boldsymbol{v}^T(t)\boldsymbol{V}\left[1-1 0\right]\boldsymbol{v}(t)+\boldsymbol{v}^T(t)\left[1-1 0\right]^T\boldsymbol{V}^T\boldsymbol{v}(t)-\\ -\boldsymbol{v}^T(t)\boldsymbol{V}\int_{t-h}^{t}\dot{\boldsymbol{q}}(r)\,\mathrm{d}\,r-\int_{t-h}^{t}\dot{\boldsymbol{q}}^T(r)\,\mathrm{d}\,r]\boldsymbol{V}^T\boldsymbol{v}(t)=0\end{gathered}\tag{66}$$

respectively, where

$$\boldsymbol{V}^T=\left[\boldsymbol{V}_1^T\ \boldsymbol{V}_2^T\right]\tag{67}$$

$$\boldsymbol{v}^T(t)\boldsymbol{V}\left[1-1 0\right]\boldsymbol{v}(t)+\boldsymbol{v}^T(t)\left[1-1 0\right]^T\boldsymbol{V}^T\boldsymbol{v}(t)=\boldsymbol{v}^T(t)\boldsymbol{U}^{\circ}\boldsymbol{v}(t)\tag{68}$$

$$\boldsymbol{U}^{\circ}=\boldsymbol{V}\left[1-1 0\right]+\begin{bmatrix}1\\-1\\0\end{bmatrix}\boldsymbol{V}^T=\begin{bmatrix}\boldsymbol{V}_1^T+\boldsymbol{V}_1 & \boldsymbol{V}_2^T-\boldsymbol{V}_1\\ * & -\boldsymbol{V}_2^T-\boldsymbol{V}_2\end{bmatrix}\tag{69}$$

On the other hand, for $h > 0$ and any semi-positive definite matrix $\boldsymbol{Q}\ge 0$ it is true

$$\begin{gathered}h\boldsymbol{v}^T(t)\boldsymbol{Q}\boldsymbol{v}(t)-h\boldsymbol{v}^T(t)\boldsymbol{Q}\boldsymbol{v}(t)=\\ h\boldsymbol{v}^T(t)\boldsymbol{Q}\boldsymbol{v}(t)-\int_{t-h}^{t}\boldsymbol{v}^T(t)\boldsymbol{Q}\boldsymbol{v}(t)\,\mathrm{d}\,r=0\end{gathered}\tag{70}$$

where

$$\boldsymbol{Q}=\begin{bmatrix}\boldsymbol{Q}_{11} & \boldsymbol{Q}_{12}\\ * & \boldsymbol{Q}_{22}\end{bmatrix}\ge 0\tag{71}$$

In this regime the constraint can be adapted for solution in the structure (67). Then it is possible to combine elements in integrals in (61), (66) and (70) as follows

$$\begin{gathered}\boldsymbol{v}^T(t)\boldsymbol{Q}\boldsymbol{v}(t)+\dot{\boldsymbol{q}}^T(r)\boldsymbol{V}^T\boldsymbol{v}(t)+\boldsymbol{v}^T(t)\boldsymbol{V}\dot{\boldsymbol{q}}(r)+\\ +\dot{\boldsymbol{q}}^T(r)\,\mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\dot{\boldsymbol{q}}(r)=\boldsymbol{s}^T(t,r)\boldsymbol{Q}^{\bullet}\boldsymbol{s}(t,r)\end{gathered}\tag{72}$$

$$\boldsymbol{Q}^{\bullet}=\begin{bmatrix}\boldsymbol{Q}_{11} & \boldsymbol{Q}_{12} & \boldsymbol{V}_1\\ * & \boldsymbol{Q}_{22} & \boldsymbol{V}_2\\ * & * & \mathrm{e}^{-\alpha_1 h}\boldsymbol{R}\end{bmatrix}\tag{73}$$

Now it is possible to express

$$h\dot{\boldsymbol{q}}(t)\boldsymbol{R}\dot{\boldsymbol{q}}(t)=\boldsymbol{v}^T(t)\begin{bmatrix}\boldsymbol{A}^T\\ \boldsymbol{K}^T\boldsymbol{B}^T\end{bmatrix}h\boldsymbol{R}\left[\boldsymbol{A}\ \boldsymbol{B}\boldsymbol{K}\right]\boldsymbol{v}(t)=\boldsymbol{v}^{(T)}(t)\boldsymbol{U}_1\boldsymbol{v}(t)\tag{74}$$

$$\boldsymbol{U}_{\boldsymbol{1}}=\begin{bmatrix}h\boldsymbol{A}^T\boldsymbol{R}\\ h\boldsymbol{K}^T\boldsymbol{B}^T\boldsymbol{R}\end{bmatrix}(h\boldsymbol{R})^{-1}\left[h\boldsymbol{R}\boldsymbol{A}\ \ h\boldsymbol{R}\boldsymbol{B}\boldsymbol{K}\right]\tag{75}$$

and

$$\begin{gathered}\alpha_1\boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{q}(t)+\boldsymbol{q}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t)+\dot{\boldsymbol{q}}^T(t)\boldsymbol{P}\boldsymbol{q}(t)=\boldsymbol{v}^T(t)\boldsymbol{U}_{\boldsymbol{2}}\boldsymbol{v}(t)=\\ =\boldsymbol{v}^T(t)\left[\begin{bmatrix}\boldsymbol{A}^T\\ \boldsymbol{K}^T\boldsymbol{B}^T\end{bmatrix}\boldsymbol{P}\left[\boldsymbol{1}\boldsymbol{0}\right]+\begin{bmatrix}1\\0\end{bmatrix}\boldsymbol{P}\left[\boldsymbol{A}\ \boldsymbol{B}\boldsymbol{K}\right]+\begin{bmatrix}1\\0\end{bmatrix}\alpha_1\boldsymbol{P}\left[\boldsymbol{1}\boldsymbol{0}\right]\right]\boldsymbol{v}(t)\end{gathered}\tag{76}$$

$$\boldsymbol{U}_2=\begin{bmatrix}\boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}+\alpha_1\boldsymbol{P} & \boldsymbol{P}\boldsymbol{B}\boldsymbol{K}\\ \boldsymbol{K}\boldsymbol{B}\boldsymbol{P} & \boldsymbol{0}\end{bmatrix}\tag{77}$$

Inequality (61) can be written now in the form

$$\dot{\mathrm{v}}(\boldsymbol{q}(t))+\alpha_1\,\mathrm{v}(\boldsymbol{q}(t))=\boldsymbol{v}^T(t)\boldsymbol{U}^{\bullet}\boldsymbol{v}(t)-\int_{t-h}^{t}\boldsymbol{s}^T(t,r)\boldsymbol{Q}^{\bullet}\boldsymbol{s}(t,r)\,\mathrm{d}\,r<0\tag{78}$$

where

$$\boldsymbol{U}^{\bullet}=h\boldsymbol{Q}+\boldsymbol{U}_1+\boldsymbol{U}_2+\boldsymbol{U}^{\circ}<0\tag{79}$$

Using Schur complement property (79) can be partitioned as

$$\boldsymbol{U}^{\bullet}=\begin{bmatrix}\boldsymbol{U}_{11} & \boldsymbol{U}_{12} & \boldsymbol{U}_{13}\\ * & \boldsymbol{U}_{22} & \boldsymbol{U}_{23}\\ * & * & \boldsymbol{U}_{33}\end{bmatrix}<0\tag{80}$$

where

$$\boldsymbol{U}_{11} = \boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \alpha_1\boldsymbol{P} + \boldsymbol{V}_1^T + \boldsymbol{V}_1 + h\boldsymbol{Q}_{11} \quad (81)$$

$$\boldsymbol{U}_{12} = \boldsymbol{P}\boldsymbol{B}\boldsymbol{K} + \boldsymbol{V}_2^T - \boldsymbol{V}_1 + h\boldsymbol{Q}_{12} \quad (82)$$

$$\boldsymbol{U}_{22} = -\boldsymbol{V}_2^T - \boldsymbol{V}_2 + h\boldsymbol{Q}_{22} \quad (83)$$

$$\boldsymbol{U}_{13} = h\boldsymbol{A}^T\boldsymbol{R} \quad (84)$$

$$\boldsymbol{U}_{23} = h\boldsymbol{K}^T\boldsymbol{B}^T\boldsymbol{R} \quad (85)$$

$$\boldsymbol{U}_{33} = -h\boldsymbol{R} \quad (86)$$

It is evident, that for given constants $\alpha_1 > 0$, $h > 0$, and matrix $\boldsymbol{K}$ the system is stable, if there exist matrices $\boldsymbol{P} > 0$, $\boldsymbol{R} > 0$, and $\boldsymbol{Q} > 0$, as well as $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ such that

$$\boldsymbol{U}^{\bullet} < 0, \qquad \boldsymbol{Q}^{\bullet} \ge 0 \quad (87)$$

Therefore, it holds

$$\mathrm{e}^{\alpha_1 t}\,\dot{\mathrm{v}}(\boldsymbol{q}(t)) - \mathrm{e}^{\alpha_1 t}\,\alpha_1\,\mathrm{v}(\boldsymbol{q}(t)) < 0 \quad (88)$$

Integrating (88) from 0 to *t* gives

$$\begin{aligned}&\int_0^t (\mathrm{e}^{\alpha_1 r}\,\dot{\mathrm{v}}(\boldsymbol{q}(r)) - \mathrm{e}^{\alpha_1 r}\,\alpha_1\,\mathrm{v}(\boldsymbol{q}(r)))\,\mathrm{d}\,r = \\ &= \mathrm{e}^{\alpha_1 r}\mathrm{v}(\boldsymbol{q}(r))\Big|_0^t = \mathrm{e}^{\alpha_1 t}\,\mathrm{v}(\boldsymbol{q}(t)) - \mathrm{v}(\boldsymbol{q}(0)) < 0\end{aligned} \quad (89)$$

and so

$$\mathrm{v}(\boldsymbol{q}(t)) < \mathrm{e}^{-\alpha_1 t}\,\mathrm{v}(\boldsymbol{q}(0)) \quad (90)$$

It is clear, that satisfying these conditions the control ensures that all trajectories of the networked controlled system are stable.

## 5. Optimization

Solving (87) with (73), (80) - (86) can be obtained any $h > 0$. Then according (56) it is possible to compute $\alpha_2 > 0$ and to approximate intervening time $h^{\circ} > 0$ as follows

$$h^{\circ} = h + \frac{\alpha_1 - \alpha^*}{\alpha_2 + \alpha^*}(h - h_0) \quad (91)$$

where

$$0 \le \tau_k \le h_0, \qquad 0 < \alpha^* < \alpha_1 \quad (92)$$

Thus, an optimal solution can be obtained as a minimization of (56) with respect to $\alpha_2 > 0$. It is obvious, that the maximum of $h > 0$ does not necessarily means the maximum of the parameter $h^{\circ}$.

Solving all matrix inequalities, i.e. (73), (80), as well as (56), one can obtain the average decay degree $\alpha_2 = 0.5\alpha^*$ for which switched system is exponentially stable.

## Concluding Remarks

This paper presents the modified method of determining the delay-dependent exponential stability criteria for the event-time--driven modes in the networked control system. Based on the linear matrix inequalities some free weighting matrix design parameters are introduced to obtain size of the available rate under which the system can stay exponential stable. It seems that these criteria are less conservative then existing ones.

### Acknowledgments

*The work presented in this paper was supported by Grant Agency of Ministry of Education and Academy of Science of Slovak Republic VEGA under Grant No. 1/0328/08. This support is very gratefully acknowledged.*

## References

[1] BOYD, D., EL GHAOUI, L., PERON, E., BALAKRISHNAN, V.: *Linear Matrix Inequalities in System and Control Theory*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[2] DRITSAS, L., TZES, A.: Constrained control of event-driven networked systems. In *Proceedings of the 17th World Congress IFAC 2008*, Seoul, 11600-11605, 2008.

[3] FILASOVÁ, A.: Lyapunov function in robust control. In *Applied & Computing Mathematics. Proceedings of the 119th Pannonian Applied Mathematical Meeting PAMM '119*, Herľany, Vol. I, 23-27, 1997.

[4] FRIDMAN, E.: New Lyapunov-Krasovskii functionals for stability of linear retarded and neutral type systems. *Systems and Control Letters*, **43**:4, 309-319, 2001.

[5] GU, K., KHARITONOV, V.L., CHEN, J.: *Stability of Time-Delay Systems*. Birkhäuser, Boston, 2003.

[6] KOLMANOVSKII, V.B., NICULESCU, S., RICHARD, J.P.: On the Lyapunov Krasovskii functionals for stability analysis of linear delay systems. *International Journal of Control*, **72**, 374-384, 1999.

[7] KROKAVEC D., FILASOVÁ, A.: *Dynamic Systems Diagnosis*. Elfa, Košice, 2007. (in Slovak)

[8] KROKAVEC D., FILASOVÁ, A.: *Discrete-Time Systems*. Elfa, Košice, 2008. (in Slovak)

[9] LI, X., de SOUZA, C.E.: Delay-dependent robust stability and stabilization of uncertain time-delayed systems: A linear matrix inequality approach. *IEEE Transactions on Automatic Control*, **42**, 1144-1148, 1997.

[10] MATVEEV, A.S., SAVKIN, A.V.: Estimation and Control over Communication Networks. Birkhäuser, Boston, 2009.

[11] NICULESCU, S.I., VERIEST, E.I., DUGARD, L., DION, J.M.: Stability and robust stability of time-delay systems: A guided tour. In *Stability and Control of Time-delay Systems*. Springer-Verlag, Berlin, 1998.

[12] PARK. P.: A delay-dependent stability criterion for systems with uncertain time-invariant delays. *IEEE Transactions on Automatic Control*, **44**, 876-877, 1997.

[13] SÁS, M., FILASOVÁ, A., KLACIK, J.: K-nearest neighbor algorithm in multi-model switching control. Annals of the University of Craiova, **2(29)**:1, 104-108, 2005.

[14] SÁS, M., FILASOVÁ, A.: Multiple-model control of MIMO systems. Bayesian view point to parameter estimation. In *Proceedings of the 8th International Carpathian Control Conference ICCC '2007*. Štrbské Pleso, 615-618, 2007.

[15] SHAKED, U., YAESH, I., de SOUZA, C.E.: Bounded real criteria for linear time systems with state-delay. *IEEE Transactions on Automatic Control*, **43**, 1116-1121, 1998.

[16] SUN, X.M., LIU, G.P., REES, D., WANG, W.: A novel method of stability analysis for networked control systems. In *Proceedings of the 17th World Congress IFAC 2008*, Seoul, 4852-4856, 2008.

[17] VESELÝ, V., ROSINOVÁ, D.: Output feedback controller design. Non-iterative LMI approach. *Journal of Electrical Engineering*, 59:6, 317-321, 2008.

[18] XIAO, S.P., WU, M., SHE, J.H.: Non-fragile delay-dependent $H_{inf}$ control of linear time-delay system with uncertainties in state and control input. *Journal of Central South University of Technlogy*, 15, 712-719, 2008.

[19] ZHANG, X.M., WU, M., SHE, J.H., HE, Y.: Delay-dependent stabilization of linear systems with time-varying state and input delays. *Automatica*, **41**:8, 1405-1412, 2005.

[20] ZHONG, Q.C.: *Robust Control of Time-delay Systems*. Springer- Verlag, London, 2009.

**Dušan Krokavec, prof. Ing. PhD.**

Technical University of Košice
Faculty of Electrical Engineering and Informatics
Department of Cybernetics and Artificial Intelligence
Letná 9
042 00 Košice
Tel.: ++421 55 602 2564
E-mail: dusan.krokavec@tuke.sk

# A Comparision of Different EKF Approaches for Parameters Estimation

**Juraj Vöröš, Ján Mikleš, Ľuboš Čirka**

**Abstract**

In many chemical engineering applications the extended Kalman filter (EKF) is often used to deal with certain classes of nonlinear systems. This paper compares basic and polynomial approach of EKF for parameters estimation of nonlinear continuous-time stochastic systems. The proposed approaches are used to estimate constants $k_{11}$ and $k_{22}$ for interacting tank-in-series process and frequency factor $k_0$ and temperature of reaction mixture $\vartheta$ for continuous stirred-tank reactor (CSTR).

**Key words:** Extended Kalman filter, nonlinear systems, parameter estimation

## Introduction

Parameter estimation is one of the steps involved in the formulation and validation of a mathematical model and refers to the process of obtaining values of the parameters from the matching of the model-based calculated values to the set of measurements (data). [3]

Many papers have studied parameters estimation using various techniques. In [5] comprehensive approach to estimate kinetic parameters when the involved reactions contain lumped chemical species is presented. This approach is based on representing rate constants with a continuous probability distribution function. In [1] simultaneously both heat transfer and kinetic parameters estimation under reacting conditions in a single tube wall-cooled fixed-bed reactor, and a two-stage parameter estimation was developed. The advantages of using maximum-likelihood estimators rather than simple least-squares estimators for the problem of finding unsaturated hydraulic parameters were demonstrated in [6]. In [4] global optimization approach tailored to the error-in-variables parameter estimation problem for nonlinear algebraic model was presented. A modified genetic algorithm to solve the parameter identification problem for nonlinear digital filter was used in [7]. Model estimation using fast orthogonal search is presented in [2].

In this work we apply basic and polynomial approaches of EKF to estimate constants $k_{11}$ and $k_{22}$ for interacting tank-in-series process and frequency factor $k_0$ and temperature of reaction mixture $\vartheta$ for continuous stirred-tank reactor and compare their performance.

## 1. The Continuous-time Extended Kalman Filter

Consider the following general nonlinear system model [9]:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, t) \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{v}, t) \\ \boldsymbol{w} &\sim (0, \boldsymbol{Q}) \\ \boldsymbol{v} &\sim (0, \boldsymbol{R}) \end{aligned} \qquad (1)$$

where $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are general nonlinear functions. The noise processes $\boldsymbol{w}$ and $\boldsymbol{v}$ are white, zero-mean, uncorrelated, and have known covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. Equation (1) is expanded using Taylor series around a nominal control $\boldsymbol{u_0}$, nominal state $\boldsymbol{x_0}$, nominal output $\boldsymbol{y_0}$, and nominal noise values $\boldsymbol{w_0}$ and $\boldsymbol{v_0}$. This gives the following approximately correct linear system

$$\begin{aligned} \Delta\dot{\boldsymbol{x}} &= \boldsymbol{A}\Delta\boldsymbol{x} + \boldsymbol{L}\boldsymbol{w} \\ &= \boldsymbol{A}\Delta\boldsymbol{x} + \tilde{\boldsymbol{w}} \\ \tilde{\boldsymbol{w}} &\sim (0, \tilde{\boldsymbol{Q}}), \tilde{\boldsymbol{Q}} = \boldsymbol{L}\boldsymbol{Q}\boldsymbol{L}^T \\ \Delta\boldsymbol{y} &= \boldsymbol{C}\Delta\boldsymbol{x} + \boldsymbol{M}\boldsymbol{v} \\ &= \boldsymbol{C}\Delta\boldsymbol{x} + \tilde{\boldsymbol{v}} \\ \tilde{\boldsymbol{v}} &\sim (0, \tilde{\boldsymbol{R}}), \tilde{\boldsymbol{R}} = \boldsymbol{M}\boldsymbol{R}\boldsymbol{M}^T \end{aligned} \qquad (2)$$

The Δ quantities in the above equations are defined as deviations from the nominal trajectory:

$$\begin{aligned} \Delta\dot{\boldsymbol{x}} &= \dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_0 \\ \Delta\boldsymbol{y} &= \boldsymbol{y} - \boldsymbol{y}_0 \end{aligned} \qquad (3)$$

where

$$\begin{aligned} \dot{\boldsymbol{x}}_0 &= \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{w}_0, t) \\ \boldsymbol{y}_0 &= \boldsymbol{h}(\boldsymbol{x}_0, \boldsymbol{v}_0, t) \end{aligned} \qquad (4)$$

We assume that the control $\boldsymbol{u}(t)$ is perfectly known, so that $\boldsymbol{u_0}(t) = \boldsymbol{u}(t)$ and $\Delta\boldsymbol{u}(t) = \boldsymbol{0}$. The matrices on the right side of (2) are given as

$$\boldsymbol{A} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_0, \boldsymbol{L} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}}\right|_0, \boldsymbol{C} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_0, \boldsymbol{M} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{v}}\right|_0 \qquad (5)$$

The Kalman filter equations for the linearized Kalman filter are

$$\begin{aligned} \Delta\dot{\hat{\boldsymbol{x}}} &= \boldsymbol{A}\Delta\hat{\boldsymbol{x}} + \boldsymbol{K}(\Delta\boldsymbol{y} - \boldsymbol{C}\Delta\hat{\boldsymbol{x}}) \\ \boldsymbol{K} &= \boldsymbol{P}\boldsymbol{C}^T\tilde{\boldsymbol{R}}^{-1} \\ \dot{\boldsymbol{P}} &= \boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^T + \tilde{\boldsymbol{Q}} - \boldsymbol{P}\boldsymbol{C}^T\tilde{\boldsymbol{R}}^{-1}\boldsymbol{C}\boldsymbol{P} \\ \hat{\boldsymbol{x}} &= \boldsymbol{x}_0 + \Delta\hat{\boldsymbol{x}} \end{aligned} \qquad (6)$$

where $\boldsymbol{P}$ is equal to the covariance of the estimation error.

Now we will extend the linearized Kalman filter to directly estimate the states of a nonlinear system and linearize the nonlinear system around the Kalman filter estimate. This is the idea of EKF.

Combine the $\dot{x}_0$ in equation (4) with $\Delta\dot{\hat{x}}$ the expression in equation (6) to obtain

$$\dot{x}_0 + \Delta\dot{\hat{x}} = f(x_0, u_0, w_0, t) + A\Delta\hat{x} + \\ + K[y - y_0 - C(\hat{x} - x_0)] \tag{7}$$

Now choose $x_0(t) = \hat{x}(t)$ so that $\Delta\hat{x}(t) = 0$ and $\Delta\dot{\hat{x}}(t) = 0$. Then equation (7) becomes

$$\dot{\hat{x}} = f(\hat{x}, u, w_0, t) + K[y - h(\hat{x}, v_0, t)] \tag{8}$$

The continuous-time EKF can be summarized as follows:

Compute the following partial derivative matrices evaluated at the current state estimate:

$$A = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}}, L = \left.\frac{\partial f}{\partial w}\right|_{\hat{x}}, C = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}}, M = \left.\frac{\partial h}{\partial v}\right|_{\hat{x}} \tag{9}$$

Compute the following matrices:

$$\begin{aligned} \tilde{Q} &= LQL^T \\ \tilde{R} &= MRM^T \end{aligned} \tag{10}$$

Execute the following Kalman filter equations:

$$\begin{aligned} \dot{\hat{x}} &= f(\hat{x}, u, w_0, t) + K[y - h(\hat{x}, v_0, t)] \\ K &= PC^T\tilde{R}^{-1} \\ \dot{P} &= AP + PA^T + \tilde{Q} - PC^T\tilde{R}^{-1}CP \end{aligned} \tag{11}$$

where the nominal noise values are given as $\boldsymbol{w_0} = \boldsymbol{0}$ and $\boldsymbol{v_0} = \boldsymbol{0}$.

This is the **basic approach of EKF** where Kalman gain matrix ***K*** design is based on covariance matrix of estimation error ***P*** obtained from differential Riccati equation.

Now we will derive the **polynomial approach of EKF**. Kalman gain matrix ***K*** design is based on solution of the Diophantine equation.

Matrix transfer functions of the observable system [8] are given as

$$C(sI - A)^{-1} = A_L^{-1}(s)B_{Ls}(s) \tag{12}$$

where $\boldsymbol{A_L}$, $\boldsymbol{B_{Ls}}$ are left coprime polynomial matrices and $\boldsymbol{A_L}$ is row reduced.

If the gain matrix ***K*** exist, it is unique and of the form

$$K = Y_R X_R^{-1} \tag{13}$$

Then $\boldsymbol{X_R}$ and $\boldsymbol{Y_R}$ are solution of the Diophantine equation

$$A_L(s)X_R + B_{Ls}(s)Y_R = O_L(s) \tag{14}$$

$\boldsymbol{O_L}(s)$ is a stable polynomial matrix with $\det O_L(s) \neq 0$ and is given from spectral factorization as follows. Adding ***sP*** to either side of algebraic Riccati equation (assuming, that the noise processes have known covariance matrices ***Q*** and ***R*** = ***I***)

$$AP + PA^T - PC^TCP = -Q \tag{15}$$

gives

$$(sI - A)P + P(-sI - A^T) = Q - PC^TCP \tag{16}$$

Multiplying from left by $C(sI - A)^{-1}$ and from right by $(-sI - A^T)^{-1}C^T$ and using (11), (12) yields

$$A_L(s)K^TB_{Ls}^T(s) + B_{Ls}(s)KA_L^T(-s) + \\ + B_{Ls}(s)KK^TB_{Ls}(-s) = B_{Ls}(s)QB_{Ls}^T(-s) \tag{17}$$

Adding $A_L(s)A_L^T(-s)$ to either side of this equation gives

$$(A_L(s) + B_{Ls}(s)K)(A_L^T(-s) + K^TB_{Ls}^T(-s)) = \\ = A_L(s)A_L^T(-s) + B_{Ls}(s)QB_{Ls}^T(-s) \tag{18}$$

Then matrix $\boldsymbol{O_L}(s)$ can be found from the spectral factorization equation

$$O_L(s)O_L^T(-s) = A_L(s)A_L^T(-s) + \\ + B_{Ls}(s)QB_{Ls}^T(-s) \tag{19}$$

## 2. Parameters Estimation

In order to estimate the parameters ***θ***, we first augment the state with the parameters as extra states with no dynamics to obtain an augment state vector [9]:

$$\tilde{x} = \begin{bmatrix} x \\ \theta \end{bmatrix} \tag{20}$$

Our augment system model can be written as

$$\dot{\tilde{x}} = \begin{bmatrix} f(x, u, w, t) \\ 0 \end{bmatrix} = \tilde{f}(\tilde{x}, u, w, t) \tag{21}$$

Note that $\tilde{f}(\tilde{x}, u, w, t)$ is a nonlinear function of the augmented state $\tilde{x}$. We can therefore use an extended Kalman filter to estimate $\tilde{x}$.

## 3. Mathematical Modelling

### 3.1 Interacting Tank-in-Series Process

We consider [8] the interacting tank-in-series process shown in Fig. 1. The process input variable is the flow rate $q_0$.

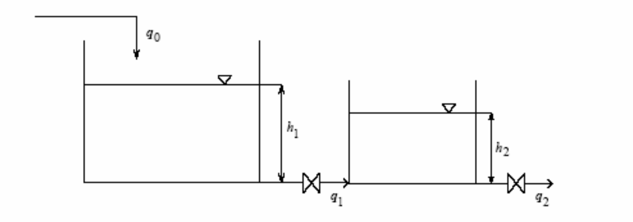


**Fig.1 An interacting tank-in-series process**

The process state variables are heights of liquid in tanks $h_1$, $h_2$. Assuming that liquid density, $F_1$ and $F_2$ are constant, mass balance for the process yields

$$F_1\frac{dh_1}{dt} = q_0 - q_1 \tag{22}$$

$$F_2\frac{dh_2}{dt} = q_1 - q_2 \tag{23}$$

Inlet flow rate $q_0$ is independent of tank states whereas $q_1$ depends on the difference between liquid heights

$$q_1 = k_{11}\sqrt{h_1 - h_2} \tag{24}$$

Outlet flow rate $q_2$ depends on liquid height in the second tank

$$q_2 = k_{22}\sqrt{h_2} \tag{25}$$

Substituting $q_1$ from equation (24) and $q_2$ from (25) into (22) and (23) we get

$$\begin{aligned} \frac{dh_1}{dt} &= \frac{q_0}{F_1} - \frac{k_{11}}{F_1}\sqrt{h_1 - h_2} \\ \frac{dh_2}{dt} &= \frac{k_{11}}{F_2}\sqrt{h_1 - h_2} - \frac{k_{22}}{F_2}\sqrt{h_2} \end{aligned} \tag{26}$$

with arbitrary initial conditions

$$\begin{aligned} h_1(0) &= h_{10} \\ h_2(0) &= h_{20} \end{aligned} \tag{27}$$

To estimate constants $k_{11}$ and $k_{22}$, two more equations are needed

$$\begin{aligned} \frac{dk_{11}}{dt} &= 0 \qquad k_{11}(0) = k_{11}^0 \\ \frac{dk_{22}}{dt} &= 0 \qquad k_{22}(0) = k_{22}^0 \end{aligned} \tag{28}$$

Equations (18) and (20) are now nonlinear system model for parameters estimation.

According to (9)

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{29}$$

where

$$a_{11} = -\frac{\hat{k}_{11}}{2F_1\sqrt{\hat{h}_1 - \hat{h}_2}} \qquad a_{12} = \frac{\hat{k}_{11}}{2F_1\sqrt{\hat{h}_1 - \hat{h}_2}}$$

$$a_{13} = -\frac{\sqrt{\hat{h}_1 - \hat{h}_2}}{F_1} \qquad a_{21} = \frac{\hat{k}_{11}}{2F_2\sqrt{\hat{h}_1 - \hat{h}_2}}$$

$$a_{22} = -\frac{\hat{k}_{11}}{2F_2\sqrt{\hat{h}_1 - \hat{h}_2}} - \frac{\hat{k}_{22}}{2F_2\sqrt{\hat{h}_2}}$$

$$a_{23} = \frac{\sqrt{\hat{h}_1 - \hat{h}_2}}{F_2} \qquad a_{24} = -\frac{\sqrt{\hat{h}_2}}{F_2}$$

and

$$\boldsymbol{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{30}$$

Parameters of the interacting tank-in-series process are shown on Tab.1.

| | | |
|---|---|---|
| $q_0$ | 1 | $m^3 h^{-1}$ |
| $k_{11}$ | 0.8 | $m^{5/2}$ h |
| $k_{22}$ | 1.5 | $m^{5/2}$ h |
| $F_1$ | 0.8 | $m^2$ |
| $F_2$ | 0.8 | $m^2$ |

**Tab.1 Parameters of the interacting tank-in-series process**

### 3.2 Continuous stirred-tank reactor

We consider CSTR [8] with a simple exothermal reaction *A*→*B* (Fig. 2).

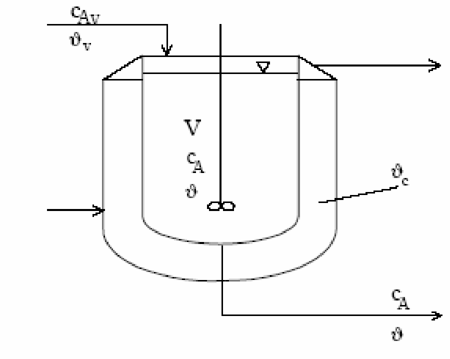


**Fig.2 A nonisothermal reactor**

For the development of a mathematical model of the CSTR, the following assumptions are made: neglected heat capacity of inner walls of the reactor, constant density and specific heat capacity of liquid, constant reactor volume, constant overall heat transfer coefficient, and constant and equal input and output volumetric flow rates. As the reactor is well-mixed, the outlet stream concentration and temperature are identical with those in the tank.

A mass balance of component A can be expressed as

$$V\frac{dc_A}{dt} = qc_{AV} - qc_A - Vr(c_A, \vartheta) \tag{31}$$

The rate of reaction is strong function of concentration and temperature (Arrhenius law)

$$r(c_A, \vartheta) = kc_A = k_0 e^{-\frac{E}{R\vartheta}} c_A \tag{32}$$

Heat balance gives

$$\begin{aligned} V\rho c_p \frac{d\vartheta}{dt} &= q\rho c_p \vartheta_V - q\rho c_p \vartheta - \\ &- \alpha F(\vartheta - \vartheta_C) + V(-\Delta H) r(c_A, \vartheta) \end{aligned} \tag{33}$$

Initial conditions are

$$\begin{aligned} c_A(0) &= c_{A0} \\ \vartheta(0) &= \vartheta_0 \end{aligned} \tag{34}$$

To estimate the temperature of reaction mixture $\vartheta$ and frequency factor $k_0$, one more equation is needed

$$\frac{dk_0}{dt} = 0 \qquad k_0(0) = k_0^0 \tag{35}$$

Equations (31), (33) and (35) are now nonlinear system model for parameters estimation. According to (9)

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 \end{pmatrix} \tag{36}$$

where

$$a_{11} = -\frac{q}{V} - \hat{k}_0 e^{-\frac{E}{R\hat{\vartheta}}} \qquad a_{12} = -\frac{\hat{c}_A E \hat{k}_0 e^{-\frac{E}{R\hat{\vartheta}}}}{R\hat{\vartheta}^2}$$

$$a_{13} = -\hat{c}_A e^{-\frac{E}{R\hat{\vartheta}}} \qquad a_{21} = \frac{\hat{k}_0 e^{-\frac{E}{R\hat{\vartheta}}}(-\Delta H)\hat{c}_A}{\rho c_P}$$

$$a_{22} = -\frac{q}{V} - \frac{\alpha F}{V\rho c_P} + \frac{\hat{k}_0 E e^{-\frac{E}{R\hat{\vartheta}}}(-\Delta H)\hat{c}_A}{R\hat{\vartheta}^2 \rho c_P}$$

$$a_{23} = -\frac{e^{-\frac{E}{R\hat{\vartheta}}}(-\Delta H)\hat{c}_A}{\rho c_P}$$

and

$$\boldsymbol{C} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \tag{37}$$

Parameters of the reaction and reactor are shown on Tab.2.

| | | |
|---|---|---|
| $c_{AV}$ | 1.2 | kmol m$^{-3}$ |
| $c_P$ | 4.05 | kJ kg$^{-1}$ K$^{-1}$ |
| $E$ | 107280 | kJ kmol$^{-1}$ |
| $F$ | 6.08 | m$^2$ |
| $k_0$ | 7.93e15 | min$^{-1}$ |
| $V$ | 1.7 | m$^3$ |
| $\Delta H$ | -150000 | kJ kmol$^{-1}$ |
| $q$ | 0.2 | m$^3$ min$^{-1}$ |
| $R$ | 8.314 | kJ kmol$^{-1}$ K$^{-1}$ |
| $\alpha$ | 41.2 | kJ m$^{-2}$ min$^{-1}$ K$^{-1}$ |
| $\vartheta_C$ | 318 | K |
| $\vartheta_V$ | 313 | K |
| $\rho$ | 998 | kg m$^{-3}$ |

**Tab.2 Parameters of the reaction and reactor**

## 4. Simulation Results

For the tank-in-series process parameters estimation simulation, the following values were tracked: $q_0(t)$ = 1 m$^3$.h$^{-1}$ for $t$ < 0 h and $q_0(t)$ = 1.1 m$^3$.h$^{-1}$ for $t \geq 0$ h. Initial conditions for estimated parameters: $k_{11}$ = 1 m$^{5/2}$h, $k_{22}$ = 1 m$^{5/2}$h. True values of parameters are: $k_{11}$ = 0.8 m$^{5/2}$h and $k_{22}$ = 1.5 m$^{5/2}$h. Fig. 3,4 show the estimation results for tank-in-series process.

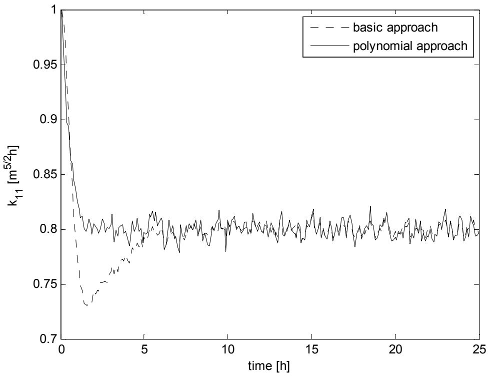


**Fig.3 Estimation of the $k_{11}$ constant**

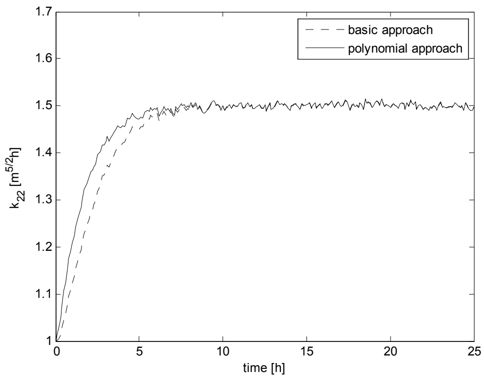


**Fig.4 Estimation of the $k_{22}$ constant**

For the CSTR parameters estimation simulation, the following values were tracked: $c_{A0}(t)$ = 1.2 kmol.m$^{-3}$ for $t$ < 0 h and $c_{A0}(t)$ = 1.15 kmol.m$^{-3}$ for $t \geq 0$ h. Initial conditions for estimated parameters: $\vartheta$ = 320 K, $k_0$ = 7x10$^{14}$ min$^{-1}$. Fig. 5,6 show the estimation results for CSTR. True value of parameter $k_0$ is 7.93x10$^{15}$ min$^{-1}$.

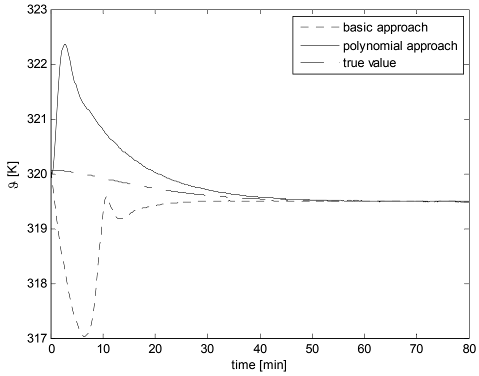


**Fig.5 Estimation of the temperature of reaction mixture**

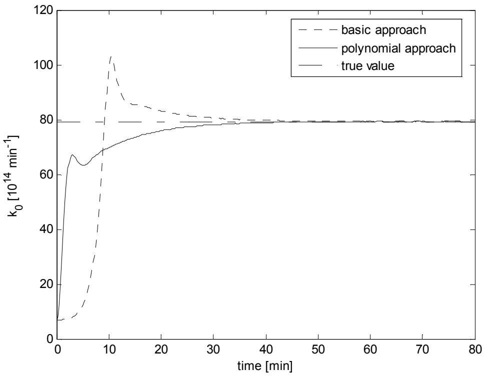


**Fig.6 Estimation of the frequency factor $k_0$**

The estimation of parameters and states is carried out in presence of noise. Because there is no general rule for the choice of the matrix $\boldsymbol{Q}$ , it was chosen experimentally in order to ensure its positive definition (diagonal matrix). From the results it is observed that both presented approaches give very high accuracy of parameters estimation. But we can see that the performance of polynomial approach algorithm is higher then the performance of basic approach algorithm, because the true values of parameters are reached faster.

## Conclusion

In this paper, parameters estimation of nonlinear continuous-time stochastic system using continuous-time extended

Kalman filter (EKF) is presented. Basic and polynomial approach of Kalman gain matrix ***K*** design was used to estimate constant $k_{11}$ and $k_{22}$ for interacting tank-in-series process and frequency factor $k_0$ and temperature of reaction mixture $\vartheta$ for continuous stirred-tank reactor.

### Acknowledgments

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09 and 1/4055/07.This work was supported by the Slovak Research and Development Agency under the contract No. VV-0029-07.

## References

[1] Cheng, Z.M., Yuan, W.K.: Simultaneous estimation of kinetic and heat transfer parameters of a wall-cooled fixed-bed reactor. *Chemical Engineering Science, 51(21)*:4791-4800, 1996.

[2] Eklund, J.M., Korenberg, M.J., McLellan, P.J.: Nonlinear system identification and control of chemical processes using fast orthogonal search. *Journal of Process Control,* 17:742-754, 2007.

[3] Englezos,P., Kalogerakis, N.: *Applied Parameter Estimation for Chemical Engineers*. Marcel Dekker, New York, 2001.

[4] Esposito, W.R., Floudas, C.A.: Parameter Estimation in Nonlinear Algebraic Models via Global Optimization. *Computers and Chemical Engineering*, 22(1):213-220, 1998.

[5] Hernández-Barajas, J.R., Vázquez-Román, R., Félix-Flores, MA.G.: A comprehensive estimation of kinetic parameters in lumped catalytic cracking reaction models. *Fuel*, 88(1):169-178, 2009.

[6] Hollenbeck, K.J., Jensen, K.H.: Maximum-likelihood estimation of unsaturated hydraulic parameters. *Journal of Hydrology*, 210(1-4):192-205, 1998.

[7] Leehter, Y., Sethares, W.A.: Nonlinear Parameter Estimation via the Genetic Algorithm. *IEEE Transactions on Signal Processing*, 42(4):914-926, 1994.

[8] Mikleš, J., Fikar, M.: *Process Modelling, Identification, and Control*. Springer Verlag, Berlin Heidelberg, 2007.

[9] Simon, D.: *Optimal State Estimation*. John Wiley&Sons, New Jersey, USA, 2006.

**Ing. Juraj Vörös**
**prof. Ing. Ján Mikleš, DrSc.**
**Ing. Ľuboš Čirka, PhD.**
Slovak University of Technology in Bratislava
Faculty of Chemical and Food Technology
Institute of Information Engineering, Automation, and Mathematics
Department of Information Engineering and Process Control
Radlinského 9
812 37 Bratislava
Tel.: +421 2 59325 351
E-mail: {juraj.voros, jan.mikles, lubos.cirka}@stuba.sk

# Real-Time Control of a Thermo-Optical Device Using Polynomial Approximation of MPC Scheme

Martin Herceg, Michal Kvasnica, Miroslav Fikar and Ľuboš Čirka

**Abstract**
This paper deals with real-time control of a thermo-optical device. A polynomial approximation of the optimal Model Predictive Control (MPC) feedback law is employed as a controller. Such an approximate controller enjoys the key benefits of MPC schemes, namely it provides all-time constraint satisfaction and closed-loop stability guarantees. The main advantage of the proposed approximation scheme is that it can be implemented in real time using very limited computational resources.

**Keywords:** model predictive control, multiparametric programming, polynomial approximation

## Introduction

MPC is a leading strategy in the control industry which offers optimal management of processes, and equally important, meets satisfaction of plant constraints [16], [6]. Based on the process model, MPC approach foresees the future behavior of the process and searches for the best possible control inputs. This process is repeated every time as new process measurements arrive. As the search for best inputs is achieved by solving an optimization problem, process constraints can be easily handled which makes MPC superior to traditional proportional-integral-derivative (PID) controllers.

Since the general introduction of predictive control by [8], numerous MPC techniques have been established and most of them ended as commercially available products [21]. This variety of MPC schemes can be separated in two groups, depending on how the particular optimization problem is solved. In the first group the optimization problem is solved on-line, that is, as the plant is under operation. This case applies to the majority of the processes in chemical industry with slow dynamics where there is enough time and computational resources for the optimization to terminate in time.

In the second group the optimization problem is solved *off-line*, that is, before plant's start-up. This approach is appealing especially for simple plants with fast dynamics, e.g. from electro-technical industry [9], [17]. This approach is often referred to as *explicit* MPC and for a recent survey see [1]. In the explicit MPC approach, most of the computational burden arising from optimization is shifted before the implementation phase, and the resulting controller is pre-computed for all admissible operating conditions. The controller takes of a form of a piecewise affine (PWA) function mapping the initial conditions to the optimal sequence of control inputs. The implementation of such controllers then consists of a mere evaluation of such a function for the currently measured value of plant states. However, the complexity of the solution (and hence the complexity of the implementation phase) grows, in the worst case, exponentially withe the problem size [24].

Up to date the best implementation scheme for evaluation of a PWA functions is the translation to a *binary search tree*, where the implementation complexity is logarithmic in the number of regions [22]. However, when considering application where the sampling frequency is very high, even the binary search tree algorithm can be of prohibitive computational complexity. An alternative approach based on polynomial approximation of the explicit MPC control law has been developed recently [14]. This method offers a suboptimal replacement of PWA function by a polynomial control law which significantly reduces requirements for storage and on-line evaluation. So far this method has been tested on a model of a DC-DC buck converter [18]. This paper presents a benchmark experiment for this method where the controlled plant is represented by a thermo-optical device with fast dynamics and rapid sampling.

## Device Description

The uDAQ28/LT thermal-optical system is an experimental device aimed primarily for education purposes [11]. The device allows for real time measurement and control of temperature and light intensity. It can be connected to a personal computer via an universal serial bus (USB) without requiring an input-output card (Fig. 1). Data acquisition and real-time control of the uDAQ28/LT device is carried out in the Matlab/Simulink environment which allows very easy manipulation with the device.

The plant represents a dynamical system which combines slow and fast dynamics. The slow process is characterized by a heat transfer and the fast process corresponds to light emission. Both processes are caused by an embedded light bulb which is controlled by an input voltage signal. In general, the plant is characterized by five inputs and eight outputs whereas only three controlled inputs and three measured outputs are of interest. A precise description of these signals is given in Tab. 1.

The construction of the device suggests offers two main control loops. The primal loop regulates the light intensity by manipulating the input voltage to the bulb (or input voltage to LED[1] diode). The second loop maintains the inner temperature in safety limits by manipulating the revolutions of a cooling fan. Presence of physical constraints on manipulated and controlled variables makes the control task challenging and the device has often been used for benchmark of constrained PID control approaches [10].

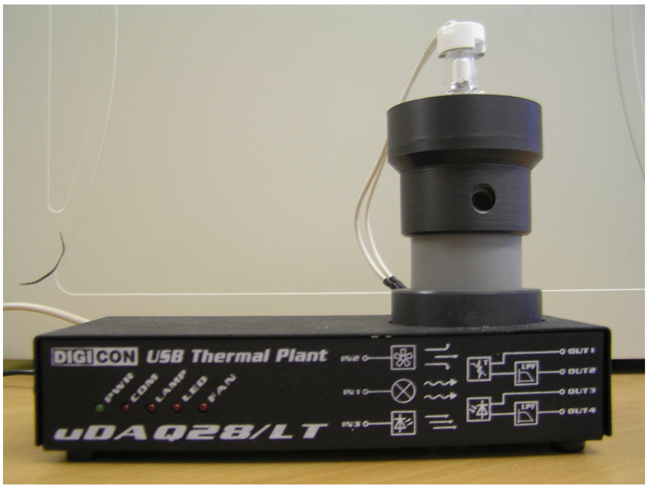


**Fig. 1 Front view on a thermo-optical device uDAQ28/LT.**

| Signal Name | Range |
|---|---|
| Input voltage to light bulb | 0-5 V |
| Input voltage to cooling fan | 0-5 V |
| Input voltage to LED | 0-5 V |
| Inner temperature | 0-100 deg C |
| Light intensity | not given |
| Revolutions of the cooling fan | 0-6000 rpm |

**Tab. 1 Description of measured and controlled signals.**

## Identification and PWA model

In the sequel, only the optical channel of the light bulb is considered. This decision is motivated by the fact that this channel is represented by a fast dynamics, which makes real-time implementation of a control system a challenging task. Due to very fast responses of the light channel, the sampling rate was selected the lowest admissible by Windows, i.e. $T_s = 0.05$ s.

Input-output relations of the optical channel have been identified with the help of IDTOOL Toolbox [23] as a second order discrete transfer function

$$G(z^{-1}) = \frac{bz^{-2}}{1 + a_1 z^{-2} + a_2 z^{-2}} \tag{1}$$

where $b$, $a_1$, $a_2$ are constant parameters and $z^{-1}$ is a discrete time delay operator [20]. IDTOOL toolbox contains the recursive least squares method of [12] which provides very good estimates of the unknown parameters. However, as transfer function is valid only locally, the identification was performed over four operating points and the results are summarized in Tab. 2.

|  | input | output | $b$ | $a_1$ | $a_2$ |
|---|---|---|---|---|---|
| (1) | 1.3 | 6.84 | 2.03 | -1.07 | 0.46 |
| (2) | 2.5 | 19.46 | 3.56 | -0.97 | 0.43 |
| (3) | 3.5 | 32.09 | 4.51 | -0.91 | 0.41 |
| (4) | 4.5 | 45.86 | 5.39 | -0.87 | 0.40 |

**Tab. 2 Identification data over four operation points.**

For the use in explicit MPC scheme, the input-output representation (1) is transformed to a discrete state-space model. It is achieved by introducing state variables with discrete time instant $k$, i.e. $v_1(k) = y(k-1)$, $v_2(k) = y(k-2)$ and the state space model reads

$$v_1(k+1) = -a_1 v_1(k) - a_2 v_2(k) + bw(k) \tag{2a}$$

$$v_2(k+1) = v_1(k) \tag{2b}$$

$$y(k) = v_2(k) \tag{2c}$$

In (2) $w(k)$ represents the input voltage applied directly to the plant and $y(k)$ is the measured output. Voltage input is constrained

$$w(k) \in [0, 5] \text{ V} \tag{3}$$

and the measured output lies inside the interval

$$y(k) \in [0, 55] \tag{4}$$

of light intensity units (are not given in the reference manual). The overall input-output behavior of the optical channel can be recovered by aggregation of the local linear models (2) which forms piecewise linear or PWA model. Here, the operating area is first split into regions and local linear models are assigned to each such region. The overall behavior of PWA model is then driven by switching between the locally valid models using logical IF-THEN rules. To perform partitioning of the operating area according to linearization points in Tab. 2, a Voronoi diagram [2] is constructed, which directly returns partitions of the state space as a sequence of convex polytopes. This operation was executed using one of the routines included in MPT toolbox [13] and it returned following regions:

$$R_1 = \{v(k) \mid 0 \le v_2(k) \le 13.15\} \tag{5a}$$

$$R_2 = \{v(k) \mid 13.15 \le v_2(k) \le 25.77\} \tag{5b}$$

$$R_3 = \{v(k) \mid 25.77 \le v_2(k) \le 38.97\} \tag{5c}$$

$$R_4 = \{v(k) \mid 38.97 \le v_2(k) \le 55\} \tag{5d}$$

To each of the regions (5), a corresponding local linear dynamics (2) is assigned, and it forms overall PWA model. Although PWA models are, in general, still non-linear, the underlying piecewise linearity allows for somewhat simpler controller design compared to full non-linear setups. Specifically, MPC problems based on PWA models can be solved explicitly, where the solution is obtained as a look-up table, easily implementable in real time.

The output from PWA model has been compared to the real measured output from the plant and the result is depicted in Fig. 2. For the given scenario PWA model follows correctly the plant's output, thus the accuracy of the model is verified. It can be noticed that at the beginning there is larger mismatch between the plant and the model. It is caused by physical properties of a filament in bulb which requires certain time to incandesce from a cold startup. As this phase

[1] Light Emitting Diode

is over, the PWA model correctly captures the optical channel of the plant and it can be employed for MPC design.

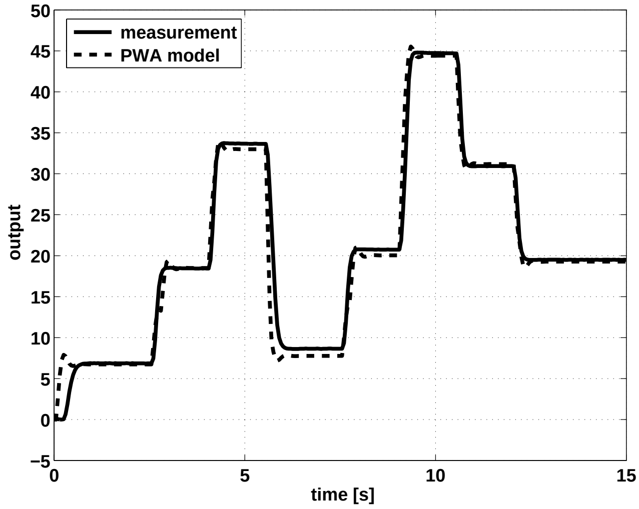


**Fig. 2 Verification of the PWA model.**

## Constrained Predictive Control

This section presents the explicit solution to MPC for the thermo-optical device uDAQ28/LT. Subsequently, the explicit solution is analyzed for stability purposes where the sets of stabilizing controllers are investigated. The polynomial approximation scheme is applied to obtain a real-time implementable controller.

### Prediction Model

In order to prevent numerical issues when employing the PWA model for MPC synthesis, it is advised to perform coordinate transformation and normalization. This can be achieved by introducing normalized variables $x_1$, $x_2$ and $u$ as follows:

$$x_1(k) = \frac{v_1(k) - v_{1,\text{ref}}}{\overline{v}_1} \tag{6a}$$

$$x_2(k) = \frac{v_2(k) - v_{2,\text{ref}}}{\overline{v}_2} \tag{6b}$$

$$u(k) = \frac{w(k) - w_{\text{ref}}}{\overline{w}} \tag{6c}$$

The suffix “ref” represent the desired steady state value, i.e.

$$v_{1,\text{ref}} = 32.09\,,\ v_{2,\text{ref}} = 32.09\,,\ w_{\text{ref}} = 3.5 \tag{7}$$

which is basically the linearization point of the third dynamics (see Tab. 2) and $\overline{v}_1 = 3.67$, $\overline{v}_2 = 3.67$, $\overline{v} = 0.5$ are constants. Applying the normalization, the transformed PWA model yields

$$f_{\text{PWA}}(x(k),u(k)) = A_i x(k) + B_i u(k) + f_i \tag{8}$$

where $i = 1, 2, 3, 4$ and state update matrices are given in Tab. 3. The state space model (8) is associated with the following regions

$$D_1 = \{x(k)\,|\,-8.75 \le x_2(k) \le -5.16\} \tag{9a}$$

$$D_2 = \{x(k)\,|\,-5.16 \le x_2(k) \le -1.72\} \tag{9b}$$

$$D_3 = \{x(k)\,|\,-1.72 \le x_2(k) \le 1.88\} \tag{9c}$$

$$D_4 = \{x(k)\,|\,1.88 \le x_2(k) \le 6.25\} \tag{9d}$$

| | | | | | | |
|---|---|---|---|---|---|---|
| $A_1$ | $B_1$ | $f_1$ | | $1.072\quad -0.464$<br>$1\quad 0$ | $0.277$<br>$0$ | $-1.492$<br>$0$ |
| $A_2$ | $B_2$ | $f_2$ | = | $0.969\quad -0.431$<br>$1\quad 0$ | $0.485$<br>$0$ | $-0.642$<br>$0$ |
| $A_3$ | $B_3$ | $f_3$ | | $0.913\quad -0.410$<br>$1\quad 0$ | $0.616$<br>$0$ | $0$<br>$0$ |
| $A_4$ | $B_4$ | $f_4$ | | $0.868\quad -0.402$<br>$1\quad 0$ | $0.735$<br>$0$ | $0.471$<br>$0$ |

**Tab. 3 Matrices of the normalized model (8).**

Besides the dynamics as in (8), the following constraints are assumed to be imposed on the behavior of the prediction model:

$$X = \{x(k)\,|\,-8.75 \le x_1(k) \le 6.25,\ -8.75 \le x_2(k) \le 6.25\} \tag{10a}$$

$$U = \{u(k)\,|\,-7 \le u(k) \le 3\} \tag{10b}$$

State constraints $X$ are derived from the operating range of light intensity (4) and input constraints $U$ represent the saturation limits (3).

### Control Problem

The aim of the control strategy is to find an optimal sequence of control inputs such that all system states are driven to a desired equilibrium. The equilibrium is given by the linearization point for the third PWA dynamics (8) and in the transformed coordinates (6) it is exactly the origin, i.e. $x_1(k) = 0$, $x_2(k) = 0$, $u(k) = 0$. Mathematically, the problem can be formulated as to find a sequence of future control moves $[u(k), u(k+1), \ldots, u(k+N-1)]$ up to horizon $N$ which steer the system states/input to the origin while satisfying constraints (10). More precisely,

$$\min_{u(k),\ \ldots,\ u(k+N-1)} \sum_{j=0}^{N-1} \left|Qx(k+j)\right|_1 + \left|Ru(k+j)\right|_1 \tag{11a}$$

$$\text{s.t.}\quad x(k+1) = f_{\text{PWA}}(x(k),u(k)) \tag{11b}$$

$$x(k+N) \in X_f \tag{11c}$$

$$x(k+j) \in X \tag{11d}$$

$$u(k+j) \in U \tag{11e}$$

where $x(k) = [x_1(k),\, x_2(k)]^T$ represents the state vector, the function $f_{\text{PWA}}(\cdot)$ describes the PWA model defined in (8) and the sets $X$, $U$ are the constraints on input and state variables given by (10). The set $X_f$ is introduced to obtain closed-loop stability guarantees [19]. The index 1 in the cost function (11a) denotes the 1-norm of given expression (sum of absolute values of vector components), matrices $Q$ and $R$ represent weighing factors.

Due to the presence of switching rules in the PWA model (8), the overall optimization problem (11) can be cast using additional binary variables as a mixed-integer linear program [4] and solved using *multiparametric* programming [5], implemented in freely available tools [13].

### Explicit MPC Synthesis

Solving problem (11) in a multiparametric fashion gives a closed form solution $u(k)$ as PWA function which maps $x(k)$ onto $U$. In particular, as was shown by [5], we have $u(k)=F_i x(k)+G_i$ if $x(k)\in P_i$ for $i=1,\ldots,n_{\text{reg}}$. Here, $P_i=\{x(k)\mid H_i x(k)\le K_i\}$ are polyhedral sets (regions) of the state-space. Similarly, a closed-form expression for the optimal cost function (11a) is again a PWA function of the state, i.e. $V(k)=M_i x(k)+L_i$ if $x(k)\in P_i$.

The problem (11) has been solved with parameters $N=\infty$, $Q=I$, $R=0.5$ with the help of the MPT toolbox [13]. The choice of $N=\infty$ guarantees that the obtained MPC feedback law will provide closed-loop stability [3].The resulting PWA control law builds a look-up table divided into 118 regions, defined in variables $x_1$, $x_2$, and these regions are plotted in Fig. 3. Over each one of these regions a local feedback law is defined as illustrates Fig. 4. Similarly, the cost function is shown in Fig. 5.

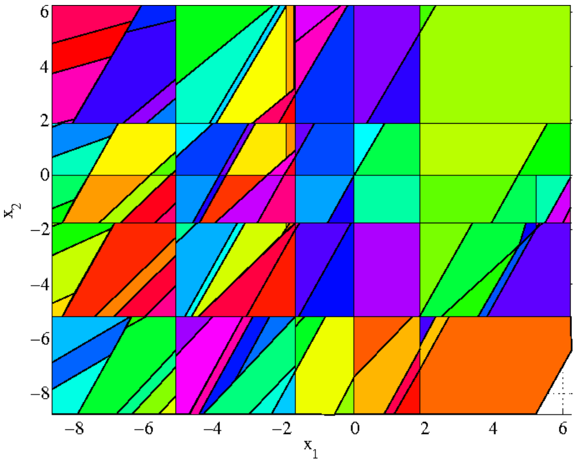


**Fig. 3 Regions of the look-up table.**

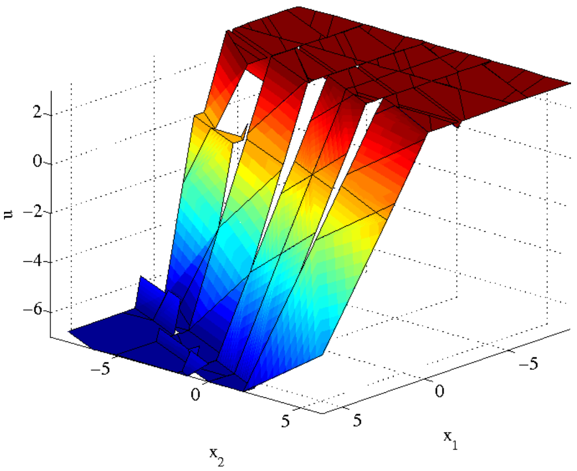


**Fig. 4 Local control laws over each region.**

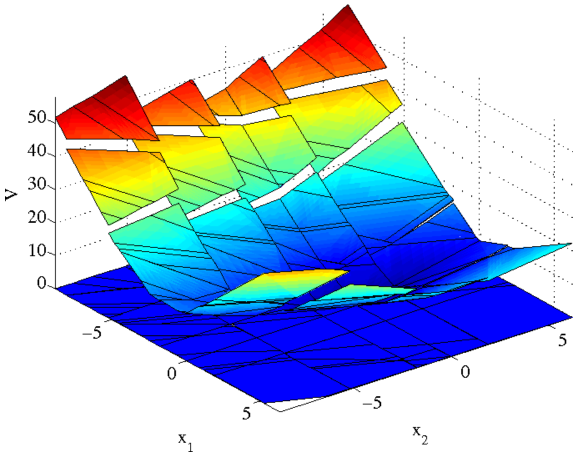


**Fig. 5 Value function.**

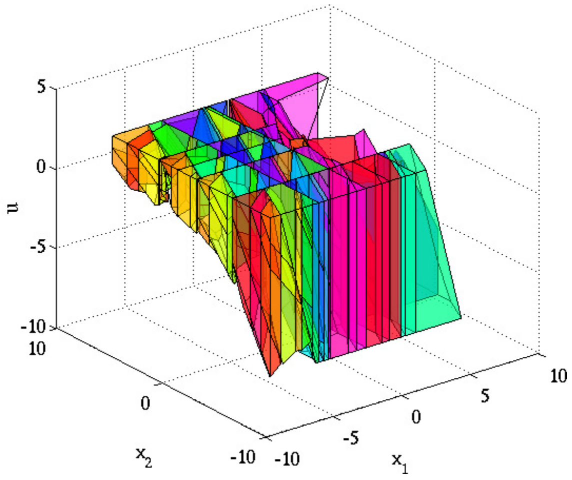


**Fig. 6 Stability tubes.**

Note that in the case of multiparametric solutions, the resulting PWA control law can be discontinuous (Fig. 4) and defined over a nonconvex set. This is a consequence of using binary variables to encode the IF-THEN rules which describe behavior of the PWA prediction model.

To implement the resulting look-up table in the on-line experiment, one has to store and evaluate the data. While storing part is limited by the available memory, the evaluation task is limited by the sampling time. The complexity of both tasks depend on the number of regions $n_{\text{reg}}$. Assuming that we have enough memory to store the look-up table, one have to still evaluate the PWA law. In fact, this task comprises of two steps

1. region identification
2. evaluation of PWA law

from which the first part consumes the most time. Even with the use of binary search tree algorithm, where the evaluation time is logarithmic in $n_{\text{reg}}$ [22], the scheme can still be prohibitive for implementation. Motivated by this fact, the goal is to apply the approximation scheme of [14] where the whole look-up table is replaced by one polynomial, which is very cheap to implement. To do so, we have to find the set of all perturbations of the control law under which the closed loop renders stability. This will be explained in the next section.

### Stability Tubes

As was shown by [7], the explicit feedback law described in the previous section is just one of many stabilizing feedbacks. Specifically, based on the explicit solution to (11), one can compute the family of controllers which all stabilize the control model (8). This family is characterized by sets in the state and input space and is called *stability tubes*.

**Definition 1** [7]: Let $V(x)$ be a Lyapunov function for the system (8) with $x \in X$ under a stabilizing controller $u(x) \in U$ . Then the set

$$S(V,\beta) = \left\{ \begin{aligned} & x(k) \in X, u(k) \in U, \\ & \qquad V(x(k+1)) - V(x(k)) \le -\beta(\|x(k)\|) \end{aligned} \right\} \tag{12}$$

is called a stability tube.

In other words, stability tubes are sets where the given Lyapunov value function $V(x)$ for system (8) decreases with a factor $\beta(\|x(k)\|) = \beta\|x(k)\|_p$ , $\beta > 0$ , $p = \{1, \infty\}$. Any control input from within the stability tube will ensure that the closed loop will be stable and constraints on variables won't be violated. Precisely,

**Theorem 1** [7]: Let the assumptions of Definition 1 be fulfilled. Then every control law $u(x(k))$ , $x(k) \in X$ , (also *any* sequence of control samples $u(k)$ ) fulfilling

$$(x(k), u(k)) \in S(V,\beta) \tag{13}$$

asymptotically stabilizes the system (8) to the origin, $\forall x(k) \in \bigcup_i P_i$ .

Algorithm for computing the stability tubes and all the relating routines are included in the MPT Toolbox. Firstly, one has to find a piecewise affine Lyapunov function for the closed loop system. As the optimal solution is computed with the infinite horizon, the value function in Fig. 5 is a Lyapunov function. Secondly, one can apply routines for computing the stability tubes and the result is a collection of polyhedrals in the joint $x$ - $u$ space and it is shown in Fig. 6.

### Polynomial Approximation

Using the approximation scheme of [14], the goal is to find a polynomial control law of the form

$$\mu(x) = [a_{11}, a_{12}]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [a_{21}, a_{22}]\begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix} + [a_{31}, a_{32}]\begin{bmatrix} x_1^3 \\ x_2^3 \end{bmatrix} \tag{14}$$

which, when applied as a state feedback, guarantees closed-loop stability and constraint satisfaction. Theorem 1 provides a sufficient condition for existence of such a polynomial feedback law in the sense that if $(x, \mu(x)) \in S(V,\beta)$ , $\forall x \in \bigcup_i P_i$ then $\mu(x)$ will provide closed-loop stability and constraint satisfaction. Therefore the search for suitable polynomial coefficients $a_{ij}$ of (14) can be cast as the following optimization problem:

$$\min_{a_{11}, \ldots, a_{32}} \sum_j \|u(x) - \mu(x)\| \tag{15a}$$

$$\text{s.t.} \qquad (x, \mu(x)) \in S(V,\beta) \tag{15b}$$

From all possible choices of $\mu(x)$ which satisfy (15b), cost function (15a) is used to select the coefficients which provide best approximation of the optimal feedback law $u(x)$ . As was shown in [14], optimization problem (15) can be formulated as a semidefinite programming problem, which can be solved using off-the-shelf tools.

The main advantage of the polynomial feedback law (14), compared to the MPC controller based on evaluating PWA feedback law, is reduction of the total implementation and storage cost. On the storage side, only the coefficients $a_{ij}$ need to be recorded in the memory, compared to storing the regions $P_i$ and the feedback laws $F_i$ and $G_i$ for PWA feedback law. The on-line implementation cost is also greatly reduced, as only polynomial evaluation for a given $x$ need to be performed to obtain a stabilizing control action.

The approximation scheme has been applied to obtain polynomial control law of type (14) with help of YALMIP [15]. Computed coefficients are given in Tab. 4.

| | |
|---|---|
| $a_{11}$ , $a_{12}$ | -0.8718, -0.0007 |
| $a_{21}$ , $a_{21}$ | -0.0519, 0.0004 |
| $a_{31}$ , $a_{32}$ | 0.0019, 0.0001 |

**Tab. 4 Coefficients of the approximated polynomial.**

Illustration of the approximation scheme is shown in Fig. 7 which represents a cross-section in stability tubes along the coordinate $x_2 = 0$ . The polyhedral sets in Fig. 7 demonstrate the space of the stability tubes where there exist a stabilizing control law according to Theorem 1. Inside this space the approximated polynomial (14) has been fitted and it is shown in Fig. 7 with a dashed line while the optimal control law is depicted with solid line.

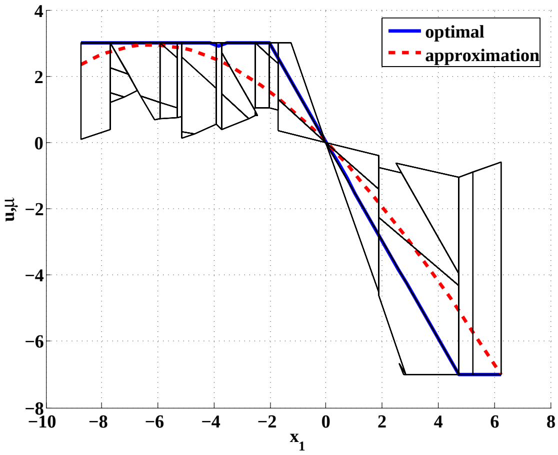


**Fig. 7 Cross-section of the control laws through $x_2 = 0$ .**

## Real-Time Implementation

In this section computational requirements are evaluated for the optimal and approximated controller. Both controllers are applied in the real-time experiment and measured performance is discussed.

### Computational Demands

Implementation of the optimal controller in the on-line experiment is limited by the sampling time $T_s = 0.05$ s. If the look-up table, obtained previously and consisting of 118 regions, is stored and evaluated using the binary search tree algorithm [22], the number of floating point operations per second (FLOPS) which are required to evaluate such a controller for a given initial condition is at most 41. The memory requirements are 2832 bytes for the control law and 1536 bytes for the search tree which gives a total of 4368 bytes.

In the polynomial approximation scheme, the number of FLOPS depend on the degree of approximated polynomial and on the polynomial degree. By considering the polynomial (14) with degree of three, the upper bound for evaluation FLOPS is 14, less than a half of the runtime for the binary search tree. More prominent, however, is the drop in memory consumption. As state above, the explicit MPC solution with 118 regions requires 4368 bytes of memory storage, while to store the polynomial feedback law (14), mere 24 bytes of memory are required (6 polynomial coefficients, each of them consuming 4 bytes when represented as floating point numbers).

## Experimental Data

The optimal explicit MPC controller as well as the polynomial feedback strategy have been implemented in real time and obtained results are shown in Fig. 8, Fig. 9 and Fig. 10. The plots represent the transition from the initial condition $x_0 = [-8.7, -8.7]^T$ to the origin. Input signal generated by the optimal controller immediately jumps to the upper limit and then gently approaches the origin. In the polynomial controller this effect is different, the controller is slightly slower, but the same stabilizing effect is achieved. State and input profiles converge to desired steady state, hence the control objective was met with both approaches. It is interesting to note that a polynomial controller acts better (in the sense of the selected performance criterion (11a)) than the optimal one. In particular, (11a) evaluates to 146.34 when the optimal MPC controller is used as a feedback, compared to value of (11a) amounting to 142.96 for the case where the polynomial controller was used. This small difference can be attributed to the fact that the optimal controller is more sensitive to changes of the states. Nevertheless, the difference is small enough to say that both controllers share roughly the same performance while the approximated controller is significantly cheaper than the optimal one.

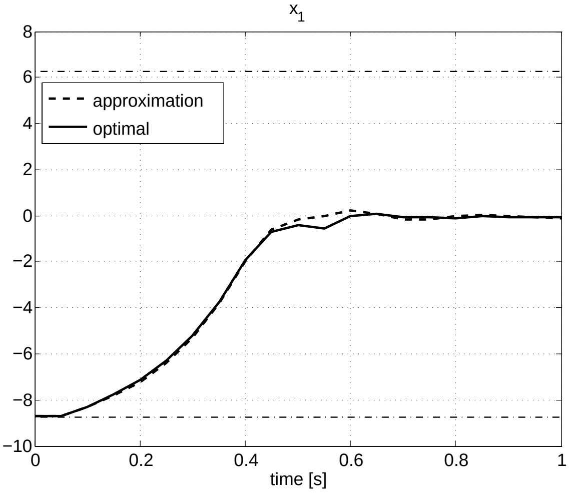


**Fig. 8 Profiles of the state variable $x_1$ .**

Performance of both controllers has not been tested on disturbance attenuation because this effect cannot be fully compensated by any of the used controllers since they do not contain an integration part. Moreover, these effects are too small to satisfactory evaluate the performance of both controllers while showing their advantages (e.g. constraint satisfaction).

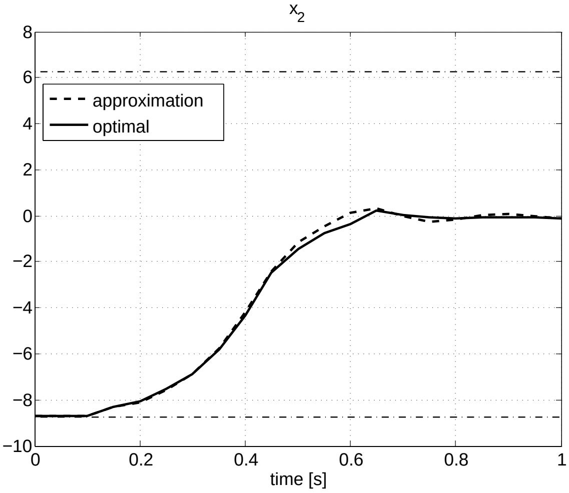


**Fig. 9 Profiles of the state variable $x_2$ .**

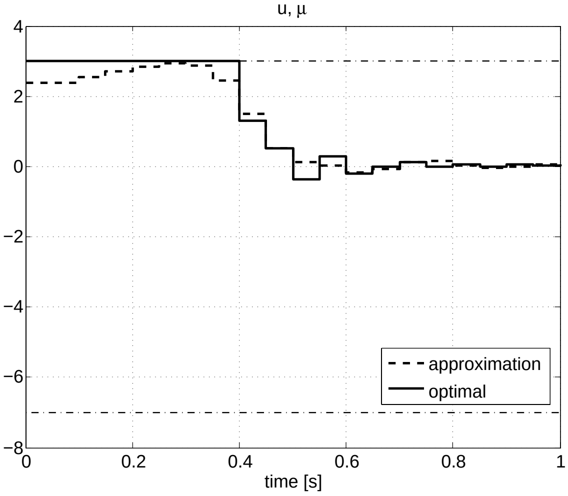


**Fig. 10 Input profiles.**

## Conclusion

Main motivation of this paper was to demonstrate a cheap alternative to explicit MPC scheme based on polynomial approximation of the optimal feedback law. Control of the optical channel of uDAQ28/LT device is considered as a benchmark example to polynomial approximation scheme of [14]. The process is identified as a second order linear discrete time system around four operating points. Based on the identification data, PWA model is constructed and deployed for MPC design. The MPC problem is solved in the multiparametric fashion, i.e. precomputing the controller for whole possible operating conditions, and the result is stored as a look-up table. The properties of the explicit solution are further exploited and a family of all stabilizing controllers is constructed. From this family, one controller of a special polynomial structure has been selected, which implementation cost is the cheapest. The polynomial controller has been experimentally tested in the closed loop, and has shown good results.

## Acknowledgment

*The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09 and 1/4055/07. This work was supported by the Slovak Research and*

*Development Agency under the contracts No. VV-0029- 07 and No. LPP-0092-07.*

## References

[1] ALESSIO A, BEMPORAD A:. A survey on explicit model predictive control. In L. Magni, D. Raimondo, and F. Allgöwer, editors, *International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, Pavia, Italy, 2008.

[2] AURENHAMMER F.: Voronoi diagramns – survey of a fundamental geometric data structure. *ACM Computing Surverys*, 3(23):345–405, 1991.

[3] BAOTIĆ M., CHRISTOPHERSEN F.J., MORARI M.: Constrained Optimal Control of Hybrid Systems With a Linear Performance Index. *IEEE Trans. on Automatic Control*, 51(12):1903-1919, December 2006.

[4] BEMPORAD A., MORARI M.: Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407-427, March 1999.

[5] BORRELLI F.: Constrained Optimal Control of Linear and Hybrid Systems. In *Lecture Notes in Control and Information Sciences*, volume 290. Springer, 2003.

[6] CAMACHO E.F., BORDONS C.: *Model Predictive Control*. Springer, 1999.

[7] CHRISTOPHERSEN F.J.: Optimal Control of Constrained Piecewise Affine Systems. In *Lecture Notes in Control and Information Sciences*, volume 359. Springer-Verlag, 2007.

[8] CLARKE D.W, MOHTADI C., TUFFS P.S.: Generalized predictive control – Parts I-II. *Automatica*, 23(2), 1987.

[9] GEYER T., PAPAFOTIOU G., MORARI M.: Hybrid Model Predictive Control of the Step-Down DC-DC Converter. *IEEE Transactions on Control Systems Technology*, 16(6):1112–1124, November 2008.

[10] HUBA M., VRANČIČ D.: Constrained control of the plant with two different modes. In J. Mikleš, M. Fikar, and M. Kvasnica, editors, *16th Int. Conf. Process Control*, pages paper Le–Tu–5, 205p.pdf, 2007.

[11] HUBA M., KURČÍK P., KAMENSKÝ M.: Thermo-optical device uDAQ28/LT. STU Bratislava, Illkovičova 3, Bratislava, 2006. In Slovak.

[12] KULHAVÝ R., KÁRNÝ M.: Tracking of slowly varying parameters by directional forgetting. In *Proceedings of the 9th IFAC World Congres*, Budapest, Hungary, 1984.

[13] KVASNICA M., GRIEDER P., BAOTIĆ M., MORARI M.: Multi-Parametric Toolbox (MPT). In *Hybrid Systems: Computation and Control*, pages 448–462, March 2004. URL. http://control.ee.ethz.ch/~mpt.

[14] KVASNICA M., CHRISTOPHERSEN F.J., HERCEG M., FIKAR M.: Polynomial approximation of closed-form MPC for piecewise affine systems. In *Proceedings of the 17th IFAC World Congres*, pages 3877–3882, Seoul, South Korea, 2008.

[15] LÖFBERG J.: Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. http://control.ee.ethz.ch/~joloef/yalmip.php.

[16] MACIEJOWSKI J.M.: *Predictive Control with Constraints*. Prentice Hall, 2002.

[17] MARIETHOZ S., BECCUTI A.G., PAPAFOTIOU G., MORARI M.: Sensorless explicit model predictive control of the DC-DC buck converter with inductor current limitation. In *IEEE Applied Power Electronics Conference*, February 2008a.

[18] MARIETHOZ S., HERCEG M., KVASNICA M.: Model Predictive Control of buck DC-DC converter with nonlinear inductor. In *IEEE COMPEL workshop on control and modeling for power electronics*, Zurich, Switzerland, August 2008b.

[19] MAYNE D.Q., RAWLINGS J.B., RAO C.V., SCOKAERT P.O.M.: Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[20] MIKLEŠ J., FIKAR M.: *Process Modelling, Identification, and Control*. Springer Verlag, Berlin Heidelberg, 2007.

[21] QIN S.J., BADGEWELL T.A.: A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[22] TØNDEL P., JOHANSEN T.A., BEMPORAD A: Evaluation of Piecewise Affine Control via Binary Search Tree. *Automatica*, 39(5):945–950, May 2003.

[23] ČIRKA Ľ., FIKAR M., PETRUŠ P.: IDTOOL 4.0 - A Dynamical System Identification Toolbox for MATLAB/Simulink. In *14th Annual Conference Proceedings: Technical Computing Prague 2006*, pages 29–29. The MathWorks, Inc. & HUMUSOFT s.r.o. & Institute of Chemical Technology in Prague, October 2006. URL http://www.kirp.chtf.stuba.sk/~cirka/idtool/?IDTOOL.

[24] ZEILINGER M., JONES C.N., MORARI M.: Real-time suboptimal Model Predictive Control using a combination of Explicit MPC and Online Computation. In *Conference on Decisionand Control*, CDC, Cancun, Mexico, December 2008.

**Ing. Martin Herceg**

Škola: Slovenská Technická Univerzita v Bratislave
Fakulta/Ústav: Fakulta Chemickej a Potravinárskej
Technológie, Ústav Automatizácie, Informatizácie
a Matematiky
Katedra: Oddelenie Informatizácie a Riadenia Procesov
Ulica: Radlinského 9
PSČ a Mesto: 812 37 Bratislava
Tel.: +421 2 52495269
Fax: +421 2 52496469

E-mail: martin.herceg@stuba.sk

# Predictive control of pressure swing adsorption

Michael Mulholland and M.A. Latifi

**Abstract**
Pressure swing adsorption requires a repeated cycle of four steps. The periods of these steps, or other defined terminal conditions, determine the rate and quality of the product, and its cost. In transient situations such as upsets or grade changes, it is not intuitively obvious how the steps should be progressively altered to bring the plant to the desired operating point in an optimal fashion. The present work considers the problem of real-time maximization of the production of a single adsorber, and maintaining a setpoint concentration in its product receiving vessel. In a modelling exercise, these objectives have been met using predictive control based on completion of the present step, plus two full future cycles to reduce the end-effect. The approach sought to be fast and robust by suitable linearisation of the system. This allowed MILP solution in the mixed logical dynamical (MLD) framework as a mixed integer dynamic optimisation (MIDO). However, this problem was ultimately solved faster and more reliably by testing all combinations for constraint violations and the objective value.

**Keywords:** PSA, hybrid systems, dynamic optimisation, predictive control

## Introduction

An increasing range of adsorbent materials is extending the use of pressure swing adsorption (PSA) in the separation of gas mixtures. These materials are designed to selectively adsorb one component from a mixture. As in vapour-liquid equilibrium, the equilibrium quantity of this adsorbed component in the solid phase increases with its partial pressure in the gas phase. Thus the solid can be used to adsorb the component at high pressure, and it can be "regenerated" by expelling the adsorbed species at low pressure. In air separation, $N_2$ is selectively adsorbed, leaving an $O_2$-rich product stream. A number of adsorbers can be arranged to work in complementary cycles so as to smooth out production flow and the use of common resources.

## 1. Operation

The present analysis will focus on a single adsorber with a product storage vessel as in figure 1. Four distinct steps, comprising the Skarstrom cycle, are required:

(1) **pressurisation:**
A open; B,C & D closed

(2) **adsorption at high pressure:**
A & B open; C & D closed

(3) **depressurisation:**
D open; A,B & C closed

(4) **purge at low pressure:**
C & D open; A & B closed

During step 2, a high purity product can be obtained, particularly if some of the product itself is used in step 4 for purging, as is shown here.

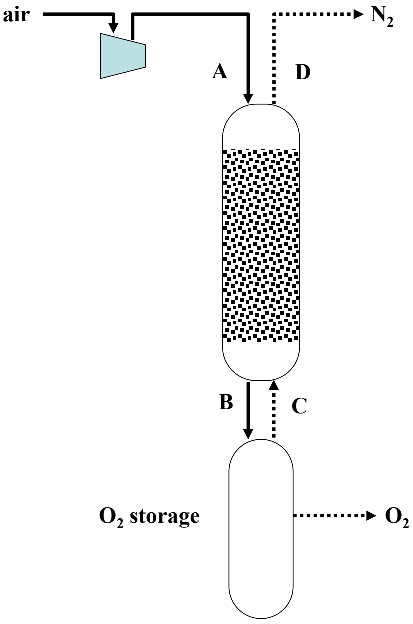


**Fig.1 A basic pressure swing adsorption configuration for air separation**

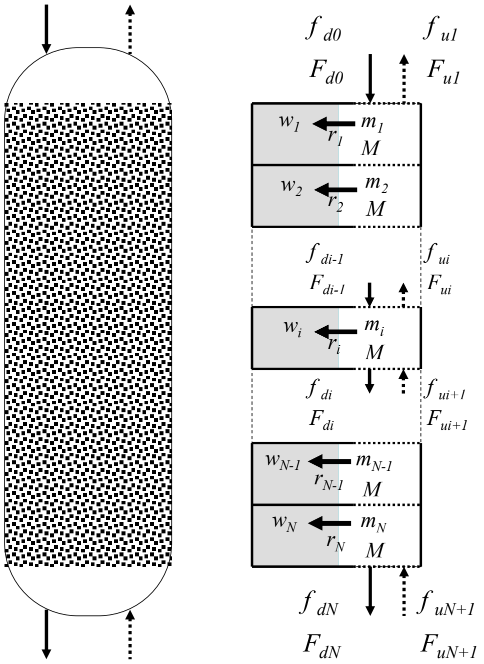
The mechanism by which a high-purity product is obtained is not entirely self-evident. It is in fact achieved by developing a suitable composition profile in the solid phase which acts to "screen" the down-ward moving air in step 2. That profile will of course oscillate through each full cycle of four steps, but the so-called "cyclic steady state" (CSS) is achieved once a fixed associated profile arises at the end of each step. Even with fixed sequencing of the valves A,B,C and D (ie. fixed periods for each step), the approach to CSS may take many cycles.
In figure 2 an adsorber is represented as discretised into *N* compartments in series. If thermal effects are neglected, this

system is defined by 2×*N*+1 states. These are the moles of the adsorbed species in the gas phase ($m_i$) and the solid phase ($w_i$) in each compartment *i*, as well as the total number of moles ($M$) in the gas phase of a compartment. For a uniformly packed bed with no frictional losses one notes that *M* is the same in all compartments, and the total number of gas moles in the system is *N*×*M* which is clearly proportional to the pressure. To re-iterate the comment above regarding the CSS, the repeated cycles needed to achieve CSS are required to achieve the supporting profiles of $m_i$ and $w_i$ , with the latter determining longer settling times as the adsorbent capacity increases. If one imagines the control problem associated with the unsteady process, one thus foresees several major hurdles:

(a) high number of states
(b) few measureable states
(c) hybrid (switched and continuous)
(d) long time-constants

To date, most of the work aimed at optimising PSA operation has focused on the optimal "positioning" on the CSS cycle. The cycle can be positioned by choosing a particular set of four times, one for each of the Skarstrom steps. Alternatively, it can be positioned by choice of a particular set of heuristic rules, eg.

[1] pressurise until pressure reaches $P_{max}$
[2] adsorb until a total of $M_{FEED}$ moles of feed have been introduced in this step
[3] depressurise until pressure reaches $P_{min}$
[4] purge until a total of $M_{PURGE}$ moles of product have been returned in this step



$$r_i = \frac{dw_i}{dt} = k\left[m_i - cw_i\right]$$

**Fig.2 Discrete representation of states in a pressure-swing adsorber**

Models have been quite detailed, accounting for thermal effects, pressure losses, nonlinear adsorption isotherms, energy, etc, and optimisations have aimed at overall economic operation. These are large non-linear optimisation problems, such as tackled in [1], [2] and [3]. Indeed, [1] does not look for a convergence in time, but rather formulates the optimisation problem around a single cycle, including in the objective function a minimisation of the deviation between the states at the beginning and end of the four steps of the cycle.

The relatively small amount of work on the unsteady-state quite likely arises from the difficulties (a),(b),(c) and (d) mentioned above. In [4] the authors developed a nonlinear distributed parameter observer for PSA based on a Luenberger arrangement for error feedback. In [5] these same authors present a control scheme which has two parts: A feed-forward section sets the time-periods for each step of the cycle, based on the sensitivity of production and purity predicted offline at CSS. A degree of adjustment of the time-periods is superimposed for correction of measured quality by PID feedback. This type of approach is extended in [6] with the feedforward based on the inversion of a reduced-order model, for example a Hammerstein representation.

One can reflect for a moment on what advantages might accrue from dynamic feedback control of PSA. A distributed process with long time-constants is inherently difficult to adjust, so manual operations are likely to be determined by heuristic criteria such as above. In start-up, shut-down or recovery from an upset, these are likely to be conservative and inefficient. What one seeks is an optimal strategy to bring the process from its current point to one which ensures product quality and rate, at minimum cost, possibly in coordination with other adsorbers. With this aim, the work below investigates the possibility of using robust linear tools in an optimal predictive control format, initially applied to a single adsorber and product tank.

## 2. Model

The adsorber is modelled as a series of *N* mixed compartments as in figure 2. Typical values are used for air separation, using a linear equilibrium relationship for the $N_2$ ($m^*$=$cw$) and ignoring the small amount of $O_2$ adsorbed. Pressure losses through the bed and thermal effects are likewise neglected. In the equations, *M* and *F* respectively represent the total gas inventory of a compartment, and the total gas flow, whereas *m* and *f* refer only to the species which is being adsorbed ($N_2$). Flows are divided into "downward" (*d*) and "upward" (*u*), of which one or the other will be zero depending on the step of the cycle.

$$\begin{aligned}
\frac{dM}{dt} &= \frac{1}{N}\left\{\begin{matrix} F_{d,0} + F_{u,N+1} - F_{d,N} - F_{u,1} \\ -\sum_{i=1}^{N} k\left[m_i - cw_i\right] \end{matrix}\right\} \\
\frac{dm_i}{dt} &= f_{d,i-1} + f_{u,i+1} - f_{d,i} - f_{u,i} - k\left[m_i - cw_i\right] \qquad (1) \\
\frac{dw_i}{dt} &= k\left[m_i - cw_i\right]
\end{aligned}$$

The only nonlinearity arises as the requirement that the effluent composition from a compartment obeys the following equations for downward or upward flow respectively.

$$\frac{f_{d,i}}{F_{d,i}} = \frac{m_i}{M} \quad \text{or} \quad \frac{f_{u,i}}{F_{u,i}} = \frac{m_i}{M} \qquad (2)$$

This was linearised using deviations (Δ) from an estimated operating point (‘)

$$\frac{f'+\Delta f}{F'+\Delta F}=\frac{m'+\Delta m}{M'+\Delta M} \qquad (3)$$

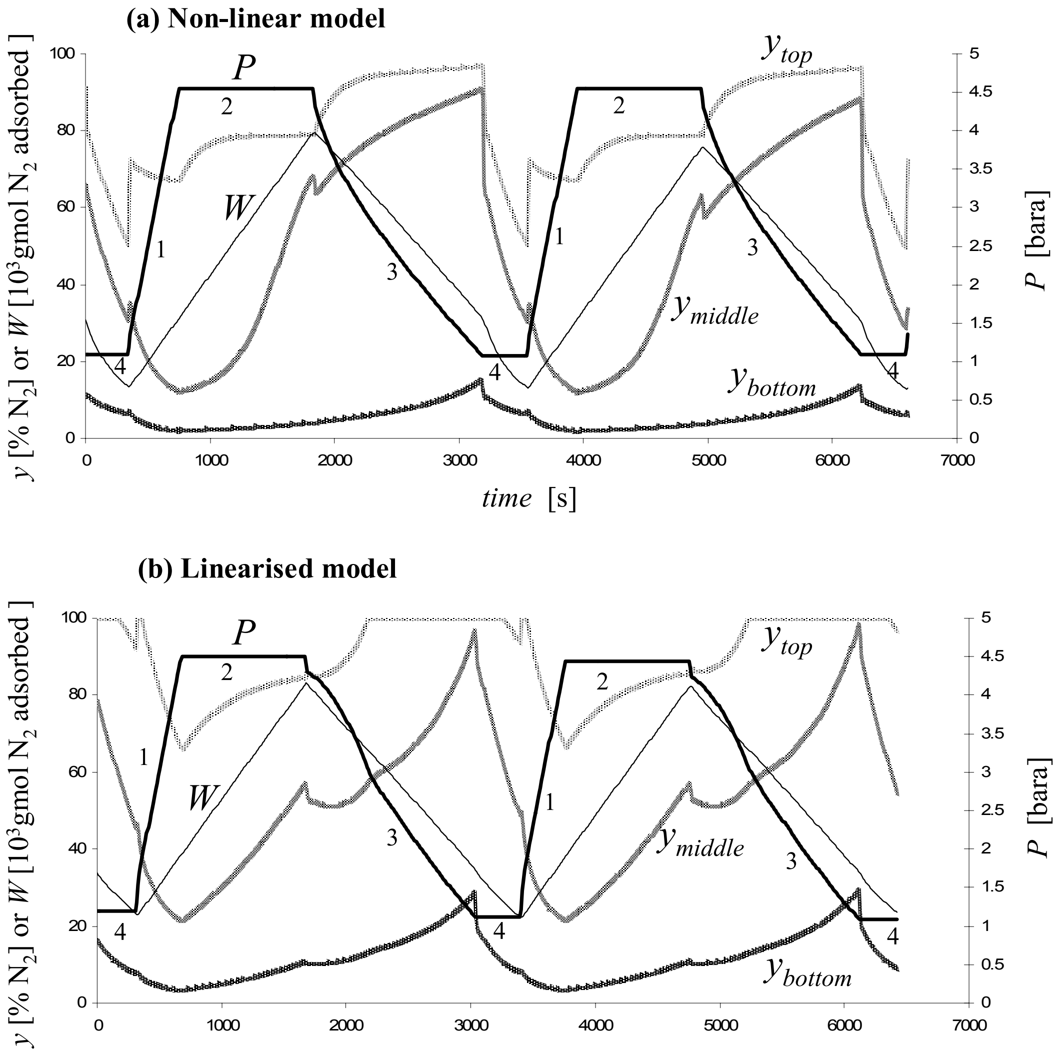
and neglect of the deviation products. The total flow profile $F_i$ in either direction (ie. $F_i'$ , $i$ = 1,...$N$) can be estimated relatively closely, as can the total number of moles in a compartment $M$. Estimates of the flow and inventory profiles of the *adsorbed* species, $f_i$’ and $m_i$’, were obtained by multiplying $F_i'$ and $M$ by a composition $y_{av}$ appropriate to each Skarstrom step. The linearisation conditions are thus summarised as follows:

[1] **pressurisation:** $M'$ at $P_{max}$ ; $F'$ reducing linearly from $F_{feed}$ at top to 0 at bottom; $y_{av}$ at 0
[2] **adsorption at high pressure:** $M'$ at $P_{max}$ ; $F'$ reducing linearly from $F_{feed}$ at top to $F_{feed}$ - $f_{feed}$ at bottom ; $y_{av}$ at 0
[3] **depressurisation:** $M'$ at 75% $P_{max}$ + 25% $P_{min}$ ; $F'$ increasing linearly from 0 at bottom to a flow $F_{depress}$ at top ; $y_{av}$ at (1+$y_{feed}$)/2
[4] **purge at low pressure:** $M'$ at $P_{min}$ ; $F'$ profile constant at the purge gas flow rate $F_{purge}$ ; $y_{av}$ at $y_{purge}$

In this way, a discrete linear model for the 2$N$+1 states is constructed for a unique $\Delta t_j$ suited to each of the Skarstrom steps:

[1] **pressurisation:**
$$\boldsymbol{x}(t+\Delta t_1)=\boldsymbol{A}_1\,\boldsymbol{x}(t)+\boldsymbol{b}_1 \quad \text{with} \quad \Delta t_1=8\,s \qquad (4)$$
[2] **adsorption at high pressure:**
$$\boldsymbol{x}(t+\Delta t_2)=\boldsymbol{A}_2\,\boldsymbol{x}(t)+\boldsymbol{b}_2 \quad \text{with} \quad \Delta t_2=16\,s \qquad (5)$$
[3] **depressurisation:**
$$\boldsymbol{x}(t+\Delta t_3)=\boldsymbol{A}_3\,\boldsymbol{x}(t)+\boldsymbol{b}_3 \quad \text{with} \quad \Delta t_3=16\,s \qquad (6)$$
[4] **purge at low pressure:**
$$\boldsymbol{x}(t+\Delta t_4)=\boldsymbol{A}_4\,\boldsymbol{x}(t)+\boldsymbol{b}_4 \quad \text{with} \quad \Delta t_4=8\,s \qquad (7)$$

Should the flow settings $F_{feed}$ , $F_{purge}$ , $F_{depress}$ (or feed or purge stream compositions) change, the corresponding matrix $\boldsymbol{A}_j$ (new linearisation point) and vector $\boldsymbol{b}_j$ (new process input) are updated. Exit flows are determined automatically in the solutions for the constant-pressure steps 2 and 4.


(a) Non-linear model

(b) Linearised model

**Fig.3 Comparison of *(a) Nonlinear Model* and *(b) Linearised Model* predictions for Skarstrom cycles using the same switching criteria (Section 1)**

In figure 3 the non-linear and linearised model predictions, both using 9 compartments, are compared for Skarstrom cycles determined by the heuristic switching rules in section 1 ($P_{min}$ = 1 bara; $P_{max}$ = 4.5 bara; Total Air used in step 2 =

$6\times10^4$ gmol; Total product used for Purge in step 4 = $1\times10^4$ gmol). For this analysis the composistion of the "Product" used for purging was fixed at 5% $N_2$. Actual cumulative product compositions were in fact 2.8% $N_2$ for the non-linear model, and 5.2% $N_2$ for the linear model, with about $1.5\times10^4$ gmol of product being made on each cycle (ie. reflux ratio = 2).

## 3. Predictive control

A single PSA unit does not appear to offer a lot of scope for dynamic optimisation. The compositions and achievable flows of the *feed air*, and the *product used for purging*, are likely to be fixed. The only control freedom left thereafter is the length of each step of the Skarstrom cycle, which could equivalently be set by heuristic rules (eg. varying the target "breakthrough" composition of the product stream in step 4). Even if the flow rates of the feed and purge streams could be varied, equation (1) shows that this is equivalent to varying the time intervals if the adsorption is not rate-controlling (*k* large enough).

Thus a single PSA unit offers just these four adjustments. In section 1 it was mentioned how these are manipulated for optimisation of the CSS. In contrast, the motive for *unsteady-state* optimisation lies in dealing with extraneous disturbances or objectives. Thus, if the system state finds itself away from its optimal value (after a disturbance, or during start-up/shut-down), it needs to be guided back in an optimal fashion. Additionally, there will be requirements to maintain a set-point inventory within the product storage vessel, and to keep it close to a set-point composition. Even for a single PSA unit, this offers interesting scope for strategic manipulation of its Skarstrom step lengths. With multiple PSA units feeding and drawing from the same product storage vessel, the problem becomes much more complex, and it may be anticipated that an *integer programming* (IP) approach for unit coordination will truly be advantageous here.

Bearing the above unsteady-state optimisation objectives in mind, the present work thus seeks to use robust linear system tools for constrained optimisation of one or more complete future Skarstrom cycles, by correct choice of the first and subsequent switching points for the steps. Only the first switch is implemented, once it is shown to be "due" (immediate). Of course, the remaining switches are only evaluated as part of the overall optimisation, and not for use. Though the step lengths could be treated as continuous variables for optimisation, a "modulated" approach is rather used, entailing a *choice* of one of several distinct periods for each step. Apart from facilitating an IP solution, this will in due course also allow coordination with other PSA units.

### 3.1 Optimal predictive control with constraints

In predictive control, one is generally aiming to make the best choice of a series of control decisions which affect the system output up to a defined future time-horizon. Only the first choice is actually implemented, before the entire optimisation is repeated on the next controller time-step. If some of the choices are discrete (eg. gears of a car), or indeed, as seen above, if there are system behavioural changes, integer variables enter the problem, and one has a *mixed integer dynamic optimisation* (MIDO) problem.

Consider the problem of a single PSA column supplying a purified gas to a storage vessel, from which users draw their requirements at arbitrary rates (figure 1). Removal of $N_2$ from air to provide an $O_2$ supply will be the example. The objective will be to maintain a desired $O_2$-rich inventory in the storage vessel, and to control its composition at a given set-point. Discretionary situations arise, for example, if demand is low, and composition is poor - in which case production can be reduced and a greater proportion of the $N_2$ removed, to raise the $O_2$ concentration in the storage vessel.

In the real-time situation, the controller is cycling asynchronously at its own time-interval. In the present case this is 10s. It does not need to match the $\Delta t_j$ of any of the Skarstrom steps *j* because the entire optimisation calculation is repeated *a priori* on each controller time-step. What is important to the control algorithm is to know the system *state* at this time. A first step was thus to develop a *state observer*. A Kalman filter based on the linearised models in equations (4) to (7), changing in sequence, was able to provide good estimates of the $2\times N+1$ state values using just three "measurement" outputs of the original *non-linear* model [ (i) pressure *P*; (ii) product outflow composition during step 2; and (iii) purge outflow composition during steps 3 and 4 ].

Apart from these state values, the predictive control algorithm of course needs to know which of the four possible steps of the Skarstrom cycle is presently being conducted. (Historical information - eg. how long it has been in this step - is not required). A look-up table indicates the required future sequence for completion of an entire Skarstrom cycle (constrained), followed by a repeat full cycle with the same step lengths (unconstrained):

[1] **pressurisation:**
*complete* 1 then do 2→3→4→1 , 2→3→4→1
[2] **adsorption at high pressure:**
*complete* 2 then do 3→4→1→2 , 3→4→1→2
[3] **depressurisation:**
*complete* 3 then do 4→1→2→3 , 4→1→2→3
[4] **purge at low pressure:**
*complete* 4 then do 1→2→3→4 , 1→2→3→4

The identified future sequence is then the basis of the optimisation. It amounts to a choice of the number of intervals $\Delta t_j$ to spend in each of the Skarstrom steps *j* (figure 4). The result is five separate interval counts. Steps occurring *after* the production step 4 would appear to play a neutral or negative role (eg. use of Product for purging). Thus the objective function used here is based on one further repetition of the cycle (*n*=1) to reduce such "end-effects". The computational load is reduced by forcing the "copies" to use the Skarstrom step lengths of the first full cycle.

The main interest is in whether the intervals left in the first (partial) step add up to less than the controller time-step. In that case the controller must take action *now* by switching to the next Skarstrom step.

The easiest way to structure the optimisation is to constrain the operation between liberal bounds such as determined by the heuristic rules in section 1 - viz, maximum and minimum pressure, and maximum total amounts of Feed and Product to be used in the adsorption and purge steps respectively. These bounds on their own determine a default Skarstrom cycle. The purpose of the optimisation then is to bring the terminal points of the cycle inwards in order to maximise the objective function.

The optimisation scheme is represented schematically in figure 4 in terms of pressure. The variable constraints are only considered at the *end* of each step - ie. at the switching point. This will however not be problematic, since the variables associated with the specified constraints all vary monotonically in each step (figure 3). An important motivation for this scheme is that if necessary, computation can be reduced by narrowing the range of choice in each step, eg. close to the number of intervals determined for that mode in the preceding optimisation. Using a centred search range, migration will still occur at successive controller steps. However, the first partial step must always extend down to 1 interval owing to its function in determining the switching time. Figure 5 illustrates how a selection is made on each step from a limited number of "models" for that step, each representing a different number of intervals – ie. a different time-span.

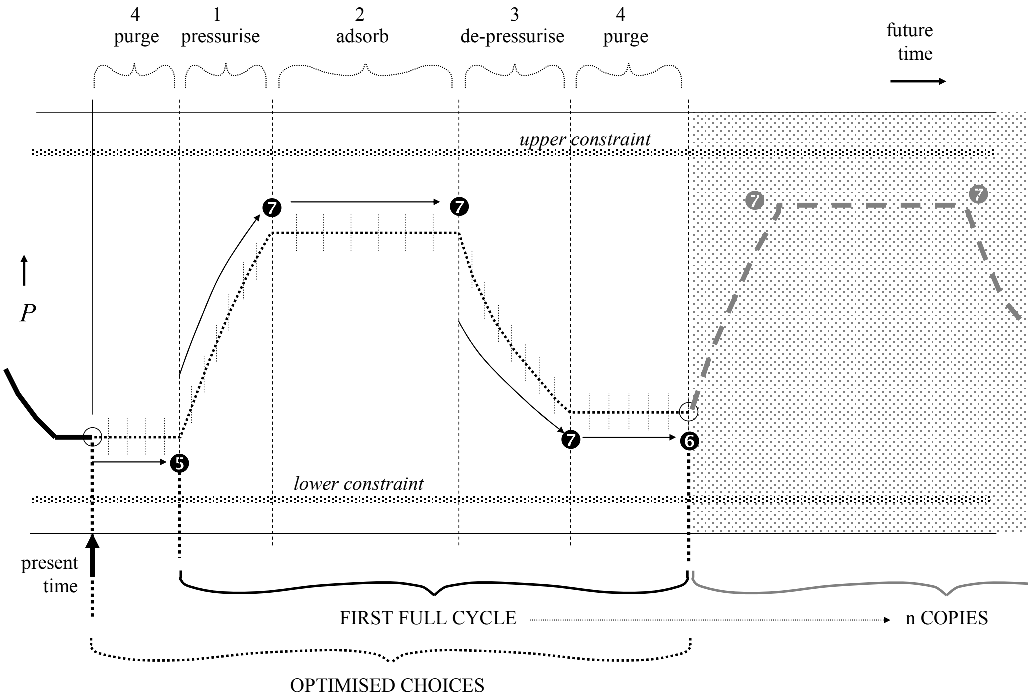


**Fig. 4 Concept of future Skarstrom step length optimisation for a system found (for example) to be in a purge mode on the controller time-step**

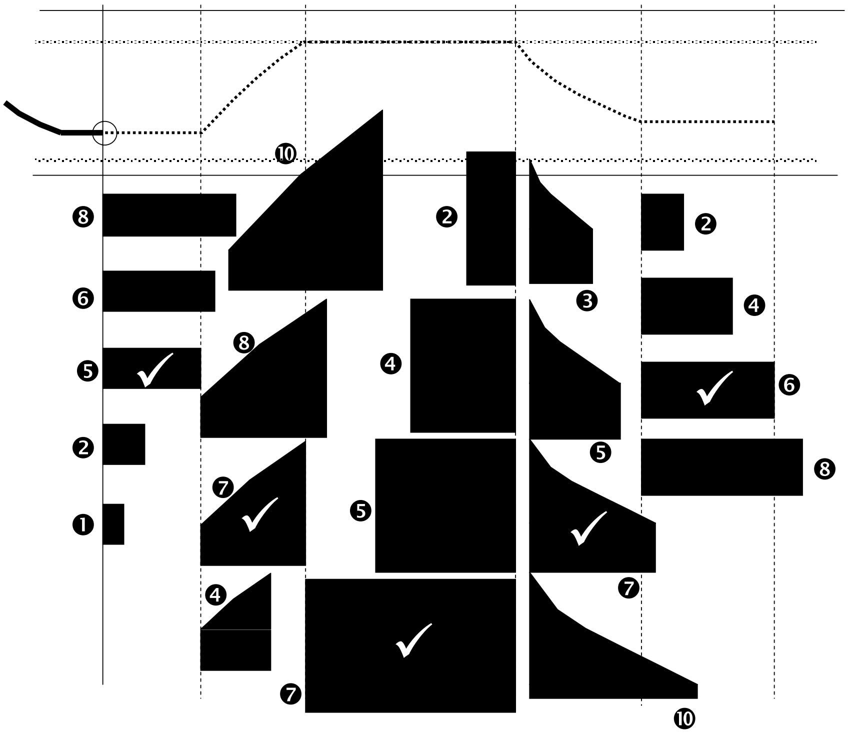


**Fig.5 Choice of transition functions of different numbers of intervals for each Skarstrom step**

### 3.2 Solution method

For each of the Skarstrom steps *j*=1,…4 a range of transition models is pre-prepared, one for each of the possible number of intervals $1 \le i \le n_{\max}$ that could be used for that step:

$$\boldsymbol{x}\left(t+i\Delta t_j\right)=\boldsymbol{A}_j^{(i)}\,\boldsymbol{x}\left(t\right)+\boldsymbol{b}_j^{(i)} \qquad (8)$$

The new arrays $\boldsymbol{A}_j^{(i)}$ and $\boldsymbol{b}_j^{(i)}$ are obtained by individually recursing equations (4) to (7). Now if the particular choices of *i* made to complete the present step *j* and the next 4 complete steps are

$$i[j],\ i[j+1],\ i[j+2],\ i[j+3],\ i[j+4]\ ,$$

where it is understood that the index values will "wrap" around in the range 1,2,3,4, then it is these choices that must be made optimally in determining the future state sequence. Representing $\boldsymbol{x}\left(t+i[j]\Delta t_j\right)$ by $\boldsymbol{x}_j$ one has

$$\boldsymbol{x}_j=\boldsymbol{A}_j^{(i[j])}\,\boldsymbol{x}_{j-1}+\boldsymbol{b}_j^{(i[j])}\qquad\text{(partial step)}\qquad(9)$$

$$\left.\begin{aligned}\boldsymbol{x}_{j+k}&=\boldsymbol{A}_{j+k}^{(i[j+k])}\,\boldsymbol{x}_{j+k-1}+\boldsymbol{b}_{j+k}^{(i[j+k])}\\ \boldsymbol{x}_{j+k+4}&=\boldsymbol{A}_{j+k}^{(i[j+k])}\,\boldsymbol{x}_{j+k+3}+\boldsymbol{b}_{j+k}^{(i[j+k])}\end{aligned}\right\}\ k=1,\dots,4\qquad(10)$$

After completion of the present partial step, two whole cycles are executed, with the second cycle re-using the same number of intervals in each step as in the first cycle.
In this form, the problem lends itself to solution in the *mixed logical dynamical* (MLD) framework of Morari and coworkers in [7], [8] and [9]. Furthermore, the use of linear dynamic models allows solution by *mixed integer linear programming* (MILP). The selection of the optimal number of steps is facilitated by binary variables $\delta$, eg. for equation (9) one requires the constraints

$$\begin{aligned}\boldsymbol{x}_j+\delta_{ij}\boldsymbol{e}_{\max}&\le\boldsymbol{A}_j^{(i)}\,\boldsymbol{x}_{j-1}+\boldsymbol{b}_j^{(i)}+\boldsymbol{e}_{\max}\\ \boldsymbol{x}_j+\delta_{ij}\boldsymbol{e}_{\min}&\ge\boldsymbol{A}_j^{(i)}\,\boldsymbol{x}_{j-1}+\boldsymbol{b}_j^{(i)}+\boldsymbol{e}_{\min}\\ \sum_{i=1}^{n_{\max}}\delta_{ij}&=\quad 1\end{aligned}\qquad(11)$$

Here the vectors **e** contain the maximum and minimum deviation values when (all but one of) the *i*-models are not obeyed (large positive and negative numbers).

Whereas the task required was quite simple - *viz.* choose the best combination of interval numbers in the first five Skarstrom steps - it became clear that the linear program was an inefficient means of solving the problem. The numerous additional constraints required for model choice as in equation (11), and to deal with variable limits, slowed down *LPsolve* (Michel Berkelaar, Eindhoven University of Technology), and caused failures. Even if continuous variables were included in the search, it would be quicker to evaluate *every apex* of the system for its objective value and compliance with constraints. Indeed, this was the procedure used to produce the results below.

### 3.3 Example

A physical description of the system is provided in figure 6. The bed is represented as *N*=9 compartments, and is considered to behave close to plug flow. Four constraints are set on the operation: $P_{\max}$=4.5 bara, $P_{\min}$=1.0 bara, maximum feed air during the adsorption step 2 $M_{\mathrm{FEED}}$= 60000 gmol; maximum product reflux during the purge step 4 $M_{\mathrm{PURGE}}$ = 10000 gmol.

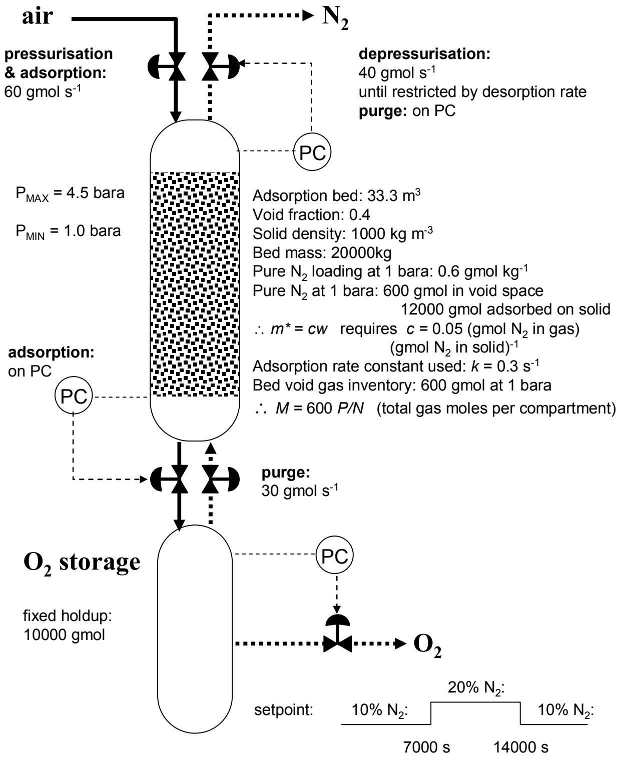

air
N$_2$
pressurisation & adsorption: 60 gmol s$^{-1}$
depressurisation: 40 gmol s$^{-1}$ until restricted by desorption rate
purge: on PC
PC
P$_{MAX}$ = 4.5 bara
P$_{MIN}$ = 1.0 bara
Adsorption bed: 33.3 m$^3$
Void fraction: 0.4
Solid density: 1000 kg m$^{-3}$
Bed mass: 20000kg
Pure N$_2$ loading at 1 bara: 0.6 gmol kg$^{-1}$
Pure N$_2$ at 1 bara: 600 gmol in void space
12000 gmol adsorbed on solid
∴ $m^* = cw$ requires $c$ = 0.05 (gmol N$_2$ in gas) (gmol N$_2$ in solid)$^{-1}$
Adsorption rate constant used: $k$ = 0.3 s$^{-1}$
Bed void gas inventory: 600 gmol at 1 bara
∴ $M = 600\,P/N$ (total gas moles per compartment)
adsorption: on PC
PC
purge: 30 gmol s$^{-1}$
O$_2$ storage
PC
fixed holdup: 10000 gmol
O$_2$
setpoint:
10% N$_2$:
20% N$_2$:
10% N$_2$:
7000 s
14000 s

**Fig. 6 Configuration for predictive control of product concentration in O₂ storage vessel**

The storage vessel is modelled simply using the balances

$$\begin{aligned}\frac{dM_S}{dt}&=F_{d,N}-F_S\\ \frac{dm_S}{dt}&=f_{d,N}-f_S\\ \frac{f_S}{F_S}&=\frac{m_S}{M_S}\end{aligned}\qquad(12)$$

where $F_S$ and $f_S$ are the total and adsorbed species outflows from the storage vessel. In this example, a constant molar inventory is maintained in the vessel ("overflow" mode), so that $F_S = F_{d,N}$.

The maximisation objective weightings for the 9 Skarstrom steps to the prediction horizon have been set as follows:

$$\begin{aligned}&objective=\\ &-100\times\sum_{k=j}^{j+8}\left|\begin{aligned}&deviation\ from\ setpoint\ concentration\\ &\ in\ storage\ vessel\ at\ end\ of\ step\ k\left[\%\mathrm{N}_2\right]\end{aligned}\right|\\ &+1\ \times\sum_{k=j}^{j+8}\left\{\begin{aligned}&\left(product\ to\ storage\ in\ step\ k\left[\mathrm{gmol}\right]\right)-\\ &\quad\left(purge\ from\ storage\ in\ step\ k\left[\mathrm{gmol}\right]\right)\end{aligned}\right\}\end{aligned}\qquad(13)$$

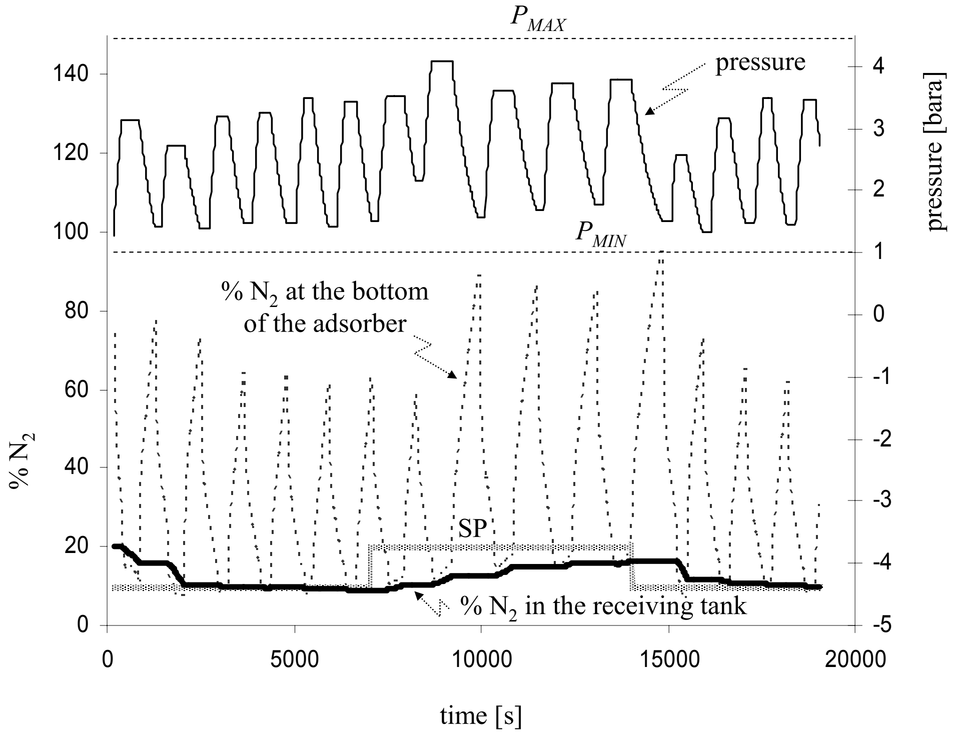


**Fig.7 Predictive control of the $N_2$ concentration in the storage vessel**

The second term represents the net production of the oxygen-rich product up to the end of the present Skarstrom step, plus the following two full Skarstrom cycles. The optimisation considers all constraints up to the end of the first cycle, but the second cycle, using the same step-lengths as the first cycle, only contributes to the objective function. It is reasoned that such constraint transgressions as may be implied in the second cycle should be small. At the start-up the contents of the storage vessel are at 20% $N_2$. The setpoint is initially at 10% $N_2$, stepping up to 20% $N_2$ at time= 7000 s, and back to 10% $N_2$ at time = 14000 s.

Figure 7 shows how the Skarstrom cycles are manipulated in order to achieve the desired concentration in the storage vessel. In this example, the storage vessel has a constant molar inventory and the objective function encourages a high production rate. The objective function is easily altered for the case of "level control" in the storage vessel, where users are drawing the product at arbitrary rates. Viewing figure 7 one recalls that the bottom material in the adsorber only proceeds to the storage vessel during the adsorption phase - ie where downflow occurs through the adsorber at a steady high pressure. Clearly the algorithm is manipulating the bottom concentration during this phase in order to progressively bring the storage vessel contents towards the setpoint.

## Conclusion

In order to make use of fast and robust observation and optimisation algorithms, a lot of effort went into finding an adequate linear representation of the pressure swing adsorption process. This required careful choice of typical flow and composition profiles for each of the four steps of the Skarstrom cycle, to act as operating points about which the linearisation could be conducted.

The distinct steps of the Skarstrom cycle presented an unusual predictive control problem. At any point in time, the optimal control must be based on completion of the present step, followed by a continuation of the Skarstrom sequence up to a defined horizon. The four steps of one cycle have different positive, negative or zero cost implications for the objective function. Possible "end-effect" bias was reduced by continuing with *whole* Skarstrom cycles up to the horizon. Two whole cycles proved adequate, of which only the first was subjected to the constraints.

The optimisation problem was perceived to be hybrid in nature, so the initial approach was to formulate it entirely in the *mixed logical dynamical* (MLD) framework as a *mixed integer dynamic optimisation* (MIDO) problem. To handle the choice of the dynamic equations, however, this required the introduction of many constraints which slowed the solution down and made it unreliable. Far better performance was achieved by solving the combinatorial problem (residual step length, plus next four step lengths) directly by exhaustive interrogation of all combinations for constraint compliance and objective value.

Example applications so far have focused on production maximisation and product composition control in an overflow-type receiving vessel. Good potential was found for predictive control of the composition by manipulation of the Skarstrom cycles. Indeed, this type of automatic control promises significant benefits, as the necessary control moves are not intuitively obvious for manual control. In future work the problem of combined composition and level control will be considered, so that users can draw product from the storage vessel at arbitrary rates.

## References

[1] LATIFI M.A., SALHI D. AND TONDEUR D. : “Optimisation-based simulation of a pressure swing adsorption process, *Adsorption* (In print) (2008)

[2] JIANG L., BIEGLER, L.T. AND FOX V.G. : “Design and optimisation of pressure swing adsorption systems with parallel implementation”, *Computers and Chemical Engineering* **29**, 393-399 (2005)

[3] KVAMSDAL H. M. AND HERTZBERG T. : “Optimisation of pressure swing adsorption systems - the effect of masstransfer during the blowdown step”, *Chemical Engineering Science* **50**, 1203-1212 (2005)

[4] BITZER M. AND ZEITZ M. : “Design of a nonlinear distributed parameter observer for a pressure swing adsorption plant”, *Journal of Process Control* **12**, 533-543 (2002)

[5] BITZER M. AND ZEITZ M. : “Process control scheme for a 2-bed pressure swing adsorption pllant”, *European Symposium on Computer Aided Process Engineering - 12,* J. Grievink and J. van Schijndel (Editors), Elsevier Science (2002)

[6] BITZER M. : “Model-based nonlinear tracking control of pressure swing adsorption plants”, in T. Meurer, K. Graichen and E.D. Gilles eds.: *Control and Observer Design for Nonlinear Finite and Infinite Dimensinal Systems,* Lecture Note in Control and Information Systems, Springer, (2005)

[7] BEMPORAD A. AND MORARI M. : “Control of systems integrating logic, dynamics and constraints”, *Automatica* **35**, 407-427 (1999)

[8] MORARI M., BEMPORAD A. AND MIGNONE, D. : “A framework for control, state estimation, fault detection and verification of hybrid systems”, *Automatisierungstechnik* **48**, 1-8 (2000)

[9] MORARI M. : “Hybrid system analysis and control via mixed integer optimisation”, *Chemical process control VI, AIChE Symposium Series No.326, Vol. 98* (2002)

**Professor Michael Mulholland**

University of KwaZulu-Natal
School of Chemical Engineering
King George V Avenue
Durban 4041
South Africa
Tel.: +27-31-2603123
Fax: +27-31-2601118
E-mail: mulholland@ukzn.ac.za

**Professor M.A. Latifi**

Laboratoire des Sciences du Génie Chimique (LSGC)
ENSIC-INPL
1 rue Grandville
54001 Nancy Cedex
France
Fax: +33 3 83 17 53 26
E-mail: latifi@ensic.inpl-nancy.fr

# Constrained NMPC Using Polynomial Chaos Theory

T. L. Aliyev, E. P. Gatzke

**Abstract**

Establishing an accurate model of a multivariable nonlinear process with uncertain parameters can be difficult. Application of control methods based on nonlinear optimization may result in sub-optimal performance due to changes in the parameters. This paper presents a new control method to handle parametric uncertainty through incorporation of a Polynomial Chaos Theory (PCT) model used in a constrained Nonlinear Model Predictive Control (NMPC) formulation. Uncertain parameters are treated as random variables with a uniform distribution. PCT expresses the entire uncertain process by a complete and orthogonal Legendre polynomial basis in terms of random variables where expanded process outputs are determined by applying Galerkin projection onto the polynomial basis. NMPC formulation has the ability to apply hard input and soft output constraints to maintain the process within specified bounds. It is shown that the proposed formulation can be applied with an adequate tuning to minimize the effect of parametric uncertainty on the process outputs.

## 1. Introduction

Parametric uncertainty affects the quality of a process model and as a result brings in significant challenges for process control engineering, design and analysis. Different methods have been used to better analyze and simulate the uncertain systems: Monte Carlo and other statistical methods, Taylor expansion of the random variables, worst cases scenarios and qualitative analysis of prediction algorithms [12]. While some of these methods are expensive and require parallel simulations to obtain the full statistics after each time step [11], and others are related to artificial intelligence and the field of decision making not currently applicable for large-scale engineering applications, Polynomial Chaos Theory (PCT) is a deterministic method that is capable of calculating the entire statistics of each uncertain variable during only one simulation. PCT analysis that includes polynomial expansion of the uncertain variables results in a multivariable system while the statistical information required for reconstruction of the original variables is stored in the form of coefficients in the basis spanned by the polynomials.

Model Predictive Control (MPC) refers to a class of control algorithms in which a dynamic model of the plant is used to predict and optimize the future behavior of the process [7,16]. At each control interval, the MPC algorithm computes a sequence of the manipulated variables to optimize the future behavior of the plant. MPC has been used extensively for control of refinery operations since MPC can accommodate multivariable systems, actuator constraints, and economic objectives. The original linear MPC method has been extended to include control of nonlinear dynamic systems by a variety of authors [5,23,9,6,13,1,19,4]. Use of more accurate nonlinear process models potentially results in improved controller performance but also requires solution of a more difficult nonlinear optimization problem. Most commercially available MPC technologies are based on a linear process model. For processes that are highly nonlinear, the performance of MPC based on a linear model can be poor. This led to the development of Nonlinear Model Predictive Control (NMPC) methods [2,8,5,4,13,20].

Many of the current NMPC schemes are based on first principles physical models of the process. However, in many cases such models are difficult to obtain, time-consuming and often not available. Process simulators can be used to obtain a nonlinear empirical mathematical model which is identified from input-output data [19]. While NMPC offers potential for improved process operation, the method also faces practical issues that are considerably more challenging than those associated with linear MPC. In particular, the problems associated with the nonlinear optimization routine that must be solved online at each sample period to generate the optimal control sequence. Guaranteed closed-loop stability of nonlinear systems using MPC based methods generally use a terminal state constraint [18,24,15] or some sort of backup control system that monitors convergence [17]. The nonlinearity of a refining process and multivariable interacting nature of such systems makes this class of process attractive to nonlinear MPC methods [25,26].

When implementing control on real multivariable chemical or petrochemical processes such as distillation or separation operations, it is essential to ensure that the process remain within established safety limits and that each product meet certain quality constraints and specifications. For control purposes, all safety constraints and product quality specifications provide a set of control objectives that must be satisfied. However, in situations where the process is characterized by limited degrees of freedom (due to input actuator saturation, nonsquare process with limited inputs) it typically becomes impossible for a controller to meet all control objectives. In these types of cases it is practically impossible for a controller to impose hard constraints on the process outputs. Direct incorporation of hard output constraints would generally lead to infeasibility in the optimization problem.

Since constrained MPC requires the solution of an optimization problem at each time step, the feasibility of that problem should be ensured. Use of a terminal state constraint to guarantee closed-loop stability can cause the nonlinear MPC optimization problem to become infeasible. If the online optimization problem is not feasible, then some constraints would have to be relaxed and the problem would be resolved. Determining the constraints one must relax in order to get a feasible problem with optimal deterioration of the objective function could be extremely difficult. A possible remedy to the problem is to consider prioritized soft constraints on process outputs by including a penalty term in the objective function.

The paper is organized as follows: first, Polynomial Chaos Theory is presented in Section 2. The proposed controller formulation is presented in Section 3 along with an explanation of a methodology for handling output constraints. The case-study used in this paper is presented and discussed in Section 4. This work uses a two-tank model as a nonlinear case-study. Open-loop and closed-loop results are presented in Sections 5 and 6, respectfully, and conclusions are drawn in Section 7.

## 2. Polynomial Chaos Theory

Polynomial Chaos Theory (PCT) was first introduced in 1938 by an American mathematician Norbert Wiener [28,29]. Wiener used Hermite polynomials to expand continuous uncertain variables into a stochastic space and represent the uncertainty in the form of probability distribution function (PDF). The approach used by Wiener was later broadened for the entire Askey scheme of orthogonal polynomials and was renamed Wiener-Askey Polynomial Chaos [30]. Any continuous uncertain variable $X(\omega)$ can be generally described using Polynomial Chaos method as follows:

$$X(\omega)=\sum_{i=0}^{\infty} x_i\phi_i(\xi(\omega)) \qquad (1)$$

where $\xi$ represents random variables in terms of $\omega$ with the type of probability distribution function suitable for the chosen polynomial basis $\phi_i$, and $x_i$ are the coefficients of expansion for this uncertain variable. The infinite dimension of the polynomial space given in Equation 1 must be replaced for computational use by a finite dimension $P$:

$$X(\omega)=\sum_{i=0}^{P} x_i\phi_i(\xi(\omega)) \qquad (2)$$

Note that $P$ equals the number of terms in the expansion starting from 0. In general, the number of terms $P$ needed to describe each uncertain variable in a PCT expanded model can be obtained using

$$P=\left(\frac{(n+k)!}{n!k!}-1\right) \qquad (3)$$

where $n$ is the number of random variables $\xi_i$ and $k$ is the maximum order of the polynomial basis to be used. Two cases are analyzed in this paper and appear in the following subsections: a case of only one uncertain parameter and a case of two uncertain parameters, i.e. $n=1$ and $n=2$, respectively.

### *2.1. Uncertainty in One Parameter*

For a process in which only one parameter is uncertain, Equation 3 becomes for $n=1$:

$$P=\left(\frac{(1+k)!}{k!}-1\right) \qquad (4)$$

which corresponds to $P=k$. This means that the number of terms needed to deterministically represent the stochastic process equals the order of the polynomial expansion. Assuming uniform distribution can choose Legendre polynomials to be used for PCT expansion. The interval of orthogonality for Legendre Polynomials is [-1, 1] and the weighting factor is 1. This translates into two inner product definitions and orthogonality conditions for Legendre polynomials expressed in terms of the $2^{nd}$-order Kronecker delta function $\delta_{mn}$ and a $3^{rd}$-order tensor C $_{ijk}$, respectfully:

$$\langle \phi_i\phi_j\rangle \equiv \langle \phi_i{}^2\rangle \delta_{ij} = \int_{-1}^{1} \phi_i(\xi)\phi_j(\xi)1d\xi \qquad (5)$$

$$\langle \phi_i\phi_j\phi_k\rangle \equiv \langle \phi_k{}^2\rangle C_{ijk} = \int_{-1}^{1} \phi_i(\xi)\phi_j(\xi)\phi_k(\xi)1d\xi \qquad (6)$$

The orthogonality normalization factors $\langle \phi_i^2\rangle$ and $\langle \phi_k^2\rangle$ that appear in Equations 5 and 6 can be found in Table 1 up to order 2. The first two terms (starting from 0) in the case of a second-order Legendre polynomial basis are obtained from Table 1: $\phi_0(\xi)=1$, $\phi_1(\xi)=\xi$, and $\phi_2(\xi)=0.5(3\xi^2-1)$, so that the full PCT expansion of a variable $X$ in the process model is expressed in the case of only one uncertain parameter and the second-order Legendre polynomials by:

$$X=\sum_{i=0}^{2} x_i\phi_i(\xi)=x_0+x_1\xi+0.5x_2(3\xi^2-1) \qquad (7)$$

| Order k | Φk(ξ) | $\int_{-1}^{1} \phi_k^2\, d\xi$ |
|---|---|---|
| 0 | 1 | 2 |
| 1 | ξ | 2/3 |
| 2 | 0.5 ($3ξ^2$ - 1) | 2/5 |

**Table 1. Legendre polynomial terms up to order 2 and orthogonality normalization factors in the case of only one uncertain parameter.**

Once all the variables in the system are expanded according to Equation 2, the resulting expressions are substituted into the governing model equation to form a PCT expanded model equation. The latter may in turn be discretized using Galerkin projection [3,22] onto the polynomial chaos basis in Equation 2 and then expressed in terms of the coefficients $x_i$, the Kronecker delta function $\delta_{ij}$ and the $3^{rd}$-order tensor C $_{ijk}$.

The terms $\langle \phi_i\phi_j\phi_k\rangle$ or, alternatively, $C_{ijk}$ can be calculated up to order 2 (total of $3^3=27$ terms) using Legendre polynomials and normalization factors from Table 1 above. The terms of a tensor $C_{ijk}$ are presented for each combination of *i* , *j* , and *k* in Table 2 [2].

| k→ | 0 | | | 1 | | | 2 | | |
|---|---|---|---|---|---|---|---|---|---|
| **i→** | **0** | **1** | **2** | **0** | **1** | **2** | **0** | **1** | **2** |
| **j=0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **1** |
| **j=1** | **0** | **1/3** | **0** | **1** | **0** | **2/5** | **0** | **2/3** | **0** |
| **j=2** | **0** | **0** | **1/5** | **0** | **2/5** | **0** | **1** | **0** | **2/7** |

**Table 2. Elements of the $3^{rd}$-order tensor $C_{ijk}$**

Using the described procedure and the data presented in the tables above, the Polynomial Chaos Theory analysis results in a new expanded deterministic model of a higher order. In fact, if the original governing model consists of $n$ differential equations, then the expanded model in the case of one uncertain parameter will consist of $n\left(k+1\right)$ equations, where $k$ is the order of the PCT expansion. The resulting PCT expanded model does not include the random variables, and if presented in a state-space model, the states of the new model are the expansion coefficients $x_i$ from Equation 2.

### *2.2. Uncertainty in Two Parameters*

For a process in which two parameters are uncertain, Equation 3 becomes for $n=2$:

$$P=\left(\frac{\left(2+k\right)!}{2!k!}-1\right) \tag{8}$$

Using Equation 8 for a second-order polynomial basis, i.e. $k=2$, it is established that $P=5$, which means that five terms are required in the PCT polynomial expansion in the form of Equation 2. Note that two random variables $\xi_1$ and $\xi_2$ now need to be used to express the terms of Legendre polynomials. The terms up to order 2 appear in Table 3. The detailed PCT analysis of this case does not appear in this paper. However, the analysis is very similar to the one presented in this work.

| First Order, k = 1 | Second Order, k = 2 |
|---|---|
| $\Phi_0 = 1$ | $\Phi_3 = \xi_1\,\xi_2$ |
| $\Phi_1 = \xi_1$ | $\Phi_4 = 0.5\,(3\xi_1^2 - 1)$ |
| $\Phi_2 = \xi_2$ | $\Phi_5 = 0.5\,(3\xi_2^2 - 1)$ |

**Table 3. Legendre polynomial terms up to order 2 and orthogonality normalization factors in the case of two uncertain parameters.**

### 3. NMPC Handling Soft Constraints

For a continuous nonlinear state-space model of the form

$$\frac{dx}{dt}(t)=f\left(t,x(t),u(t)\right) \tag{9}$$
$$y(t)=h\left(t,x(t),u(t)\right)$$

a general nonlinear discrete time dynamic model with *M* past input terms, $n_u$ inputs, $n_y$ outputs, move horizon of *m*, and prediction horizon of *p*, is formulated according to:

$$y_j(k+M)=\sum_{i=1}^{n_u}\sum_{l=k}^{k+M-1}\left[\alpha_{j,i}(l)\,g\left(u_i(l)\right)\right] \tag{10}$$

along with a constant output disturbance term as:

$$y_j(k)\Big|_{k\in[1,p]}=y_j^{Model}+d_j \tag{11}$$

In equation 10, coefficients $\alpha_{j,i}\left(l\right)$ relate output *j* to a general nonlinear input term $g\left(u_i(l)\right)$ at each time *l*. In Equation 11, $y_j^{Model}$ is the model of output j using the discrete representation in the form given in Equation 10, $y_j(k)$ is a predicted value of output *j* at time *k*, and the disturbance update $d_j$ is defined as:

$$d_j=y_j^{Model}(0)-y_j^{Meas}(0) \tag{12}$$

where for each output *j*, $y_j^{Model}(k)$ is the model value at the current time $k=0$ and $y_j^{Meas}(k)$ is the process measurement at the current time $k=0$. In this model, values for $u_i$ before time $k=0$ are known and values for times greater than $m-1$ are fixed to $u(k+m-1)$. This formulation chooses a sequence of input moves over the move horizon (*m*) that minimizes a 2-norm cost function. A 2-norm is used in the MPC objective function in this work to avoid performance issues associated with the 1-norm formulations [21]. The 2-norm objective function with soft constraints takes the form:

$$\phi=\sum_{j=1}^{n_y}\sum_{k=1}^{p}\Gamma_{y,j}\left(e_j(k)\right)^2+\sum_{i=1}^{n_u}\sum_{l=1}^{m}\Gamma_{u,i}\left(\Delta u_i(l)\right)^2$$
$$+\sum_{j=1}^{n_y}\sum_{k=1}^{p}\Gamma_{y_{soft},j}\left(s_j(k)\right)^2 \tag{13}$$

where $e_j(k)$ and $s_j(k)$ are the values of error predicted for the $k^{th}$ time step into the future for each output *j*. The error (*e*) is defined as

$$e_j(k)\,|_{k\in[1,p]}=y_{p,j}(k)-y_{sp,j}(k) \tag{14}$$

where $y_{sp,j}(k)$ is the known setpoint value of output *j* at time *k* and $y_{p,j}(k)$ is the predicted value of output *j* at time *k*, updated based upon process model mismatch at the current time. The term $\Delta u_i$ defines changes in input *i* according to

$$\Delta u_i(k)\,|_{k\in[1,m]}=u_i(M+k)$$
$$-u_i(M+k-1) \tag{15}$$

The soft constraint violation (*s*) is defined for those values of output *j* that are outside the range $\left[y_{soft}^{l}(k),\cdots y_{soft}^{u}(k)\right]$. For model predictions above the upper soft constraint limit, the soft constraint violation is defined as:

$$s_j(k)\,|_{k\in[1,p]}=y_{p,j}(k)-y_{soft,j}^{u}(k) \tag{16}$$

For violation below the lower soft constraint limit, this violation is defined as:

$$s_j(k)\,|_{k\in[1,p]}=y_{soft,j}^{l}(k)-y_{p,j}(k) \tag{17}$$

The soft constraint violation is zero otherwise. This allows for violation of output constraints without making the controller optimization problem infeasible. $\Gamma_{y,j}$, $\Gamma_{u,i}$ and $\Gamma_{y_{soft},j}$ are weighting factors used to define the relative importance of each objective function term in Equation 13. However, penalty values must be tuned for this process using the weight in the $\Gamma_{soft}$ matrix. The term $\Gamma_{y_{soft},j}$ is an penalty on the output error that is applied depending on soft constraints on that output. Note that the value of $\Gamma_{y_{soft},j}$ is at least an order of magnitude larger than the maximum value of $\Gamma_{y,j}$. This is done to ensure that soft constraint violations are minimized as much as possible.

In some cases, there may be multiple output constraints. For example, a process may make multiple product types that are defined by the measured product quality. High profit products must meet stringent quality limits while lower quality products may be sold at a lower cost. To implement this approach, based on deviation from the setpoint error in Equation 14, multiple layers of soft constraints with different penalties can be implemented. In this formulation, the value of the soft penalty $\Gamma_{y_{soft},j}$ in the objective function, Equation 13, increases by order of magnitude with each additional layer, starting from a tight range (for errors less than or equal to some small value $\varepsilon$) and ending with a broader loose range (for errors greater than 1.5$\varepsilon$, for example). The value of the quality range $\varepsilon$ as well as a penalty used for each layer of constraints can be adjusted to process needs depending on the output and the sign of the error in Equation 14, i.e. the constraint layers boundaries are not necessarily symmetrical around the setpoint.

Combining Equations 11, 12, and 14-17 results in a single objective function (Equation 13) that depends only on the input values. The resulting optimization problem becomes:

$$\min \phi \qquad u^l \le u \le u^u \tag{18}$$

The inclusion of soft constraints that are only active in portions of the parameter space make the objective function nonsmooth. The convexity of the objective function $\phi$ could be examined in detail.

Optimization is implemented using *fminsearch* - multidimensional nonlinear minimization - in Matlab. In MPC formulations, the prediction horizon (*p*) can be chosen as a large value to promote stability. Stability can also be ensured through the use of a hard constraint which drives the terminal state error to zero. This theoretical guarantee for nominal stability fails in cases where an unreachable setpoint is provided, as the optimization problem is infeasible [10]. In such cases, a soft constraint could be used to drive the system to a stable operating point when possible.

## 4. Case study: Two tanks in series

The case study analyzed in this paper is a simple two tank model described in Figure 1. The constant cross-sectional tank areas are $A_1$ and $A_2$, and the liquid heights are $h_1$ and $h_2$, respectfully. There are two valves at the outlet of each tank whose coefficients are $k_2$ and $k_3$. The flowrate into the first tank is $F_1(t)$, the flowrates from the tanks are proportional to the valve coefficients and the square root terms of the liquid levels through:

$$F_2(t) = k_2\sqrt{h_1(t)} \qquad F_3(t) = k_3\sqrt{h_2(t)} \tag{19}$$

The material balance around the system results in the following mathematical model:

$$\frac{dh_1(t)}{dt} = \frac{1}{A_1}\left(F_1(t) - k_2\sqrt{h_1(t)}\right) \qquad \frac{dh_2(t)}{dt} = \frac{1}{A_2}\left(k_2\sqrt{h_1(t)} - k_3\sqrt{h_2(t)}\right) \tag{20}$$

It is worth noting at this point that in most complex chemical processes it is impossible for a mathematical model to fully represent all the aspects of the ongoing process operation. However, the nonlinear model given in Equation 20 is considered the most suitable representation of the two-tank model. Note that time dependency of all the variables was omitted in the model equation.

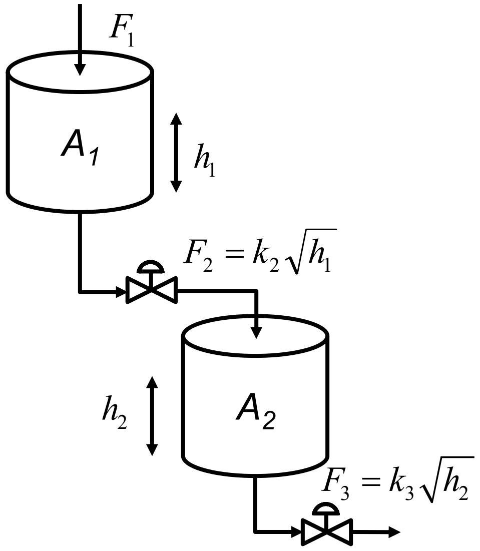


**Figure 1. Flowchart of two connected tanks with liquid levels h $_1$ and h $_2$, cross-sectional areas A $_1$ and A $_2$.**

In order to effectively analyze the nonlinearity of the system, the square root terms $\sqrt{h_i}$ can be approximated using the Taylor series expansion in the neighborhood of the points $h_i^0$:

$$f(h_i) = f(h_i^0) + f'(h_i^0)(h_i - h_i^0) + \frac{f''(h_i^0)}{2}(h_i - h_i^0)^2 + \ldots \frac{f^{(n)}(h_i^0)}{n!}(h_i - h_i^0)^n \tag{21}$$

where $f(h_i) = \sqrt{h_i}$.

### *4.1. Uncertainty in One Parameter $k_2$*

### 4.1.1. First-Order Taylor Approximation

Using the first order Taylor series expansion given in Equation 21, the nonlinear square term $\sqrt{h_i(t)}$ for any $i$ reduces to:

$$\sqrt{h_i(t)} = \sqrt{h_i^0} + \frac{1}{2\sqrt{h_i^0}}\left(h_i(t) - h_i^0\right) \tag{22}$$

If $h_i^0$ is known, one can substitute Equation 22 into the model in Equation 20 so that the modified model becomes:

$$\frac{dh_1(t)}{dt} = \frac{1}{A_1}\left[F_1(t) - \frac{1}{2}k_2\sqrt{h_1^0} - \frac{k_2}{2\sqrt{h_1^0}}h_1(t)\right]$$

$$\frac{dh_2(t)}{dt} = \frac{1}{A_2}\left[\frac{1}{2}k_2\sqrt{h_1^0} + \frac{k_2}{2\sqrt{h_1^0}}h_1(t) - \frac{1}{2}k_3\sqrt{h_2^0} - \frac{k_3}{2\sqrt{h_2^0}}h_2(t)\right] \tag{23}$$

Since the first valve coefficient $k_2$ is the only uncertain variable, one can apply polynomial chaos expansion to the state variables $h_1$ and $h_2$, and the only uncertain parameter $k_2$ to get:

$$
\begin{aligned}
k_2(\xi) &= \sum_{i=0}^{P} k_{2,i}\phi_i(\xi) \\
h_1(t,\xi) &= \sum_{i=0}^{P} h_{1,i}(t)\phi_i(\xi) \\
h_2(t,\xi) &= \sum_{i=0}^{P} h_{2,i}(t)\phi_i(\xi)
\end{aligned}
\tag{24}
$$

In this PCT expansion, $\phi_i(\xi)$ (where $\xi$ is a random variable with uniform distribution) can be chosen as Legendre polynomials for which the interval of orthogonality is [-1, 1] and the weighting function is simply 1, so that the first three Legendre polynomial terms (for P = 2) in the case of only one uncertain parameter as given in Table 1 are: $\phi_0(\xi)=1$, $\phi_1(\xi)=\xi$, and $\phi_2(\xi)=\frac{1}{2}(3\xi^2-1)$. Now one can insert Equation 24 into Equation 23 to obtain:

$$
\begin{aligned}
\sum_{i=0}^{P}\frac{dh_{1,i}(t,\xi)}{dt}\phi_i(\xi) &= \frac{1}{A_1}\left[F_1(t)-\frac{1}{2}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\,\phi_i(\xi)\right] \\
&-\frac{1}{A_1}\left[\frac{1}{2\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)\phi_i(\xi)\phi_j(\xi)\right] \\
\sum_{i=0}^{P}\frac{dh_{2,i}(t,\xi)}{dt}\phi_i(\xi) &= \frac{1}{A_2}\left[\frac{1}{2}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\,\phi_i(\xi)\right] \\
&+\frac{1}{A_2}\left[\frac{1}{2\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)\phi_i(\xi)\phi_j(\xi)\right] \\
&-\frac{1}{A_2}\left[\frac{1}{2\sqrt{h_2^0}}k_3\sum_{j=0}^{P}h_{2,j}(t)\phi_j(\xi)-\frac{1}{2}k_3\sqrt{h_2^0}\right]
\end{aligned}
\tag{25}
$$

Applying the orthogonality condition given in Equation 5 and using a tensor notation from Equation 6 it is possible to discretize the PCT expanded model in Equation 25 using Galerkin projection onto the polynomial chaos basis in Equation 24 to get:

$$
\begin{aligned}
&(\forall k\in\{0,\dots,P\})\quad \frac{dh_{1,k}(t)}{dt} = \\
&\frac{\int_{-1}^{1}\left\{\frac{1}{A_1}\left[F_1(t)-\frac{1}{2}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\,\phi_i(\xi)\right]\right\}\phi_k\,d\xi}{\int_{-1}^{1}\phi_k^2\,d\xi} \\
&+\frac{\int_{-1}^{1}\left\{\frac{1}{A_1}\left[-\frac{1}{2\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)\phi_i(\xi)\phi_j(\xi)\right]\right\}\phi_k\,d\xi}{\int_{-1}^{1}\phi_k^2\,d\xi}
\end{aligned}
$$

$$
\begin{aligned}
&(\forall k\in\{0,\dots,P\}) \\
&\frac{dh_{2,k}(t)}{dt} = \\
&\frac{\int_{-1}^{1}\left\{\frac{1}{A_2}\left[\frac{1}{2}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\,\phi_i(\xi)-\frac{1}{2\sqrt{h_2^0}}k_3\sum_{j=0}^{P}h_{2,j}(t)\phi_j(\xi)\right]\right\}\phi_k\,d\xi}{\int_{-1}^{1}\phi_k^2\,d\xi} \\
&+\frac{\int_{-1}^{1}\left\{\frac{1}{A_2}\left[\frac{1}{2\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)\phi_i(\xi)\phi_j(\xi)-\frac{1}{2}k_3\sqrt{h_2^0}\right]\right\}\phi_k\,d\xi}{\int_{-1}^{1}\phi_k^2\,d\xi}
\end{aligned}
\tag{26}
$$

This equation represents the PCT expanded two tanks model, where instead of two original model equations the expanded model now consists of $2(k+1)$ equations. It can be modified and rewritten in terms of a 3$^{rd}$-order tensor $C_{ijk}$ and the Kronecker delta $\delta_{mn}$ using Equations 6 and 5, respectively:

$$
\begin{aligned}
\frac{dh_{1,k}(t)}{dt} &= \frac{1}{A_1}\left[-\frac{1}{2\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)C_{ijk}\right] \\
&+\frac{1}{A_1}\left[\frac{\int_{-1}^{1}F_1(t)\phi_k d\xi}{\int_{-1}^{1}\phi_k^2 d\xi}-\frac{1}{2}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\delta_{ik}\right] \\
\frac{dh_{2,k}(t)}{dt} &= \frac{1}{A_2}\left[\frac{1}{2\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)C_{ijk}\right] \\
&+\frac{1}{A_2}\left[\frac{1}{2}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\delta_{ik}-\frac{1}{2\sqrt{h_2^0}}k_3\sum_{j=0}^{P}h_{2,j}(t)\delta_{jk}\right] \\
&+\frac{1}{A_2}\left[-\frac{\int_{-1}^{1}\frac{1}{2}k_3\sqrt{h_2^0}\phi_k\,d\xi}{\int_{-1}^{1}\phi_k^2 d\xi}\right]
\end{aligned}
\tag{27}
$$

A denominator $\int_{-1}^{1}\phi_k^2\,d\xi$ in Equations 26 and 27 that accounts for orthogonality of Galerkin projection is given for $k$ up to order 2 in Table 1. The terms $\langle\phi_i\phi_j\phi_k\rangle$ or, alternatively, $C_{ijk}$ can be calculated using the inner product definition in Equation 6. These terms are summarized in Table 2. In the first part of the resulting PCT expanded model (Equation 27) a term $\int_{-1}^{1}F_1(t)\phi_k d\xi$ contributes only when $k=0$ due to the properties of Legendre polynomials. For this simplest case of zero-order, the resulting model consisting of two differential equations is:

$$\frac{dh_{1,0}(t)}{dt} = \frac{1}{A_1}\left[F_1(t) - \frac{1}{2}k_{2,0}\sqrt{h_1^0}\right] - \frac{1}{A_1}\left[\frac{1}{2\sqrt{h_1^0}}k_{2,0}\,h_{1,0}(t)\right] \tag{28}$$

$$\frac{dh_{2,0}(t)}{dt} = \frac{1}{A_2}\left[\frac{1}{2}k_{2,0}\sqrt{h_1^0} - \frac{1}{2}k_3\sqrt{h_2^0}\right] + \frac{1}{A_2}\left[\frac{1}{2\sqrt{h_1^0}}k_{2,0}\,h_{1,0}(t) - \frac{1}{2\sqrt{h_2^0}}k_3 h_{2,0}(t)\right]$$

For this zero-order model, $C_{000} = 1$ was used. The resulting first-order PCT expanded model consists of four differential equations with up to $P = 1$ terms in each. Using the values of $C_{ijk}$ for $k = 0$ and $k = 1$, one can obtain from Equation 27:

$$\frac{dh_{1,0}(t)}{dt} = \frac{1}{2A_1}\left[2F_1(t) - \frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,0}(t) - \frac{1}{3\sqrt{h_1^0}}k_{2,1}\,h_{1,1}(t) - \sqrt{h_1^0}\,k_{2,0}\right]$$

$$\frac{dh_{1,1}(t)}{dt} = \frac{1}{2A_1}\left[-\frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,1}(t) - \frac{1}{\sqrt{h_1^0}}k_{2,1}\,h_{1,0}(t) - \sqrt{h_1^0}\,k_{2,1}\right]$$

$$\frac{dh_{2,0}(t)}{dt} = \frac{1}{2A_2}\left[\frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,0}(t) + \frac{1}{3\sqrt{h_1^0}}k_{2,1}\,h_{1,1}(t)\right] + \frac{1}{2A_2}\left[-\frac{1}{\sqrt{h_2^0}}k_3\,h_{2,0}(t) - k_3\sqrt{h_2^0} + \sqrt{h_1^0}\,k_{2,0}\right] \tag{29}$$

$$\frac{dh_{2,1}(t)}{dt} = \frac{1}{2A_2}\left[\frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,1}(t) + \sqrt{h_1^0}\,k_{2,1}\right] + \frac{1}{2A_2}\left[\frac{1}{\sqrt{h_1^0}}k_{2,1}\,h_{1,0}(t) - \frac{1}{\sqrt{h_2^0}}k_3\,h_{2,1}(t)\right]$$

Equation 29 can also be rewritten in a state-space form:

$$\frac{d\underline{x}_{PCT}(t)}{dt} = A_{PCT}(t)\underline{x}_{PCT}(t) + B_{PCT}(t)\underline{u}(t) + \Gamma_{PCT} \tag{30}$$

$$\underline{y}_{PCT}(t) = C_{PCT}(t)\underline{x}_{PCT}(t) + D_{PCT}(t)\underline{u}(t)$$

where $\underline{x}_{PCT}(t) = \left[h_{1,0}(t)h_{1,1}(t)h_{2,0}(t)h_{2,1}(t)\right]^T$ is a vector of expanded states,

and $\underline{y}_{PCT}(t) = \underline{x}_{PCT}(t)$ is the output vector that includes all the expanded states.

Matrices $A_{PCT}$, $B_{PCT}$, $C_{PCT}$, $D_{PCT}$ and $\Gamma_{PCT}$ can then be identified as follows:

$$A_{PCT} = \begin{pmatrix} -\frac{1}{2}\frac{k_{2,0}}{A_1\sqrt{h_1^0}} & -\frac{1}{6}\frac{k_{2,1}}{A_1\sqrt{h_1^0}} & 0 & 0 \\ -\frac{1}{2}\frac{k_{2,1}}{A_1\sqrt{h_1^0}} & -\frac{1}{2}\frac{k_{2,0}}{A_1\sqrt{h_1^0}} & 0 & 0 \\ \frac{1}{2}\frac{k_{2,0}}{A_2\sqrt{h_1^0}} & \frac{1}{6}\frac{k_{2,1}}{A_2\sqrt{h_1^0}} & -\frac{1}{2}\frac{k_3}{A_2\sqrt{h_2^0}} & 0 \\ \frac{1}{2}\frac{k_{2,1}}{A_2\sqrt{h_1^0}} & \frac{1}{2}\frac{k_{2,0}}{A_2\sqrt{h_1^0}} & 0 & -\frac{1}{2}\frac{k_3}{A_2\sqrt{h_2^0}} \end{pmatrix} \tag{31}$$

$$B_{PCT} = \begin{pmatrix} \frac{1}{A_1} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{32}$$

$$\Gamma_{PCT} = \begin{pmatrix} -\frac{1}{2A_1}\sqrt{h_1^0}\,k_{2,0} \\ -\frac{1}{2A_1}\sqrt{h_1^0}\,k_{2,1} \\ \frac{1}{2A_2}\sqrt{h_1^0}\,k_{2,0} - k_3\frac{1}{2A_2}\sqrt{h_2^0} \\ \frac{1}{2A_2}\sqrt{h_1^0}\,k_{2,1} \end{pmatrix} \tag{33}$$

$$C_{PCT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad D_{PCT} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{34}$$

The second-order PCT expanded model consists of six differential equations with $P = 2$ terms in each.

$$\frac{dh_{1,0}(t)}{dt}=\frac{1}{2A_1}\left[2F_1(t)-\frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,0}(t)\right]$$
$$+\frac{1}{2A_1}\left[-\frac{1}{3\sqrt{h_1^0}}k_{2,1}\,h_{1,1}(t)-\sqrt{h_1^0}\,k_{2,0}\right]$$
$$\frac{dh_{1,1}(t)}{dt}=\frac{1}{2A_1}\left[-\frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,1}(t)-\sqrt{h_1^0}\,k_{2,1}\right]$$
$$+\frac{1}{2A_1}\left[-\frac{1}{\sqrt{h_1^0}}k_{2,1}\,h_{1,0}(t)-\frac{2}{5\sqrt{h_1^0}}k_{2,1}\,h_{1,2}\right]$$
$$\frac{dh_{1,2}(t)}{dt}=\frac{1}{2A_1}\left[-\frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,2}(t)-\frac{2}{3\sqrt{h_1^0}}k_{2,1}\,h_{1,1}(t)\right]$$
$$\frac{dh_{2,0}(t)}{dt}=\frac{1}{2A_2}\left[\frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,0}(t)+k_{2,0}\sqrt{h_1^0}\right]$$
$$+\frac{1}{2A_2}\left[-\frac{1}{\sqrt{h_2^0}}k_3h_{2,0}(t)+\frac{1}{3\sqrt{h_1^0}}k_{2,1}\,h_{1,1}(t)-k_3\sqrt{h_2^0}\right]$$
$$\frac{dh_{2,1}(t)}{dt}=\frac{1}{2A_2}\left[\frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,1}(t)+\sqrt{h_1^0}\,k_{2,1}\,h_{1,0}(t)\right]$$
$$+\frac{1}{2A_2}\left[\frac{2}{5\sqrt{h_1^0}}k_{2,1}\,h_{1,2}-\frac{1}{\sqrt{h_2^0}}k_3h_{2,1}(t)+k_{2,1}\sqrt{h_1^0}\right] \tag{35}$$
$$\frac{dh_{2,2}(t)}{dt}=\frac{1}{2A_2}\left[\frac{1}{\sqrt{h_1^0}}k_{2,0}\,h_{1,2}(t)\right]$$
$$+\frac{1}{2A_2}\left[\frac{2}{3\sqrt{h_1^0}}k_{2,1}\,h_{1,1}-\frac{1}{\sqrt{h_2^0}}k_3h_{2,2}(t)\right]$$

### 4.1.2. Second-Order Taylor Approximation

Using the second order Taylor series expansion given in Equation 21, the nonlinear square term $\sqrt{h_i}$ for any $i$ reduces to:

$$\sqrt{h_i}=\sqrt{h_i^0}+\frac{1}{2\sqrt{h_i^0}}(h_i-h_i^0)-\frac{1}{8\left(\sqrt{h_i^0}\right)^3}\left(h_i-h_i^0\right)^2 \tag{36}$$

If $h_i^0$ is known, can substitute Equation 36 into the model in Equation 20 so that the modified model consisting of two differential equations becomes:

$$\frac{dh_1(t)}{dt}=\frac{k_2}{A_1}\left[\frac{F_1(t)}{k_2}-\frac{1}{2}\sqrt{h_1^0}+\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\left(h_1^0\right)^2\right]$$
$$+\frac{k_2}{A_1}\left[-\left(\frac{1}{2\sqrt{h_1^0}}+\frac{h_1^0}{4\left(\sqrt{h_1^0}\right)^3}\right)h_1(t)\right]$$
$$+\frac{k_2}{A_1}\left[\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\left(h_1(t)\right)^2\right]$$

$$\frac{dh_2(t)}{dt}=\frac{k_2}{A_2}\left[\frac{1}{2}\sqrt{h_1^0}-\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\left(h_1^0\right)^2\right]$$
$$+\frac{k_2}{A_2}\left[\left(\frac{1}{2\sqrt{h_1^0}}+\frac{h_1^0}{4\left(\sqrt{h_1^0}\right)^3}\right)h_1(t)\right]$$
$$+\frac{k_2}{A_2}\left[-\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\left(h_1(t)\right)^2\right]$$
$$+\frac{k_3}{A_2}\left[-\frac{1}{2}\sqrt{h_2^0}+\frac{1}{8\left(\sqrt{h_2^0}\right)^3}\left(h_2^0\right)^2\right]$$
$$+\frac{k_3}{A_2}\left[-\left(\frac{1}{2\sqrt{h_2^0}}+\frac{h_2^0}{4\left(\sqrt{h_2^0}\right)^3}\right)h_2(t)\right]$$
$$+\frac{k_3}{A_2}\left[\frac{1}{8\left(\sqrt{h_2^0}\right)^3}\left(h_2(t)\right)^2\right] \tag{37}$$

This model can be simplified into

$$\frac{dh_1(t)}{dt}=\frac{k_2}{A_1}\left[\frac{F_1(t)}{k_2}-\frac{3}{8}\sqrt{h_1^0}-\frac{3}{4\sqrt{h_1^0}}h_1(t)\right]$$
$$+\frac{k_2}{A_1}\left[\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\left(h_1(t)\right)^2\right]$$

$$\frac{dh_2(t)}{dt}=\frac{k_2}{A_2}\left[\frac{3}{8}\sqrt{h_1^0}+\frac{3}{4\sqrt{h_1^0}}h_1(t)\right]$$
$$\frac{k_2}{A_2}\left[-\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\left(h_1(t)\right)^2\right]$$
$$+\frac{k_3}{A_2}\left[-\frac{3}{8}\sqrt{h_2^0}-\frac{3}{4\sqrt{h_2^0}}h_2(t)\right]$$
$$+\frac{k_3}{A_2}\left[\frac{1}{8\left(\sqrt{h_2^0}\right)^3}\left(h_2(t)\right)^2\right] \tag{38}$$

Here, again set the valve coefficient $k_2$ as the only uncertain variable. At this point one can apply polynomial chaos expansion to the state variables $h_1$ and $h_2$, and the uncertain parameters $k_2$ to obtain the same PCT expansion as in Equation 24. However, the model in Equation 38 that was obtained using Taylor approximation up to second order, now includes nonlinear multiplication of three variables: $k_2h_1h_1$ and $k_3h_2h_2$, respectively. Now, by inserting Equation 24 into the model in Equation 37, or in other words presenting the model using Legendre polynomials $\phi_i(\xi)$, yields:

$$A_1\sum_{i=0}^{P}\frac{dh_{1,i}(t,\xi)}{dt}\phi_i(\xi)=F_1(t)-\frac{3}{8}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\,\phi_i(\xi)$$
$$-\frac{3}{4\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)\phi_i(\xi)\phi_j(\xi)$$
$$+\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\sum_{i=0}^{P}\sum_{j=0}^{P}\sum_{l=0}^{P}k_{2,i}\,h_{1,j}(t)h_{1,l}(t)\phi_i(\xi)\phi_j(\xi)\phi_l(\xi)$$

$$A_2\sum_{i=0}^{P}\frac{dh_{2,i}(t,\xi)}{dt}\phi_i(\xi)=\frac{3}{8}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\,\phi_i(\xi)$$
$$+\frac{3}{4\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)\phi_i(\xi)\phi_j(\xi)$$
$$-\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\sum_{i=0}^{P}\sum_{j=0}^{P}\sum_{l=0}^{P}k_{2,i}\,h_{1,j}(t)h_{1,l}(t)\phi_i(\xi)\phi_j(\xi)\phi_l(\xi)$$
$$-\frac{3}{8}k_3\sqrt{h_2^0}-\frac{3}{4\sqrt{h_2^0}}k_3\sum_{j=0}^{P}h_{2,j}(t)\phi_j(\xi)$$
$$\frac{1}{8\left(\sqrt{h_2^0}\right)^3}k_3\sum_{i=0}^{P}\sum_{j=0}^{P}h_{2,i}(t)h_{2,j}(t)\phi_i(\xi)\phi_j(\xi) \tag{39}$$

Now a $4^{th}$-order tensor notation should be introduced in addition to that in Equation 6:

$$\left\langle\phi_i\phi_j\phi_l\phi_k\right\rangle\equiv\left\langle\phi_k^2\right\rangle D_{ijlk} \tag{40}$$

where an inner product is defined for Legendre polynomials with weighting factor 1 according to:

$$\left\langle\phi_i\phi_k\right\rangle=\int_{-1}^{1}\phi_i(\xi)\phi_k(\xi)1d\xi$$
$$\left\langle\phi_i\phi_j\phi_k\right\rangle=\int_{-1}^{1}\phi_i(\xi)\phi_j(\xi)\phi_k(\xi)1d\xi \tag{41}$$
$$\left\langle\phi_i\phi_j\phi_l\phi_k\right\rangle=\int_{-1}^{1}\phi_i(\xi)\phi_j(\xi)\phi_l(\xi)\phi_k(\xi)1d\xi$$

In Equations 6 and 40, $C_{ijk}$ and $D_{ijlk}$ are $3^{rd}$ and $4^{th}$-order tensors, respectively, that can be determined based on the knowledge of Legendre polynomial terms $\phi_i$.

At this point it is possible to discretize the PCT expanded model in Equation 39 using Galerkin projection onto the polynomial chaos basis in Equation 24 to get:

$$\int_{-1}^{1}\phi_k^2\,d\xi\frac{dh_{1,k}(t)}{dt}= \tag{42}$$
$$+\int_{-1}^{1}\frac{1}{A_1}\left[H_1+H_2+H_3\right]$$
$$H_1=F_1(t)-\frac{3}{8}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\,\phi_i(\xi)$$
$$H_2=-\frac{3}{4\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)\phi_i(\xi)\phi_j(\xi)$$
$$H_3=\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\sum_{i=0}^{P}\sum_{j=0}^{P}\sum_{l=0}^{P}k_{2,i}\,h_{1,j}(t)h_{1,l}(t)\phi_i(\xi)\phi_j(\xi)\phi_l(\xi)$$

$$\int_{-1}^{1}\phi_k^2\,d\xi\frac{dh_{2,k}(t)}{dt}= \tag{43}$$
$$\int_{-1}^{1}\frac{1}{A_2}\left[H_4+H_5+H_6+H_7+H_8\right]\phi_k\,d\xi$$
$$H_4=\frac{3}{8}\sqrt{h_1^0}\sum_{i=0}^{P}k_{2,i}\,\phi_i(\xi)$$
$$H_5=\frac{3}{4\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P}k_{2,i}\,h_{1,j}(t)\phi_i(\xi)\phi_j(\xi)$$
$$H_6=-\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\sum_{i=0}^{P}\sum_{j=0}^{P}\sum_{l=0}^{P}k_{2,i}\,h_{1,j}(t)h_{1,l}(t)\phi_i(\xi)\phi_j(\xi)\phi_l(\xi)$$
$$H_7=-\frac{3}{8}k_3\sqrt{h_2^0}-\frac{3}{4\sqrt{h_2^0}}k_3\sum_{j=0}^{P}h_{2,j}(t)\phi_j(\xi)$$

$$H_8 = \frac{1}{8\left(\sqrt{h_2^0}\right)^3} k_3 \sum_{i=0}^{P}\sum_{j=0}^{P} h_{2,i}\left(t\right) h_{2,j}(t)\phi_i(\xi)\phi_j(\xi)$$

Equations 42 and 43 represent the PCT expanded nonlinear two tanks model, where instead of two original model equations the expanded model now consists of $2(k+1)$ equations. These equations can be modified and rewritten in terms of a $3^{rd}$-order tensor $C_{ijk}$ , a $4^{th}$-order tensor $D_{ijlk}$, and the Kronecker delta $\delta_{ik}$ using Equations 6, 40 and 5, respectively:

$$A_1 \frac{dh_{1,k}(t)}{dt} = \frac{\int_{-1}^{1} F_1\left(t\right)\phi_k d\xi}{\int_{-1}^{1} \phi_k^2 d\xi}$$

$$-\frac{3}{8}\sqrt{h_1^0}\sum_{i=0}^{P} k_{2,i}\delta_{ik} - \frac{3}{4\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P} k_{2,i}\, h_{1,j}\left(t\right)C_{ijk}$$

$$+\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\sum_{i=0}^{P}\sum_{j=0}^{P}\sum_{l=0}^{P} k_{2,i}\, h_{1,j}\left(t\right)h_{1,l}\left(t\right)D_{ijlk}$$

$$A_2 \frac{dh_{2,k}(t)}{dt} = \frac{3}{8}\sqrt{h_1^0}\sum_{i=0}^{P} k_{2,i}\delta_{ik}$$

$$+\frac{3}{4\sqrt{h_1^0}}\sum_{i=0}^{P}\sum_{j=0}^{P} k_{2,i}\, h_{1,j}\left(t\right)C_{ijk}$$

$$-\frac{1}{8\left(\sqrt{h_1^0}\right)^3}\sum_{i=0}^{P}\sum_{j=0}^{P}\sum_{l=0}^{P} k_{2,i}\, h_{1,j}\left(t\right)h_{1,l}\left(t\right)D_{ijlk}$$

$$-\frac{\int_{-1}^{1} \tfrac{3}{8}k_3\sqrt{h_2^0}\phi_k\, d\xi}{\int_{-1}^{1} \phi_k^2 d\xi} - \frac{3}{4\sqrt{h_2^0}}k_3\sum_{j=0}^{P} h_{2,j}\left(t\right)\delta_{jk} \qquad (44)$$

$$+\frac{1}{8\left(\sqrt{h_2^0}\right)^3}k_3\sum_{i=0}^{P}\sum_{j=0}^{P} h_{2,i}\left(t\right)h_{2,j}\left(t\right)C_{ijk}$$

A denominator $\int_{-1}^{1}\phi_k^2\, d\xi$ in Equations 42 and 44 that accounts for orthogonality of Galerkin projection was given for $k$ up to order 2 in Table 1. The terms $\left\langle \phi_i\phi_j\phi_k \right\rangle$ or, alternatively, $C_{ijk}$ were also calculated up to order 2 (total of $3^3 = 27$ terms) using Legendre polynomials and normalization factors from Table 1. These terms appear in Table 2.
The main difference between the PCT expanded model after applying a Galerkin projection that was developed in the previous section and given in Equation 27 and the one obtained for the second order Taylor approximation - Equation 44 - is the addition of the $4^{th}$-order tensor $D_{ijlk}$. Its coefficients can be computed for different combinations of indexes $i,\ j,\ l,$ and $k$ using an inner product definition that was given in Equations 41 and 40:

$$\left\langle \phi_i\phi_j\phi_l\phi_k \right\rangle = \int_{-1}^{1} \phi_i(\xi)\phi_j(\xi)\phi_l(\xi)\phi_k(\xi)\,d\xi = \left\langle \phi_k^2 \right\rangle D_{ijlk} \qquad (45)$$

whereas Legendre polynomials $\phi_i(\xi)$ were presented in Table 1 for $i=[0\ldots2]$. It is, however, easier to use the fact that integration over an odd function under the limits $[-1,1]$ always results in zero, which again eliminates all the tensor terms for which the sum $(i+j+l+k)$ is an odd number. Moreover, in cases when any one of the indexes $i,\ j$ or $l$ is zero, since $\phi_0(\xi)=1$, using Equations 6 and 49 and choosing, for instance, $j=0$:

$$\left\langle \phi_i\phi_0\phi_l\phi_k \right\rangle \equiv \left\langle \phi_k^2 \right\rangle D_{ijlk} = \left\langle \phi_i\phi_l\phi_k \right\rangle \equiv \left\langle \phi_k^2 \right\rangle C_{ilk} \qquad (46)$$

so that:

$$for\ j=0: \quad D_{ijlk} = C_{ilk} \qquad (47)$$

whereas the $3^{rd}$-order tensor coefficients $C_{ilk}$ in Equation 47 can be found in Table 2. In the first part of the resulting PCT expanded model (Equation 27) a term $\int_{-1}^{1} F_1\left(t\right)\phi_k d\xi$ contributes only when $k=0$ due to the properties of Legendre polynomials. For the simplest case of zero-order PCT expansion, the resulting model consisting of two differential equations is:

$$\frac{dh_{1,0}(t)}{dt} = \frac{1}{A_1}\left[F_1\left(t\right) - \frac{3}{4\sqrt{h_1^0}}k_{2,0}\, h_{1,0}(t)\right]$$

$$+\frac{1}{A_1}\left[-\frac{3}{8}k_{2,0}\sqrt{h_1^0} + \frac{1}{8}k_{2,0}\frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3}\right]$$

$$\frac{dh_{2,0}(t)}{dt} = \frac{1}{A_2}\left[\frac{3}{4\sqrt{h_1^0}}k_{2,0}\, h_{1,0}(t)\right]$$

$$+\frac{1}{A_2}\left[\frac{3}{8}k_{2,0}\sqrt{h_1^0} - \frac{3}{8}k_3\sqrt{h_2^0} + \frac{1}{8}k_3\frac{h_{2,0}(t)^2}{\left(\sqrt{h_2^0}\right)^3}\right]$$

$$+\frac{1}{A_2}\left[-\frac{3}{4\sqrt{h_2^0}}k_3 h_{2,0}(t) - \frac{1}{8}k_{2,0}\frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3}\right] \qquad (48)$$

For this zero-order expanded model, $C_{000}=1$ and $D_{0000}=1$ were used. Equation 48 can be analyzed as the PCT expanded system achieves steady-state. In this case, the derivatives of the expanded states will vanish which will result in two nonlinear equalities:

$$F_1^{ss} - \frac{3}{4\sqrt{h_1^0}} k_{2,0}\, h_1^{ss} - \frac{3}{8} k_{2,0} \sqrt{h_1^0} + \frac{1}{8} k_{2,0} \frac{\left(h_1^{ss}\right)^2}{\left(\sqrt{h_1^0}\right)^3} = 0$$

(49)

$$\frac{3}{4\sqrt{h_1^0}} k_{2,0}\, h_1^{ss} + \frac{3}{8} k_{2,0} \sqrt{h_1^0} - \frac{3}{8} k_3 \sqrt{h_2^0}$$

$$-\frac{3}{4\sqrt{h_2^0}} k_3 h_2^{ss} + \frac{1}{8} k_3 \frac{\left(h_2^{ss}\right)^2}{\left(\sqrt{h_2^0}\right)^3} - \frac{1}{8} k_{2,0} \frac{\left(h_1^{ss}\right)^2}{\left(\sqrt{h_1^0}\right)^3} = 0$$

Summing the two equalities in Equation 49 results in a more comfortable set:

$$F_1^{ss} - \frac{3}{4\sqrt{h_1^0}} k_{2,0}\, h_1^{ss} - \frac{3}{8} k_{2,0} \sqrt{h_1^0} + \frac{1}{8} k_{2,0} \frac{\left(h_1^{ss}\right)^2}{\left(\sqrt{h_1^0}\right)^3} = 0$$

(50)

$$F_1^{ss} - \frac{3}{8} k_3 \sqrt{h_2^0} - \frac{3}{4\sqrt{h_2^0}} k_3 h_2^{ss} + \frac{1}{8} k_3 \frac{\left(h_2^{ss}\right)^2}{\left(\sqrt{h_2^0}\right)^3} = 0$$

In both Equations 49 and 50, $h_1^{ss} = h_{1,0}\left(t \to \infty\right)$ and $h_2^{ss} = h_{2,0}\left(t \to \infty\right)$ are the steady-state values of the states, and $F_1^{ss} = F_1\left(t \to \infty\right)$ is the steady state value of the input. The equalities that appear in Equation 50 are second order polynomials that can be easily solved using traditional methods for a known steady state input value $F_1^{ss}$.

The first-order PCT expanded model consists of four differential equations with up to $P = 1$ terms in each. Using the values of $C_{ijk}$ and $D_{ijlk}$ for $k = 0$ and $k = 1$, one can obtain from Equation 44:

$$A_1 \frac{dh_{1,0}(t)}{dt} = F_1\left(t\right) - \frac{3}{4\sqrt{h_1^0}} k_{2,0}\, h_{1,0}(t)$$

$$-\frac{3}{8} k_{2,0} \sqrt{h_1^0} + \frac{1}{8} k_{2,0} \frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3} - \frac{1}{4} k_{2,1} \frac{h_{1,1}(t)}{\sqrt{h_1^0}}$$

$$+\frac{1}{24} k_{2,0} \frac{h_{1,1}(t)^2}{\left(\sqrt{h_1^0}\right)^3} + \frac{1}{12} k_{2,1} \frac{h_{1,0}(t)\, h_{1,1}(t)}{\left(\sqrt{h_1^0}\right)^3}$$

$$A_1 \frac{dh_{1,1}(t)}{dt} = \frac{3}{40} k_{2,1} \frac{h_{1,1}(t)^2}{\left(\sqrt{h_1^0}\right)^3} - \frac{3}{4} k_{2,0} \frac{h_{1,1}(t)}{\sqrt{h_1^0}}$$

$$+\frac{1}{4} k_{2,0} \frac{h_{1,0}(t)\, h_{1,1}(t)}{\left(\sqrt{h_1^0}\right)^3} - \frac{3}{4} k_{2,1} \frac{h_{1,0}(t)}{\sqrt{h_1^0}}$$

$$+\frac{1}{8} k_{2,1} \frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3} - \frac{3}{8} k_{2,1} \sqrt{h_1^0}$$

$$A_2 \frac{dh_{2,0}(t)}{dt} = -\frac{1}{24} k_{2,0} \frac{h_{1,1}(t)^2}{\left(\sqrt{h_1^0}\right)^3} + \frac{1}{4} k_{2,1} \frac{h_{1,1}(t)}{\sqrt{h_1^0}}$$

$$+\frac{1}{24} k_3 \frac{h_{2,1}(t)^2}{\left(\sqrt{h_2^0}\right)^3} - \frac{1}{12} k_{2,1} \frac{h_{1,0}(t)\, h_{1,1}(t)}{\left(\sqrt{h_1^0}\right)^3}$$

$$+\frac{1}{8} k_3 \frac{h_{2,0}(t)^2}{\left(\sqrt{h_2^0}\right)^3} - \frac{1}{8} k_{2,0} \frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3} + \frac{3}{4} k_{2,0} \frac{h_{1,0}(t)}{\sqrt{h_1^0}}$$

$$+\frac{3}{4} k_3 \frac{h_{2,0}(t)}{\sqrt{h_2^0}} + \frac{3}{8} k_{2,0} \sqrt{h_1^0} - \frac{3}{8} k_3 \sqrt{h_2^0}$$

$$A_2 \frac{dh_{2,1}(t)}{dt} = \frac{3}{8} k_{2,1} \sqrt{h_1^0} + \frac{3}{4} k_{2,0} \frac{h_{1,1}(t)}{\sqrt{h_1^0}}$$

$$-\frac{3}{40} k_{2,1} \frac{h_{1,1}(t)^2}{\left(\sqrt{h_1^0}\right)^3} - \frac{1}{8} k_{2,1} \frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3}$$

$$-\frac{3}{4} k_3 \frac{h_{2,1}(t)}{\sqrt{h_2^0}} + \frac{1}{4} k_3 \frac{h_{2,0}(t)\, h_{2,1}(t)}{\left(\sqrt{h_2^0}\right)^3} \tag{51}$$

$$+-\frac{1}{4} k_{2,0} \frac{h_{1,0}(t)\, h_{1,1}(t)}{\left(\sqrt{h_1^0}\right)^3} + \frac{3}{4} k_{2,1} \frac{h_{1,0}(t)}{\sqrt{h_1^0}}$$

For this first-order expanded model, coefficients $D_{1001} = D_{0101} = D_{0011} = 1$ and $D_{1111} = \frac{3}{5}$ were used based on Equation 47 and the values of corresponding coefficients $C_{ijk}$ from Table 2. Higher order PCT expanded models can be developed in a similar way. They include too many terms to be included in this paper.

Equation 51 represents a set of four nonlinear differential equations that can be presented in a general compact form:

$$\frac{d\underline{x}_{PCT}\left(t\right)}{dt} = \underline{f}_{PCT}\left(\underline{x}_{PCT}\left(t\right), \underline{u}_{PCT}\left(t\right)\right)$$

(52)

$$\underline{y}_{PCT}\left(t\right) = \underline{h}_{PCT}\left(\underline{x}_{PCT}\left(t\right)\right)$$

The uncertain model given in Equation 51 can be initially analyzed for the simplest case of no uncertainty in the variable $k_2$, i.e. $k_{2,1} = 0$. For this case, Equation 51 is reduced to:

$$A_1\frac{dh_{1,0}(t)}{dt}=F_1(t)-\frac{3}{4\sqrt{h_1^0}}k_{2,0}\,h_{1,0}(t)-\frac{3}{8}k_{2,0}\sqrt{h_1^0}$$
$$+\frac{1}{8}k_{2,0}\frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3}+\frac{1}{24}k_{2,0}\frac{h_{1,1}(t)^2}{\left(\sqrt{h_1^0}\right)^3}$$

$$A_1\frac{dh_{1,1}(t)}{dt}=-\frac{3}{4}k_{2,0}\frac{h_{1,1}(t)}{\sqrt{h_1^0}}+\frac{1}{4}k_{2,0}\frac{h_{1,0}(t)\,h_{1,1}(t)}{\left(\sqrt{h_1^0}\right)^3}$$

$$A_2\frac{dh_{2,0}(t)}{dt}=-\frac{1}{24}k_{2,0}\frac{h_{1,1}(t)^2}{\left(\sqrt{h_1^0}\right)^3}+\frac{1}{24}k_3\frac{h_{2,1}(t)^2}{\left(\sqrt{h_2^0}\right)^3}$$
$$+\frac{3}{4}k_{2,0}\frac{h_{1,0}(t)}{\sqrt{h_1^0}}+\frac{1}{8}k_3\frac{h_{2,0}(t)^2}{\left(\sqrt{h_2^0}\right)^3}+\frac{3}{8}k_{2,0}\sqrt{h_1^0}$$
$$-\frac{1}{8}k_{2,0}\frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3}-\frac{3}{4}k_3\frac{h_{2,0}(t)}{\sqrt{h_2^0}}-\frac{3}{8}k_3\sqrt{h_2^0}$$

$$A_2\frac{dh_{2,1}(t)}{dt}=\frac{3}{4}k_{2,0}\frac{h_{1,1}(t)}{\sqrt{h_1^0}}-\frac{1}{4}k_{2,0}\frac{h_{1,0}(t)\,h_{1,1}(t)}{\left(\sqrt{h_1^0}\right)^3}$$
$$-\frac{3}{4}k_3\frac{h_{2,1}(t)}{\sqrt{h_2^0}}+\frac{1}{4}k_3\frac{h_{2,0}(t)\,h_{2,1}(t)}{\left(\sqrt{h_2^0}\right)^3} \tag{53}$$

Moreover, if it is assumed that all the parameters are certain it would make sense to further simplify the model by eliminating the states distribution terms by setting $h_{1,1}(t)=0$ and $h_{2,1}(t)=0$, so that Equation 53 becomes

$$\frac{dh_{1,0}(t)}{dt}=\frac{1}{A_1}\left[F_1(t)-\frac{3}{4\sqrt{h_1^0}}k_{2,0}\,h_{1,0}(t)\right]$$
$$+\frac{1}{A_1}\left[-\frac{3}{8}k_{2,0}\sqrt{h_1^0}+\frac{1}{8}k_{2,0}\frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3}\right] \tag{54}$$

$$\frac{dh_{2,0}(t)}{dt}=\frac{1}{A_2}\left[\frac{3}{4}k_{2,0}\frac{h_{1,0}(t)}{\sqrt{h_1^0}}+\frac{1}{8}k_3\frac{h_{2,0}(t)^2}{\left(\sqrt{h_2^0}\right)^3}+\frac{3}{8}k_{2,0}\sqrt{h_1^0}\right]$$
$$+\frac{1}{A_2}\left[-\frac{1}{8}k_{2,0}\frac{h_{1,0}(t)^2}{\left(\sqrt{h_1^0}\right)^3}-\frac{3}{4}k_3\frac{h_{2,0}(t)}{\sqrt{h_2^0}}-\frac{3}{8}k_3\sqrt{h_2^0}\right]$$

which in turn is identical to the first-order PCT model given in Equation 48. Equation 51 can be analyzed for the steady-state constraints. At steady state all the derivatives in Equation 51 must vanish which results in a set of four equality constraints in which the variables $h_{1,0}$, $h_{1,1}$, $h_{2,0}$, $h_{2,1}$ are the values of the expanded states at steady state. Given all the constant parameters and the steady-state input value $F_1$ the task of evaluating steady-state constraints for this first-order expanded PCT model reduces to solving a set of four second-order algebraic equations with four unknowns $h_{1,0}$, $h_{1,1}$, $h_{2,0}$, $h_{2,1}$.

### 5. Open-loop Results

The original system model was introduced in Equation 20. To solve that model one needs to make sure that the system is open-loop stable. The eigenvalues of the system given in Equation 20 are $\lambda_1=-\frac{k_2}{A_1}$ and $\lambda_2=-\frac{k_3}{A_2}$. To ensure open-loop stability, i.e. both eigenvalues less than $-1$, can select the following values for the parameters used in the model: $A_1=1$ [units of area], $A_2=2$ [units of $area$], $k_2=3$ [units of $\frac{area}{time}$], and $k_3=4$ [units of $\frac{area}{time}$]. Substituting these values into the original model in Equation 20, one gets:

$$\frac{dh_1(t)}{dt}=F_1(t)-3\sqrt{h_1(t)}$$
$$\frac{dh_2(t)}{dt}=1.5\sqrt{h_1(t)}-2\sqrt{h_2(t)} \tag{55}$$

Equation 55 is an open-loop stable system of two nonlinear differential equations, for which one can assume zero initial states: $h_1(0)=h_2(0)=0$. The steady-state values, $h_{1,ss}$ and $h_{2,ss}$ can be easily obtained from Equation 55 using $\frac{dh_i(t)}{dt}=0$. For a steady state input value $F_1^{ss}=12$ [units of $\frac{volume}{time}$] the result is: $h_1^{ss}=16$ [units of height], and $h_2^{ss}=9$ [units of height]. To obtain the solution of the model in Equation 55, one can linearize the model with respect to steady state values as shown below using first-order Taylor approximation.

#### *5.1. Zero-Order PCT Expansion*

The simplest zero-order PCT expanded model was derived earlier and is given in Equation 28. For the given problem with only one uncertain parameter $k_2$ it is expanded using Legendre polynomials up to order $0$ according to:

$$k_2=k_{2,0}\,\phi_0(\xi)=k_{2,0} \tag{56}$$

in which $\phi_0(\xi)=1$ was introduced earlier in Table 1. A uniformly distributed (dimensionless) random variable $\xi$ does not appear in the final expanded model given in Equation 29, and is only used for PCT expansion as was shown in the previous section. With analogy to the original problem where a parameter $k_2$ was assumed to have a steady state value of $3$, one can choose a mean value $k_{2,0}=3$ [units of $\frac{area}{time}$]. Substituting this mean value together with all the other parameters into Equation 28, yields the following zero-order PCT expanded model:

$$\frac{dh_{1,0}(t)}{dt} = \left[ F_1(t) - \frac{3}{2\sqrt{h_1^0}} h_{1,0}(t) - \frac{3}{2}\sqrt{h_1^0} \right]$$

(57)

$$\frac{dh_{2,0}(t)}{dt} =$$

$$\frac{1}{2}\left[ \frac{3}{2\sqrt{h_1^0}} h_{1,0}(t) + \frac{3}{2}\sqrt{h_1^0} - 2\sqrt{h_2^0} - \frac{2}{\sqrt{h_2^0}} h_{2,0}(t) \right]$$

This zero-order PCT expanded model that assumes no distribution in the uncertain parameter $k_2$ depends on the starting points $h_1^0$ and $h_2^0$ that were used for Taylor approximation. Since this work addresses deviations from steady conditions, it is assumed that Taylor approximation applied around the steady state points $h_1^{ss}$ and $h_2^{ss}$ so that for a steady state input value $F_1^{ss} = 12$ [units of $\frac{volume}{time}$]:

$$h_1^0 = h_1^{ss} = 16$$

(58)

$$h_2^0 = h_2^{ss} = 9$$

[units of height] and the Taylor approximation up to order 1 in Equation 22 reduces to

$$\sqrt{h_1} = 2 + \frac{1}{8} h_1$$

(59)

$$\sqrt{h_2} = \frac{3}{2} + \frac{1}{6} h_2$$

Now, if one inserts Equation 59 into the original nonlinear model in Equation 55, the latter becomes:

$$\frac{dh_1(t)}{dt} = \frac{1}{2} F_1(t) - \frac{3}{8} h_1(t)$$

(60)

$$\frac{dh_2(t)}{dt} = \frac{3}{16} h_1(t) - \frac{1}{3} h_2(t)$$

Substituting the values from Equation 58 into Equation 57, yields:

$$\frac{dh_{1,0}(t)}{dt} = \frac{1}{2} F_1(t) - \frac{3}{8} h_{1,0}(t)$$

(61)

$$\frac{dh_{2,0}(t)}{dt} = \frac{3}{16} h_{1,0}(t) - \frac{1}{3} h_{2,0}(t)$$

As expected, the two resulting models in Equations 60 and 61 are identical, for in the simplest zero-order PCT expanded case $h_i = h_{i,0}$. To obtain the solution of the linear model in Equation 60, can implement Laplace transform of each differential equation to get:

$$s\,H_1(s) - h_1(0) = \frac{1}{2} F_1(s) - \frac{3}{8} H_1(s)$$

(62)

$$s\,H_2(s) - h_2(0) = \frac{3}{16} H_1(s) - \frac{1}{3} H_2(s)$$

Eliminating the initial conditions $h_1(0) = h_2(0) = 0$ Equation 62 is simplified to

$$H_1(s) = \frac{1}{2} F_1(s) \frac{1}{\left(s + \frac{3}{8}\right)}$$

(63)

$$H_2(s) = \frac{3}{32} F_1(s) \frac{1}{\left(s + \frac{3}{8}\right)\left(s + \frac{1}{3}\right)}$$

where $H_i(s)$ and $F_1(s)$ are Laplace transforms in s-domain of $h_i(t)$ and $F_1(t)$, respectively. Given a constant steady state input value $F_1^{ss} = 12$ [units of $\frac{volume}{time}$] and applying the inverse Laplace transform on Equation 63, one obtains the solution to the original (or zero-order PCT) linearized model:

$$h_1(t) = 16 - 16e^{-\frac{3}{8}t}$$

(64)

$$h_2(t) = 9 - 81e^{-\frac{1}{3}t} + 72e^{-\frac{3}{8}t}$$

Figure 2 shows the comparison between the original nonlinear two tank model (blue) and the First-order Taylor approximated model (red).

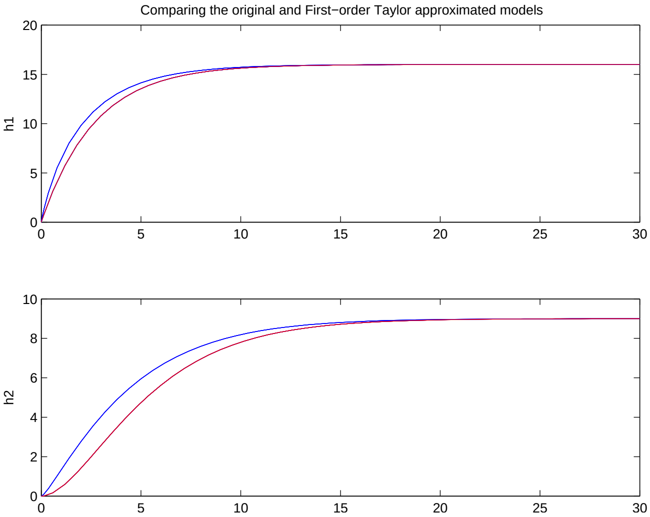


**Figure 2. Comparison between the original model and the First-order Taylor approximated model of two tanks.**

### *5.2. First-Order PCT Expansion*

The first order PCT expanded model was also derived earlier and given in Equation 29. For the given problem with only one uncertain parameter $k_2$ it is expanded using Legendre polynomial up to order $1$ according to:

$$k_2 = k_{2,0}\phi_0(\xi) + k_{2,1}\phi_1(\xi) = k_{2,0} + k_{2,1}\xi \tag{65}$$

in which $\phi_0(\xi) = 1$ and $\phi_1(\xi) = \xi$ were introduced earlier in Table 1. Similar to the previous case of zero-order PCT expansion, where a mean value of $k_{2,0} = 3$ [units of $\frac{area}{time}$] was used, now a first-order distribution value of $k_{2,1} = 0.3$ [units of $\frac{area}{time}$] is being used to introduce disturbance around the mean value in the uncertain parameter $k_2$. The starting points for Taylor approximation are also chosen as in Equa-

obsah
articles

tion 58. Substituting all the values into Equation 29, yields the following first-order PCT expanded model:

$$\frac{dh_{1,0}(t)}{dt}=\frac{1}{2}\left[2F_1(t)-\frac{3}{4}h_{1,0}(t)-\frac{1}{40}h_{1,1}(t)-12\right]$$

$$\frac{dh_{1,1}(t)}{dt}=\frac{1}{2}\left[-\frac{3}{4}h_{1,1}(t)-\frac{3}{40}h_{1,0}(t)-1.2\right] \tag{66}$$

$$\frac{dh_{2,0}(t)}{dt}=\frac{1}{4}\left[\frac{3}{4}h_{1,0}(t)+\frac{1}{40}h_{1,1}(t)-\frac{4}{3}h_{2,0}(t)\right]$$

$$\frac{dh_{2,1}(t)}{dt}=\frac{1}{4}\left[\frac{3}{4}h_{1,1}(t)+\frac{3}{40}h_{1,0}(t)-\frac{4}{3}h_{2,1}(t)+1.2\right]$$

Equation 66 now consists of four differential equations (with assumed zero initial conditions) that can be solved using Matlab [14] as a function of the process input $F_1$. Equation 66 can be rewritten in a state-space format presented in Equation 67 as:

$$\frac{d\underline{x}_{PCT}(t)}{dt}=A_{PCT}(t)\underline{x}_{PCT}(t)$$
$$+B_{PCT}(t)\underline{u}(t)+\Gamma_{PCT} \tag{67}$$

$$\underline{y}_{PCT}(t)=C_{PCT}(t)\underline{x}_{PCT}(t)$$
$$+D_{PCT}(t)\underline{u}(t)$$

where

$\underline{x}_{PCT}(t)=\left[h_{1,0}(t)\quad h_{1,1}(t)\quad h_{2,0}(t)\quad h_{2,1}(t)\right]^T$ is a vector of expanded states, and $\underline{y}_{PCT}(t)=\underline{x}_{PCT}(t)$ is the output vector that includes all the expanded states. Matrices $A_{PCT}$, $B_{PCT}$, $C_{PCT}$, $D_{PCT}$ and $\Gamma_{PCT}$ are identified as follows:

$$A_{PCT}=\begin{pmatrix}-\frac{3}{8} & -\frac{1}{80} & 0 & 0\\ -\frac{3}{80} & -\frac{3}{8} & 0 & 0\\ \frac{3}{16} & \frac{1}{160} & -\frac{1}{3} & 0\\ \frac{3}{160} & \frac{3}{16} & 0 & -\frac{1}{3}\end{pmatrix} \tag{68}$$

$$B_{PCT}=\begin{pmatrix}1\\0\\0\\0\end{pmatrix},\quad \Gamma_{PCT}=\begin{pmatrix}-6\\-0.6\\0\\0.3\end{pmatrix} \tag{69}$$

$$C_{PCT}=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix},\quad D_{PCT}=\begin{pmatrix}0\\0\\0\\0\end{pmatrix} \tag{70}$$

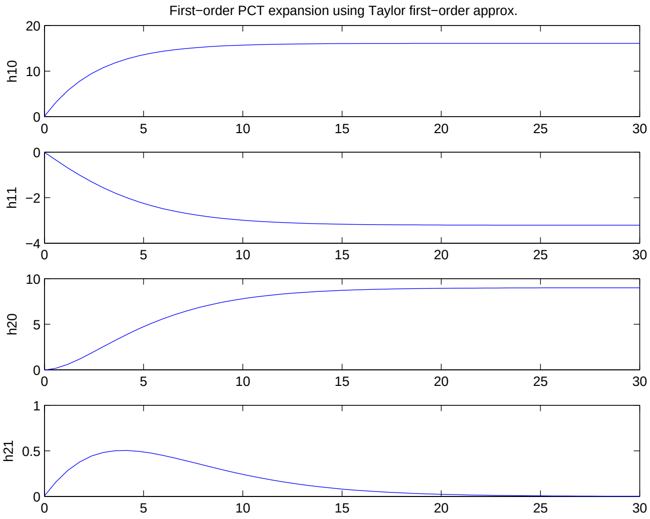


**Figure 3. First-order PCT expanded model showing the states mean values and distributions for *F₁*=12.**

Figure 3 shows the states mean values $h_{1,0}$ and $h_{2,0}$ and their respective distributions $h_{1,1}$ and $h_{2,1}$ as they achieve the steady state. At this point, it might be useful to apply the same disturbance to the original problem (with no application of Polynomial Chaos Theory). This can be done in Matlab.

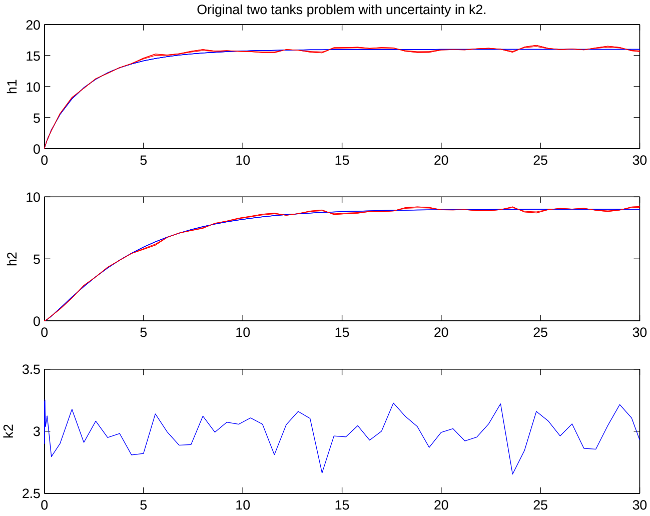


**Figure 4. The effect of uncertainty in the input parameter k_2 for the original problem.**

Figure 4 shows the states $h_1$ and $h_2$ for two cases: the original problem of a constant value of a parameter $k_2=3$ [units of $\frac{area}{time}$] (shown in blue) and the case when this parameter is subject to a random normal distribution of about $10\%$ around its mean value (shown in red).

## 6. Closed-loop Results

This section examines the closed-loop results for control of the two-tank model. Initially, NMPC control formulation was applied on the original two-tank model with uncertain valve coefficient $k_2$. The nonsquare $2x1$ control system is examined using NMPC controller formulation and the proposed soft constraint methodology. The results appear in Figures 5 and 6. The fact that the original model used in the case-study for this work consists of two outputs and only one input makes it impossible for the controller to track all possible setpoints. Moreover, only certain ratio of the steady-state values can be achieved in this case. Therefore, the authors were more interested in analyzing the ability to apply soft constraints on the process outputs and thus enable safe operation within the bounds of interest even when a small disturbance is applied. By tuning the controller parameters such as penalties in the cost function it is possible to push the process into the desired region or make it track a certain setpoint if absolutely needed.

### *6.1. NMPC with soft constraints on the process outputs applied on the original two-tank model*

Figure 5 shows closed-loop control of the original model with soft constraints on the outputs. This control operation used zero setpoint tracking penalties and equal soft constraints penalties for the two outputs. As can be observed from Figure 5 the controller is able to handle soft constraints on the outputs until a small disturbance hits the process at time $t = 18$. The disturbance causes the process to deviate from the previous bounds up until its removal. A smaller disturbance is then tested with the same setpoints and slightly enhanced soft constraints bounds on the outputs while the tuning is set to track the setpoint of the second output. The results appear in Figure 6. It follows from the plot that no deviation from the safety bounds is observed despite the disturbance being applied. It is worth noting that the uncertain parameter $k_2$ was simulated to randomly change within a small region around its steady-state value.

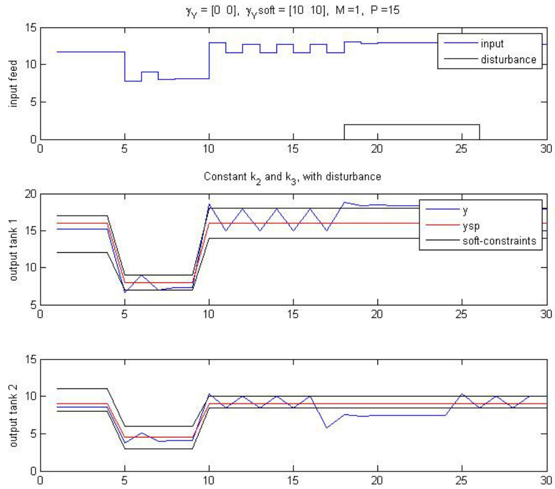


**Figure 5. NMPC with soft constraints on the process outputs with zero steady-state regulation penalty in the objective function. Observed no setpoint tracking and no soft-constraints violation until a small disturbance was applied and after its effect was removed.**

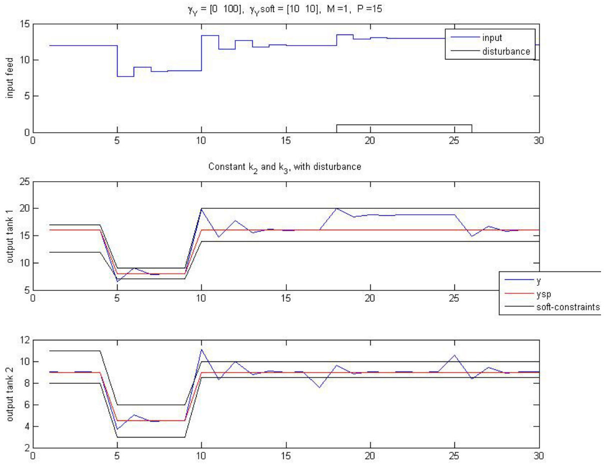


**Figure 6. NMPC with soft constraints on the process outputs with a large steady-state regulation penalty on the second output in the objective function. Shown setpoint tracking and no soft constraints violation even when a small disturbance was applied.**

### *6.2. NMPC with soft constraints on the process outputs applied on the PCT expanded model*

As was described earlier in the paper PCT analysis when used for control purposes enables direct control of various components of the expanded model. In particular, a possible control tuning strategy when trying to decrease the effect of parametric uncertainty on the system's performance might include an increased penalty on those components that represent uncertainty. The first-order PCT expanded two-tank model with one uncertain parameter $k_2$, used with second-order Taylor approximation, was developed in Section 4 and resulted in a dynamic nonlinear model with four states given in Equation 51. Two of the states represent the mean components of the original outputs while the other two indicate the deviation from the mean values. A $4x1$ control system is formulated for this expanded model with the feed into the first tank being the only manipulated input.

Different control strategies can be implemented at this point by changing the tuning parameters. Obviously, for a given nonzero uncertainty in parameter $k_2$, based on the PCT analysis developed in the previous chapters it is practically impossible to completely minimize distribution components without shutting down the flow into the system. In general control problems of this type a hard input constraint can be used to eliminate this natural optimization solution. In order to fulfill the research goals of this work, the authors tried to implement a control strategy that enables to apply soft constraints on all the expanded outputs thus decreasing the risk of process deviation from the safety or economically reasoned bounds.

Figure 7 represents the case for which the penalties in the cost function are tuned to maintain the mean components within certain bounds, while the components associated with uncertainty are unbound. Figure 8, alternatively, describes the case when only the distribution components are bound to certain limits. A comparison between the plots leads to a conclusion that it is possible to slightly decrease the uncertainty in the outputs using the proposed control methodology without changing the parametric uncertainty of the system.

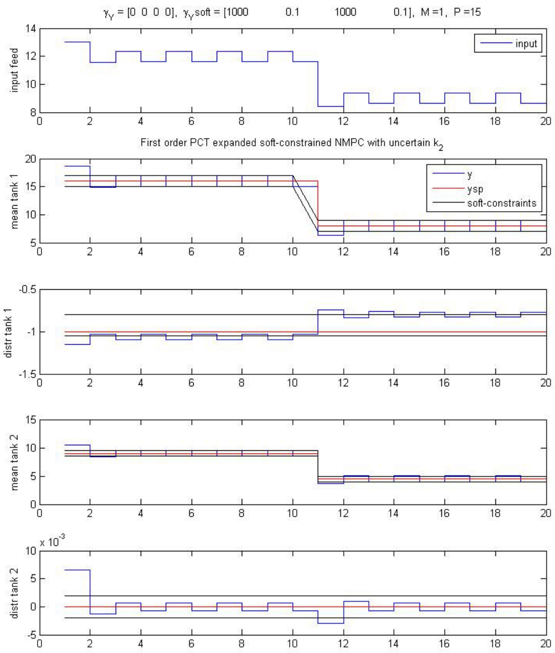


Figure 7. NMPC with soft constraints on the mean component of process outputs using a first-order PCT expanded model with one uncertain parameter $k_2$.

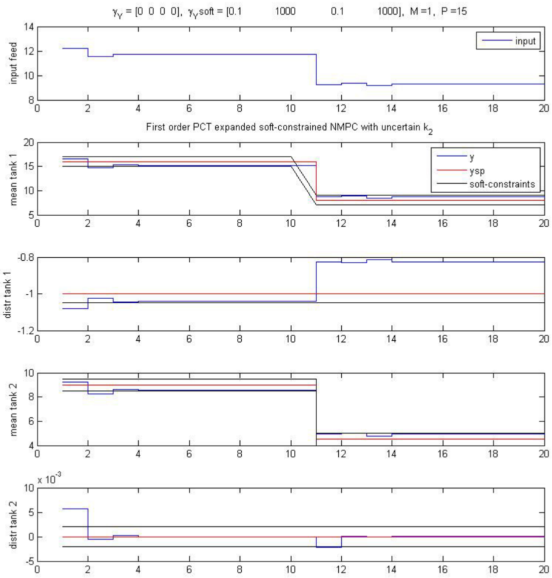


Figure 8. NMPC with soft constraints on the distribution components of process outputs using a first-order PCT expanded model with one uncertain parameter $k_2$.

## 7. Conclusions

Polynomial Chaos Theory analysis can be effectively applied on nonlinear systems with uncertain parameters. The main advantage of using PCT in a time domain lies in the ability to analytically obtain the expanded solution in a single computational run. The expanded system increases the number of the original states in the system depending on the order of PCT expansion and the number of uncertain parameters, so that the resulting dynamic PCT expanded model provides outputs that represent mean and distribution components of the original states. Provided the original distribution of the random variables these components can be reorganized to represent the original states. The multivariable expanded solution can be used along with the proposed Nonlinear Model Predictive Control formulation to control the individual outputs of the PCT expanded model. The use of NMPC controller can decrease the sensitivity of the model to changes in the uncertain parameters by applying large penalties in the cost function on those components that represent uncertainty. Application of soft constraints on the PCT expanded process outputs enables safe operation within certain bounds.

## Acknowledgments

The authors would like to acknowledge financial support from National Science Foundation CAREER Award CTS-0238663 as well ACS PRF Grant 47261-AC9.

## References

[1] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright. Advances in Control – Highlights of ECC '99, chapter Nonlinear Predictive Control and Moving Horizon Estimation – An Introductory Overview, pages 391–449. Springer-Verlag, London, 1999.

[2] F. Allgöwer and A. Zheng. Nonlinear Model Predictive Control. Birkhauser, 1993.

[3] C.-W. Shu B. Cockburn, G. E. Karniadakis. Discontinuous Galerkin Methods. Springer, 2000.

[4] L. T. Biegler. Efficient Solution of Dynamic Optimization and NMPC Problems. In International Symposium on Nonlinear Model Predictive Control: Assessment and Future Directions, pages 46–63, Ascona, Switzerland, 1998.

[5] L. T. Biegler and J. B. Rawlings. Optimization Approaches to Nonlinear Model Predictive Control. pages 543–571, Padre Island, TX, 1991.

[6] B. Ettedgui, M. Cabassud, M.-V. Le Lann, N. L. Ricker, and G. Casamatta. NMPC-based Thermal Regulation of a Fed-Batch Chemical Reactor Incorporating Parameter Estimation. pages 365–370, Banff, Canada, 1997.

[7] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: theory and practice-a survey. Automatica, 25(3):335–348, 1989.

[8] M. A. Henson. Nonlinear Model Predictive Control: Current Status and Future Directions. Computers and Chemical Engineering, (23):187–202, 1998.

[9] M. A. Henson and D. E. Seborg. Nonlinear Process Control. Prentice Hall, 1997.

[10] C. E. Long, P. K. Polisetty, and E. P. Gatzke. Nonlinear Model Predictive Control Using Deterministic Global Optimization. 16(6):635–644, 2006.

[11] T. Lovett. A Polynomial Chaos Approach to Network Simulation Under Uncertainty. PhD thesis, University of South Carolina, 2004.

[12] D. Plexousakis M. Papagelis. Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. Engineering Applications of Artificial Intelligence 18, pages 781–789, 2005.

[13] R. D. Marco, D. Semino, and A. Brambilla. From Linear to Nonlinear Model Predictive Control: Comparison of Different Algorithms. 36:1708–1716, 1997.

[14] The MathWorks. MATLAB 6.1. Prentice Hall, 2000.

[15] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained Model Predictive Control: Stability and Optimality. Automatica, 36:789–814, 2000.

[16] E. S. Meadows and J. B. Rawlilngs. Model Predictive Control. In M. A. Henson and D. E. Seborg, editors, Nonlinear Process Control, chapter 5, pages 233–310. Prentice Hall, 1997.

[17] P. Mhaskar and N. H. El-Farra an P. D. Christofides. Techniques for uniting Lyapunov-based and model predictive control. In International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control, pages 133–144, Freudenstadt, German, August 2005.

[18] K. R. Muske and J. B. Rawlings. The stability of constrained receding horizon control with state estimation. pages 2837–2841, Baltimore, MD, 1994.

[19] S. J. Qin and T. A. Badgwell. An Overview of Nonlinear MPC Applications. In Nonlinear Model Predictive Control: Assessment and Future Directions, Ascona, Switzerland, 1999. Birkhauser.

[20] R. S. Parker and E. P. Gatzke and R. Mahadevan and E. S. Meadows and F. J. Doyle III. Nonlinear Model Predictive Control: Issues and Applications. In Nonlinear Predictive Control, Theory and Practice, pages 33–58. The Institution of Electrical Engineering, London, 2001.

[21] C. V. Rao and J. B. Rawlings. Linear Programming and Model Predictive Control. Journal of Process Control, 10:283–289, 2000.

[22] R. G. Rice and D. D. Do. Applied Mathematics and Modeling for Chemical Engineers. John Wiley & Sons, New York, 1995.

[23] N. L. Ricker and J. H. Lee. Nonlinear Model Predictive Control of the Tennessee Eastman Challenge Process. Computers and Chemical Engineering, 19:961–981, 1995.

[24] P. O. M. Sckokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal Model Predictive Control (Feasibility Implies Stability). IEEE Transactions on Automatic Control, 44(3):648–654, 1999.

[25] J. Simminger, E. Hernández, Y. Arkun, and F. J. Schork. A Constrained Multivariable Nonlinear Model Predictive Controller Based on Iterative QDMC. Toulouse, France, 1991.

[26] S. Skogestad and I. Postlethwaite. Multivariable Feedback Control Analysis and Design. John Wiley & Sons, 605 Third Ave. New York, NY, 10158, 1 edition, 1996.

[27] Anton H.C. Smith. Robust and Optimal Control Using Polynomial Chaos Theory. PhD thesis, University of South Carolina, 2007.

[28] N. Wiener. The homogeneous chaos. Amer J. Math, pages 897–936, 1938.

[29] N. Wiener and A. Winter. The discrete chaos. Amer J. Math, pages 279–298, 1943.

[30] D. Xiu and G. E. Karniadakis. The wiener-askey polynomial chaos for stochastic differential equations. SIAM J. Sci. Comput., pages 619–644, 2002.

**T. L. Aliyev**
**E. P. Gatzke**

Department of Chemical Engineering
University of South Carolina Columbia, SC 29208
e-mail: aliyev@engr.sc.edu,gatzke@sc.edu

# Flight recovery system

**Pavel Hospodář, Martin Hromčík**

**Abstract**
Primary task of an automatic recovery system is to solve a situation when a pilot loses orientation. This space disorientation happens when there is a variance of angle position between what pilot thinks and real physical angle position of the airplane. This situation occurs firstly in case when the pilot cannot see the horizon (by low or zero visibility, during a night flight over monotonous terrain without distinct segmentation, when wrongly reading/failure of position indicators, with disturbance of the pilot and losing concentration under high pressure e.g.).

**Keywords:** automatic control, disorientation, θ-Φ diagram, MPC

## Introduction

Main reason of disorientation is the fact that human sensors of angular velocity (inner ear – semicircular channels) are insensitive to angular speeds under 2°/s. If the pilot does not have visual information this insensitiveness is integrating and becoming a drift. This drift leads to lose bearings. Automatic recovery system can stabilize the plane without involving the pilot and bring it to slightly climbing flight. Then the system hands over the control back to the pilot.

This whole procedure has a few limit factors. Above all it is a marginal angle of attack and sideslip angle (separation of the streamline from the profile – aerodynamic limitation), maximum pitch and roll rate (mechanical limitations of the plane), maximum folds in particular directions (physiological pilot protection) and a control limit (maximum helm deviation and maximum velocity of position change). On this account was a model predictive control (MPC) method chosen.

Superior decision level for MPC is a θ- Φ diagram. Diagram defines the procedure of controlling the plane in various plane positions. In dependence on its position angles it determines next control/motion of the plane.

## 1. Flight model

Flight motion characteristic is described by moment and force equations with six degrees of freedom [4][5]. These equations are strongly nonlinear due to influence of aerodynamic coefficients and describing these equations is very complicated. For control purposes it is preferred to use particular working point (airspeed, altitude, mass of aircraft, angle of attack ...) then linearize the model and store the result in state-space form [1].

*State-space model*

We have separated the model to longitudinal and lateral motion. State-space model used for control is derived from motion of training aircraft L-39. Original separated longitudinal and lateral motions are fused together and supplemented with equation of altitude and vertical speed [2]. Altitude equation is derived from vertical speed ( $\dot{h}$ ) which is directly depending on air speed (*V*), angle of attack (*α*) and pitch angle (*θ*):

$$\dot{h} = V \sin \gamma + u_z = V \sin(\theta - \alpha) + u_z \qquad (1)$$

If we suppose small fly path angle $\gamma$, we can write goniometric dependence[1] $\sin(\gamma) \approx \gamma$

$$\dot{h} = V \cdot \gamma + u_z = V(\theta - \alpha) + u_z \qquad (2)$$

where $u_z$ is wind disturbance. If we integrate last equation, we get altitude equation with initial condition $H_0$. States matrix is supplemented of altitude equation and output matrix is supplemented of vertical speed equation.

## 2. Model predictive control

This is a type of control which uses optimal state-feedback and predictive strategy for optimal design sequence of control action with reference to future states and outputs of system[3]. Discrete model is described as follows:

$$\begin{aligned} x(k+1) &= M \cdot x(k) + N \cdot u(k) \\ y(k) &= C \cdot x(k) + D \cdot u(k) \end{aligned} \qquad (3)$$

We search for control sequence $u(k)$ on the prediction horizon with length $T_p$ which minimizes following criterion:

$$J = \sum_{t=0}^{T-1} \left\{ q(t)[y(t) - w(t)]^2 + r(t)u(t)^2 \right\} \qquad (4)$$

Where $q(t)$ and $r(t)$ are weights of regulation deviation and control action, $w(t)$ is referential signal. Horizon of prediction is a time after which we find optimal control sequence. We specify the prediction of outputs as system response on the prediction horizon $T_p$:

[1] This form true only for angle lower than 5°

$$\begin{bmatrix} y(k) \\ y(k+1) \\ y(k+1) \\ \vdots \\ y(k+T_p-1) \end{bmatrix} = \begin{bmatrix} C \\ CM \\ CM^2 \\ \vdots \\ CM^{T_p-1} \end{bmatrix} \cdot x(k) + \\ + \begin{bmatrix} D & & & & \\ CN & D & & & \\ CMN & CN & D & & \\ \vdots & & & \ddots & \\ CM^{T_p-2}N & & & CN & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+T_p-1) \end{bmatrix} \qquad (5)$$

We can write this equation in matrix form:

$$y_k = V \cdot x(k) + S \cdot u_k = \widetilde{y}_k + S \cdot u_k \qquad (6)$$

Where $\widetilde{y}_k$ is system response to initial condition *x(k)* and $S.u_k$ is system response to control sequence on horizon of prediction $T_p$. We establish this equation to the scalar criterion:

$$J(u_k \mid x(k), w_k) = (\widetilde{y}_k + S \cdot u_k - w_k)^T Q(\widetilde{y}_k + S \cdot u_k - w_k) \\ + u_k^T R u_k \qquad (7)$$

Matrix $Q$ defines weight of regulation deviation and matrix $R$ defines weight of control values. Vector $w_k$ is reference sequence on prediction horizon. We can weigh only amplitude of inputs with this criterion formulation. If we want to weigh input change $\Delta u(k) = u(k) - u(k-1)$ we must modify previous criterion to the following form:

$$J(u_k \mid x(k), w_k) = (\widetilde{y}_k + S \cdot u_k - w_k)^T Q(\widetilde{y}_k + S \cdot u_k - w_k) \\ + \Delta u_k^T \cdot R \cdot \Delta u_k \qquad (8)$$

where:

$$\Delta u_k = D_i u_k - \widetilde{u}_k$$

$$D_i = \begin{bmatrix} 1 & 0 & & & \\ -1 & 1 & 0 & & \\ 0 & -1 & 1 & 0 & \\ & 0 & & \ddots & 0 \\ & & 0 & -1 & 1 \end{bmatrix} \quad \widetilde{u}_k = \begin{bmatrix} u(k-1) \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (9)$$

We obtain following form if we substitute previous equation to the equation (8):

$$J(u_k \mid x(k), w_k) = \widetilde{y}_k^T Q \widetilde{y}_k + \widetilde{y}_k^T Q S u_k - \widetilde{y}_k^T Q w + \\ + u_k^T S^T Q \widetilde{y}_k + u_k^T S^T Q S u_k - u_k^T S^T Q w - w^T Q \widetilde{y}_k \\ - w^T Q S u_k + w^T Q w + (D_i u_k)^T R (D_i u_k) - \widetilde{u}_k R D_i u_k + \\ + (D_i u_k)^T R \widetilde{u}_k + \widetilde{u}_k^T R \widetilde{u}_k = u_k^T (S^T Q S + D_i^T R D_i) u_k + \\ u_k^T \left[ (SQ)^T (\widetilde{y}_k - w) - D_i^T R \widetilde{u}_k \right] + \\ + \left[ (\widetilde{y}_k - w)^T Q S - \widetilde{u}_k R D_i^T \right] u_k^T + c \qquad (10)$$

Constant $c$ describes all variables which don't depend upon vector $u_k$ and haven't influence on optimization criterion. It is used quadratic programming for criterion solving and minimization function has following form:

$$\min_x J(x) = \min_x \frac{1}{2} x^T H x + f^T x \qquad (11)$$

Where matrix $H$ and vector $f$ are equal:

$$H = S^T Q S + D_i^T R D_i \\ f^T = (SQ)^T (\widetilde{y}_k - w) - D_i^T R \widetilde{u}_k \qquad (12)$$

*Constraints of inputs/outputs*

In light of limitations it is suitable to solve MPC regulation problematic by quadratic programming in terms of limiting particular parameters of regulation. We can solve constraint equation as: $A.x <= b$, where $x$ is optimal input vector on prediction horizon $T_p$. When determining maximal and minimal output values, we proceed as follows:

$$Y_{\min} \le y(k) \le Y_{\max} \\ Y_{\min} \le V \cdot x(k) + S \cdot u_k \le Y_{\max} \qquad (13)$$

We obtain finally inequality for output constraint only by simple translation:

$$S \cdot u_k \le Y_{\max} - V \cdot x(k) \\ S \cdot u_k \le -Y_{\min} + V \cdot x(k) \qquad (14)$$

We can formulate input constraint:

$$u_{\min} \le u_k \le u_{\max} \\ u_k \le u_{\max} \\ -u_k \le u_{\min} \qquad (15)$$

As next we define inequality for change of input constraint as:

$$du_{\min} \le \frac{u_k - u_{k-1}}{T_s} \le du_{\max} \\ u_k \le du_{\max} \cdot T_s + u_{k-1} \\ -u_k \le -du_{\min} \cdot T_s - u_{k-1} \qquad (16)$$

We can write whole inequality for constraint of optimal predictive control in matrix form:

$$A \cdot x \le b \\ \begin{bmatrix} -S \\ S \\ eye(n \cdot T_s) \\ -eye(n \cdot T_s) \\ eye(n \cdot T_s) \\ -eye(n \cdot T_s) \end{bmatrix} u_k \le \begin{bmatrix} Y_{\max} - V \cdot x(k) \\ -Y_{\min} + V \cdot x(k) \\ u_{\max} \\ u_{\min} \\ du_{\max} \cdot T_s + u_{k-1} \\ -du_{\min} \cdot T_s - u_{k-1} \end{bmatrix} \qquad (17)$$

Where $n$ is order of the system and $T_s$ is sample period. We can count matrix $A$**,** from previous equation, at the beginning of control, during control process it remains the same.

On the contrary it has to be counted limitating factor $b$ in every control step. Hereby is then proposed MPC regulator for linear model of aircraft motion with limited control magnitude and output values.

## 3. Recovery control

Until now we have not mentioned recovery control. In the concrete by which way/logic is recovery control leaded to stabilize the airplane. Such system logic can be preferably described by Φ-θ diagram. Location of the airplane is in every moment given by two values Φ (roll) and θ (pitch). Value of third position angle, heading, is not important for us in light of recovery control. Every such pair is represented by a point in Φ-θ diagram which is divided in sectors. In every sector is defined which part of control acts are turned on, i. e. which acts are to be provided.

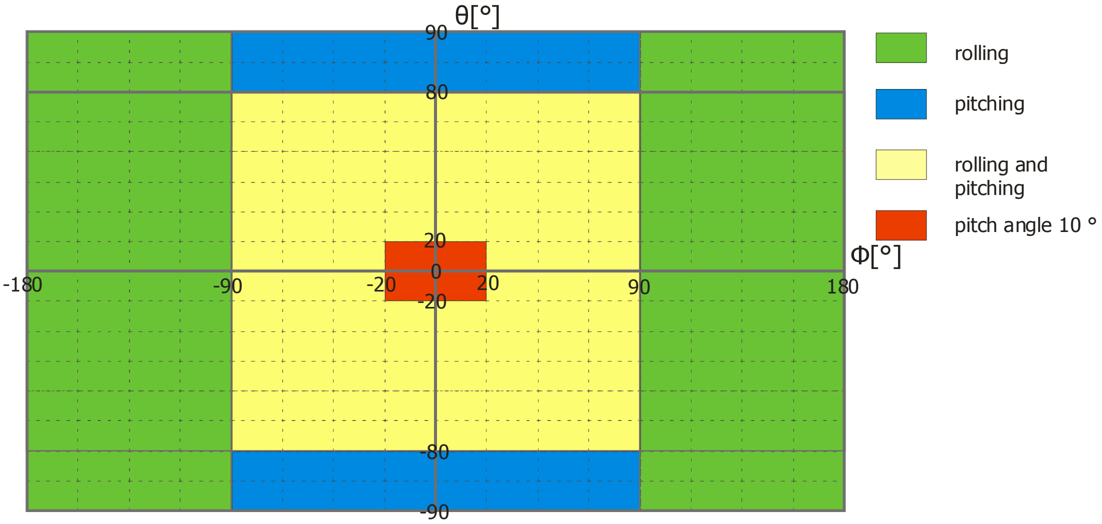


**Fig. 1 Φ-θ diagram**

Basic consideration is that the goal of recovery system is slightly climbing flight with zero roll. As next it is necessary to ensure primary rolling (roll change) at great Φ angles followed by simultaneous rolling and pitching (pitch change). Reversely when recovering the plane with big initial pitch it it necessary to ensure pitching first. Then with lower value of pitch can be provided simultaneous rolling and pitching. This is show on figure 1.

It is obvious that there are two marginal sectors. Firstly it is rolling sector where absolute roll value is greater than 90° and secondly pitching sector where absolute pitch value is greater than 80°. For pitch between -80° and 80° and roll between -90° and 90° applies synchronous rolling and pitching to zero mode. Finally for absolute values of elevation smaller than 20° is the airplane stabilized to zero roll and pitch of 10° to keep the airplane slightly climbing.

Following function realizes Φ-θ diagram by examining the output predicted on the prediction horizon $\boldsymbol{T_p}$. This is used to determine when the roll and pitch reaches requested angle.. For particular step is set the control from next sector. Here it is used control where we optimize value of actuating intervention when knowing requested reference in advance. This is set by Φ-θ diagram logic so that for green sector (only rolling) is pitch constant and reqest for roll is zero. For yellow sector (synchronous rolling and pitching) is reference for roll and pitch zero. And finally in red sector (slight climb) is roll reference zero and pitch reference 10°.

## 4. Simulation

In this paragraph is introduced example of simulation of control system where initial value of pitch is 85°, roll is 120° and heading is set to 45°. The airplane is then in almost vertical flight rewounded on the back. This can simulate the final phase of spiral during which the pilot lost orientation and the airplane goes spontaneously to spiral heading towards the ground thanks to unstable spiral mode.

On the figure 2 is shown that maximal input magnitudes values were not overcome. Transition speeds of control surface deviation are in bounds set by us. Then it is to see that multiples, or more precisely mechanical-physiological limitations, are under limit values. Flight limit factors, such as angle of attack/yaw angle, are also in bounds.

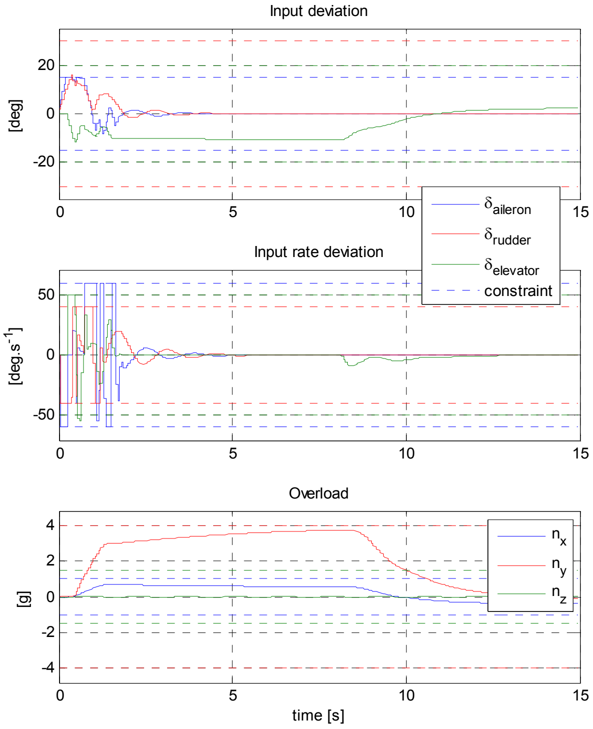


**Fig. 2 flight magnitudes during recovery**

**(inputs deviation, inputs rates deviation, acceleration)**

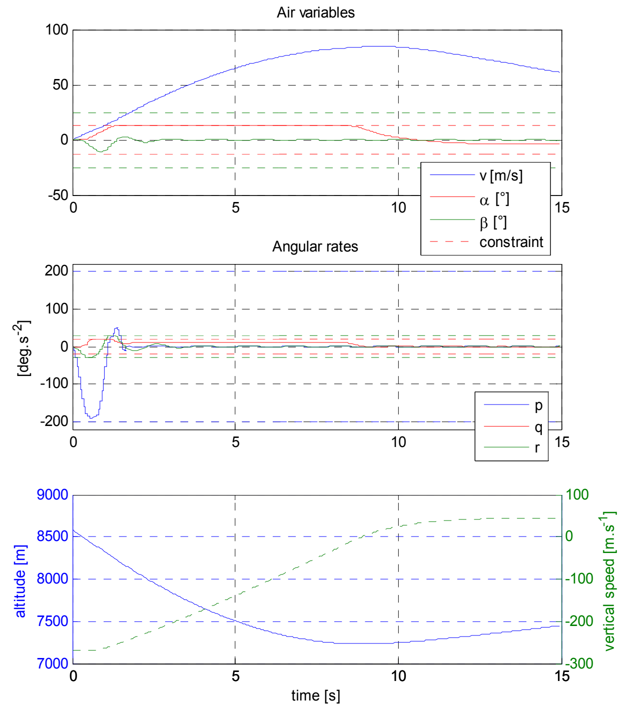


**Fig. 3 flight magnitudes during recovery**

**(air speed, angle of attack, side slipe angle, roll rate, pitch rate, yaw rate, altitude, vertical speed)**

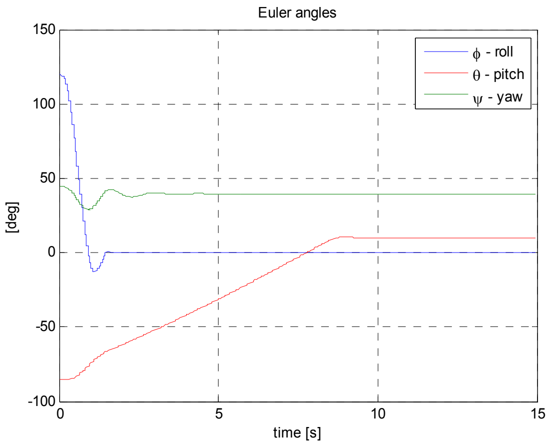


**Fig. 4 flight angles during recovery**

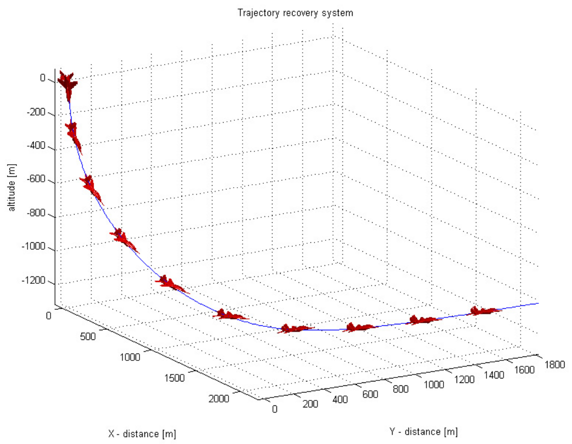


**Fig. 5 path of flight**

From course of elevation it is obvious that change of roll is much faster than by pitch. This is caused partly by mass persistence in particular axes and partly by limited angle velocities. It is worth noticing that for change of pitch is firstly used maximal roll velocity which has to be then lowered due to limited angle of attack.

## Conclusion

In this paper we have presented how to design MPC strategy to control a linear aircraft model with a superior decision function. An advantage of MPC strategy is prediction ability of states and outputs and in dependence on prediction horizon we can optimize inputs for whole recovery maneuver. MPC for a nonlinear aircraft model will be a subject of future work.

### Acknowledgments

This work was supported by the Ministry of Industry and Trade of the Czech Republic (MPO) under project No. FT-TA3/044.

### References

[1] PECH, Z., VĚK. V. Flight management system. ČVUT v Praze, (2006), ISBN 80-01-03374-0

[2] STEVENS, B., L., LEWIS, F., L.: Aircraft Control and Simulation. John Wiley and Sons, Inc., N.Y. (1992), ISBN 0471613975.

[3] HAVLENA, V., ŠTECHA, J., Modern control theory. ČVUT v Praze, (1999)

[4] MCRUER, D., ASHKENAS, I. AND GRAHAM, D., Aircraft dynamics and automatic control (1973), ISBN: 0-691-08083-6

[5] ETKIN, B. Dynamics of atmospheric flight. John Wiley and Sons, Inc., N.Y. (1972), ISBN 0-471-24620-4.

**Ing. Pavel Hospodář**
**Ing. Martin Hromčík, PhD.**

CTU at Prague
Faculty of Electrical Engineering
Department of Control Engineering
Karlovo náměstí 13, 12135
+420 224357681
hospik@post.cz
xhromcik@control.felk.cvut.cz

# Global optimization for parameter estimation of dynamic systems

Radoslav Paulen, Miroslav Fikar, Michal Čižniar, M. A. Latifi

**Abstrakt**

This work deals with the problem of finding a global solution for parameter estimation problem of a dynamic system described by a set of ordinary differential equations (ODE). Deterministic spatial branch and bound optimization algorithm is used to find the solution of problem. Upper bound is generated by sequential approach to dynamic optimization problem. Lower bound is provided by a solution of convex relaxation of the original problem. Selected examples from chemical engineering are solved and the resulting solution is discussed.

**Keywords:** global optimization, convex relaxation, ODE, parameter estimation

## Introduction

Human science is now so developed that the optimization problems became the part of daily life of every researcher, developer or designer in engineering, computational chemistry, finance and medicine amongst many other fields. This is especially true if we speak about parameter estimation, when we must find the best possible representation of phenomena (processes) happening in real world, plant or apparatus. Due to the fact that processes are usually described by set of differential equations, methods of dynamic optimization must be used to find (local) solution of optimization of these processes. For a number of years, researchers have known that dynamic optimization problems encountered in chemical engineering applications exhibit multiple local optima. This property, which can be attributed to non-convexity of the functions participating in most chemical engineering models, implies that standard local optimization methods will often yield suboptimal solutions to problems. Global optimization algorithms based on the deterministic approach such as generalized Benders decomposition [2, 5, 6], branch and bound [1, 12, 13] and interval analysis [7, 11, 14] guarantee the finite ε-conver-gence (convergence to the global optimum in finite computational steps for a given finite error tolerance) and the global optimality of the obtained solution. Spatial Branch-and-Bound (sBB) algorithms are the extension of traditional BB algorithms to continuous solution spaces. They are termed “spatial” because they successively partition the Euclidean space where the problem is defined into smaller and smaller regions where the problem is solved recursively by generating converging sequences of upper and lower bounds to the objective function value. In this work a deterministic sBB global optimization algorithm is used for global dynamic optimization problems with set of first-order parameter dependent differential equations in the constraints, where upper bound is obtained from solution of original dynamic optimization problem using sequential approach. A possible solution with simultaneous approach was described in uur previous work [3]. Lower bound is computed by solving the convex relaxed original problem with variable bounding proposed by [10]. Strategy of constant bounds is used instead of affine or αBB bounds (proposed by [9]) due to its usefulness and relative simplicity. The main purpose of this paper is to demonstrate utilization of sBB global optimization algorithm, to apply successive way to obtain bounds on variables participating on convex relaxation of original problem and to solve chosen examples relevant to chemical engineering.

## 1. Problem Statement

In this section we describe original non-convex dynamic optimization problem. Its solution gives an upper bound for sBB algorithm.

### 1.1 Dynamic Process Model

The processes considered is described by the following set of first-order parameter dependent, typically non-linear, differential equations

$$\dot{x} = f(t, x(t,p), p) \quad \forall t \in I \equiv [t_o, t_N] \tag{1}$$

where $t \in I \subset R$, denotes time as the independent variable and N is the number of points considered additionally to the initial point $t_0$, $p \subset R^r$ is the vector of parameters of the process, $x \subset R^n$ stands for the vector of state variables. The function f is such that $f : I \times R^n \times R^p \rightarrow R^n$. The solution x(t,p) of this set satisfies the initial condition

$$x(t_o, p) = x_o(p) \tag{2}$$

where the function $x_o$ is such that $x_o : R^r \rightarrow R^n$.

### 1.2 Process Constraints

Inequality constraints can be imposed at discrete time points, $t_i$ . These are point constraints of the form

$$g_i(x(t_i, p), p) \le 0 \quad i = 0,1,\dots,N \tag{3}$$

where the functions $g_i$, i=0,1,...,N, are such that $g_i : R^n \times R^r \rightarrow R^{s_i}$. Of course any equality point constraint can be replaced by two inequality point constraints. Lower and upper bounds are imposed on the parameters p:

$$p^L \leq p \leq p^U \qquad (4)$$

### 1.3 Objective Function

The objective function for a dynamic optimization problem can be expressed in terms of the values of the state variables at discrete points and of the parameters

$$J(x(t_i, p), p; i = 0,1,\ldots,N) \qquad (5)$$

The function J is such that J : $R^{n(N+1)}$ x $R^r$ → R. Integral terms that may appear in the objective function can always be eliminated by introducing additional state variables and equations in the set of differential equations.

### 1.4 Dynamic Optimization Problem

The formulation of the dynamic optimization problem studied is given by

$$\begin{aligned}
&\min_{p} J(x(t_i, p), p; i = 0,1,\ldots,N)\\
&s.t.\ \dot{x} = f(t, x(t,p), p) \quad \forall t \in I\\
&\qquad x(t_o, p) = x_o(p) \qquad\qquad (6)\\
&\qquad g_i(x(t_i, p), p) \leq 0 \quad i = 0,1,\ldots,N\\
&\qquad p^L \leq p \leq p^U
\end{aligned}$$

The following assumptions are made on the properties of the functions in (6):

- J(x($t_i$,p),p; i=0,1,...,N) is once continuously differentiable with respect to x($t_i$,p), i=0,1,...,N, and p on $R^{n(N+1)}$ x $R^r$.
- each element of $g_i$(x($t_i$,p),p, i=0,1,...,N), is once continuously differentiable with respect to x($t_i$,p) and p on $R^n$ x $R^r$.
- each element of f(t,x,p) is continuous with respect to t and once continuously differentiable with respect to x and p on I x $R^n$ x $R^r$.
- each element of $x_o$(p) is once continuously differentiable with respect to p on $R^r$.
- f(t,x,p) satisfies a uniqueness condition on *I* x $R^n$ x $R^r$.

The sequential approach is used for the solution of this dynamic optimization problem. The gradients with respect to p can be evaluated using the parameter sensitivities. These are given from the solution of the sensitivity equations [15, 16]. Due to the generally non-convex nature of the functions used in the formulation of the dynamic optimization problem, the solution obtained using the sequential approach and a standard gradient-based NLP technique, is a local optimum and therefore provides an upper bound for the global optimum solution.

## 2. Convex Relaxation of Problem

As it was mentioned before, BB algorithms are operating with concept of relaxations. In this sectionwe briefly describe a possible convex relaxation presented in [9] of the non-convex dynamic optimization problem that was introduced in the previous section. The solution of this convex relaxation problem provides a lower bound for the global optimum of the non-convex problem. First, we reformulate the NLP problem (6) as

$$\begin{aligned}
&\min_{\hat{x},p} J(\hat{x}, p)\\
&s.t.\ g_i(\hat{x}_i, p) \leq 0 \quad i = 0,1,\ldots,N\\
&\qquad \hat{x}_i = x(t_i, p) \quad i = 0,1,\ldots,N \qquad\qquad (7)\\
&\qquad p^L \leq p \leq p^U
\end{aligned}$$

where $\hat{x}$ is a vector of new added optimized variables and values of x($t_i$,p), i=0,1,...,N, are obtained from the solution of the IVP

$$\begin{aligned}
&\dot{x} = f(t, x, p) \quad \forall t \in I \qquad\qquad (8)\\
&x(t_o, p) = x_o(p)
\end{aligned}$$

### 2.1 Bounding Variables and Solution of IVP

It is very useful and in many cases essential to have bounds on variables, which are participating in optimization problem. For case of problem (7) bounds on parameters p are user–defined and bounds on variables $x_i$ can depend just on bounds of these parameters. Within the generation of bounds on $x_i$, which will definitely replace the presence of dynamic information in (7), relaxation of dynamic information will be formed. The dependence of convex relaxations on variable bounds is a common feature of deterministic global optimization algorithms. Since state variables appear in the non–convex objective function and constraints, a method for the derivation of rigorous bounds on these variables at point $t_i$, i=0,1,...,N, is needed. This issue can be resolved by generating bounds on the solution space of the dynamic system. Lower and upper parameter independent bounds can be determined for the solution x(t,p) of IVP (8) such that

$$\underline{x} \leq x(t, p) \leq \bar{x} \quad \forall p \in [p^L, p^U], \forall t \in I \qquad (9)$$

where the inequalities are understood component-wise. Considering the assumptions and theorem given in [9] it can be assumed that, if f is continuous and satisfies a uniqueness condition on $I_0$≡($t_0$, $t_N$] x $R^n$ x [$p^L$, $p^U$], then the solution $\underline{x}(t)$ and $\bar{x}(t)$ of the following IVP satisfies

$$\begin{aligned}
&\dot{\underline{x}}_k = \inf\ f_k(t, \underline{x}_k, [\underline{x}_{k-}, \bar{x}_{k-}], [p^L, p^U] \ \ \forall t \in I\\
&\underline{x}(t_o) = \inf x_o([p^L, p^U]) \qquad\qquad k = 1,2,\ldots,n\\
&\dot{\bar{x}}_k = \sup f_k(t, \bar{x}_k, [\underline{x}_{k-}, \bar{x}_{k-}], [p^L, p^U]) \ \ \forall t \in I \qquad (10)\\
&\bar{x}(t_o) = \sup x_o([p^L, p^U]) \qquad\qquad k = 1,2,\ldots,n
\end{aligned}$$

These IVPs provide a practical procedure to construct bounding trajectories for IVP (8) if the appropriate continuity and uniqueness conditions are satisfied. Natural interval extensions are used as inclusion functions [8].

### 2.2 Convex Relaxation of Dynamic Information

The set of equalities in (7) can be written as two sets of inequalities

$$\begin{aligned}
&\hat{x}_i - x(t_i, p) \leq 0 \quad i = 0,1,\ldots,N \qquad\qquad (11)\\
&x(t_i, p) - \hat{x}_i \leq 0 \quad i = 0,1,\ldots,N
\end{aligned}$$

Their relaxation is given by

$$\begin{aligned} &\hat{x}_i - \breve{x}^-(t_i,p) \le 0 \quad i=0,1,\dots,N \\ &\breve{x}(t_i,p) - \hat{x}_i \le 0 \quad i=0,1,\dots,N \end{aligned} \qquad (12)$$

where $\breve{x}$ denotes the convex underestimator of the specified function and $x^-(t_i,p)=-x(t_i,p)$. Thus, the function $\breve{x}$ $(t_i,p)$ is a convex underestimator of $x(t_i, p)$ and the function - $\breve{x}^-$ $(t_i, p)$ is a concave overestimator of $x(t_i,p)$. The generation of these under and overestimators is the most challenging step in the construction of the convex relaxation of the problem because no analytical form is available for $x(t_i,p)$. The constant bounds are given by inequalities

$$\underline{x}(t_i) \le \hat{x}_i \le \bar{x}(t_i) \quad i=0,1,\dots,N \qquad (13)$$

These inequalities are valid convex underestimators and concave overestimators for $x(t_i,p)$ and therefore they can replace inequalities (12). These bounds do not depend on the parameters p themselves, but do depend on the bounds on p.

### 2.3 Convex Relaxation of the NLP

After underestimating the objective function and overestimating the feasible region, the convex relaxation of the NLP problem (7) is given by

$$\begin{aligned} &\min_{\hat{x},p,w} J(\hat{x},p,w) \\ &s.t.\ g_i(\hat{x}_i,p,w) \le 0 \quad i=0,1,\dots,N \\ &\quad \underline{x}(t_i) \le \hat{x}_i \le \bar{x}(t_i) \quad i=0,1,\dots,N \\ &\quad C(\hat{x},p,w) \le 0 \\ &\quad p \in [p^L,p^U] \end{aligned} \qquad (14)$$

where J denotes the convex underestimator of the specified function, C denotes the set of additional constraints arising from the convex relaxation of non-convex terms of special types and w denotes the vector of new variables introduced by this relaxation.

## 3. Examples

The global optimization algorithm presented in [10] was implemented using MATLAB 6.5. To obtain the bounds on variables we used the constant bounds approach (see section 2.2). This technique is relatively simple, but suitable for the examples solved in this work. Solution of NLP problems was found using function *fmincon*. It is an implementation of a general NLP solver, provided by the Optimization Toolbox, uses either a subspace trust region method, based on the interior–reflective Newton method, or a sequential quadratic programming method. The MATLAB function *ode45* was used for the integration of IVPs. It is an implementation of a Runge-Kutta method based on the Dormand-Prince pair. The interval calculations needed were performed explicitly using interval arithmetic. Interval arithmetic computations are shown for each example. The three examples are parameter estimation problems in chemical kinetics modeling. All the case studies were solved on a workstation Dell Optiplex GX250, 3 GHz Intel Pentium 4 CPU with 1GB RAM.

### 3.1 Example 1: Irreversible Liquid-phase Reaction of the First Order

First example is a parameter estimation problem with two parameters and two differential equations as the constraints. It was published in [4] as well as in [9]. It involves a first-order irreversible isothermal liquid-phase chain reaction.

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

The problem can be formulated as follows:

$$\begin{aligned} &\min_{k_1,k_2} \sum_{j=1}^{10}\sum_{i=1}^{2} (x_i(t=t_j,k_1,k_2) - x_i^{\exp}(t_j))^2 \\ &s.t.\ \dot{x}_1 = -k_1 x_1 \quad \forall t \in [0,1] \\ &\quad \dot{x}_2 = k_1 x_1 - k_2 x_2 \quad \forall t \in [0,1] \\ &\quad x_1(t=0,k_1,k_2) = 1 \\ &\quad x_2(t=0,k_1,k_2) = 0 \\ &\quad 0 \le k_1 \le 10 \\ &\quad 0 \le k_2 \le 10 \end{aligned} \qquad (15)$$

where $x_1$ and $x_2$ are the mole fractions of components A and B, respectively. $k_1$ and $x_2$ are the rate constants of the first and second reaction, respectively. $x_i(t_j)$ is the experimental point for the state variable i at time $t_j$. The points used are taken from [4]. Applying the procedure defined by (9) leads to the expressions

$$\begin{aligned} &\dot{\underline{x}}_1 = \inf\,(-[k_1^L,k_1^U]\times \underline{x}_1] \ \ \forall t \in [0,1] \\ &\underline{x}_{10} = \inf\, x_{10}([p^L,p^U]) \\ &\dot{\underline{x}}_2 = \inf\,([k_1^L,k_1^U]\times[\underline{x}_1,\bar{x}_1] - \\ &\quad -[k_2^L,k_2^U]\times \underline{x}_2) \ \ \forall t \in [0,1] \\ &\underline{x}_{20} = \inf\, x_{20}([p^L,p^U]) \end{aligned} \qquad (16)$$

$$\begin{aligned} &\dot{\bar{x}}_1 = \sup(-[k_1^L,k_1^U]\times \underline{x}_1] \ \ \forall t \in [0,1] \\ &\bar{x}_{10} = \sup x_{10}([p^L,p^U]) \\ &\dot{\bar{x}}_2 = \sup([k_1^L,k_1^U]\times[\underline{x}_1,\bar{x}_1] - \\ &\quad -[k_2^L,k_2^U]\times \underline{x}_2) \ \ \forall t \in [0,1] \\ &\bar{x}_{20} = \sup x_{20}([p^L,p^U]) \end{aligned} \qquad (17)$$

which using the interval arithmetic calculation results in four bounding IVPs

$$\begin{aligned} &\dot{\underline{x}}_1 = -k_1^U\,\underline{x}_1 \quad \forall t \in [0,1] \\ &\underline{x}_{10} = 1 \\ &\dot{\underline{x}}_2 = k_1^L\,\underline{x}_1 - k_2^U\,\underline{x}_2 \quad \forall t \in [0,1] \\ &\underline{x}_{20} = 0 \end{aligned} \qquad (18)$$

$$\begin{aligned} &\dot{\bar{x}}_1 = -k_1^L\,\underline{x}_1 \quad \forall t \in [0,1] \\ &\bar{x}_{10} = 1 \\ &\dot{\bar{x}}_2 = k_1^U\,\bar{x}_1 - k_2^L\,\underline{x}_2) \quad \forall t \in [0,1] \\ &\bar{x}_{20} = 0 \end{aligned} \qquad (19)$$

Solutions of these ODEs represent a convex underestimator and concave overestimator of the relaxed problem solution space. The global optimization algorithm converged with the relative convergence criterion ε set to 1e-2. The global optimum parameter found was $k_1$=5.0035 and $k_2$=1.0000 and the value of the objective function for the global optimum parameter was equal to 1.1856e-6. 3436 iterations were necessary for convergence of the algorithm in 2632 seconds. The experimental points and trajectories of state variables for global optimum are shown in Fig. 1. The upper bound calculation was performed once every 100 iterations. Results show that for this problem with simple dynamics and only two parameters, algorithm is efficient and quite fast while we need less than one hour to obtain a solution. This is because data were generated using integration of system with parameter values k = [5, 1] with no error added.

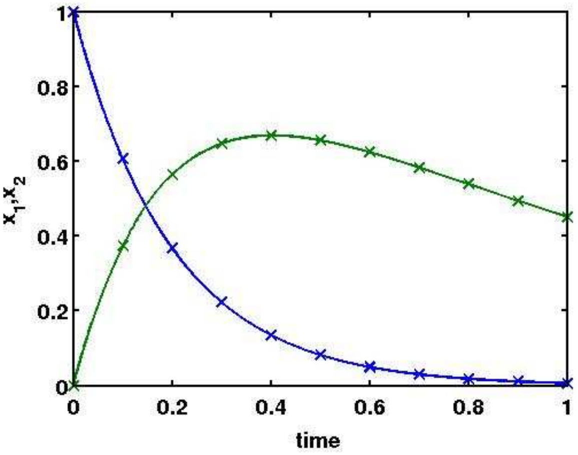


**Fig.1 Experimental points and state variable trajectories for the globally optimal parameters in Example 1**

### 3.2 Example 2: Catalytic Cracking of Gas Oil

This example is a parameter estimation problem with three parameters and two differential equations in the constraints. It appears in [4] and [10]. It involves an overall reaction of catalytic cracking of gas oil (A) to gasoline (Q) and other products (S).

$$A \xrightarrow{k_1} Q \xrightarrow{k_2} S$$

$$A \xrightarrow{k3} S$$

The problem is formulated as follows:

$$\min_{k_1,k_2,k_3} \sum_{j=1}^{20}\sum_{i=1}^{2}(x_i(t=t_j,k_1,k_2,k_3)-x_i^{\exp}(t_j))^2$$

$$\begin{aligned} s.t.\ \dot{x}_1 &= -(k_1+k_3)x_1^{\ 2} \quad \forall t\in[0,0.95] \\ \dot{x}_2 &= k_1x_1^{\ 2}-k_2x_2 \quad \forall t\in[0,0.95] \\ x_1(t&=0,k_1,k_2,k_3)=1 \\ x_2(t&=0,k_1,k_2,k_3)=0 \\ 0 &\le k_1 \le 20 \\ 0 &\le k_2 \le 20 \\ 0 &\le k_3 \le 20 \end{aligned} \qquad (20)$$

where $x_1$ and $x_2$ are the mole fractions of components A and Q, respectively. $k_1$, $k_2$ and $k_3$ are the rate constants of the respective reactions. $x_i(t_j)$ is the experimental measurement for the state variable i at time $t_j$. The measurement points used are again taken from [4]. Applying the procedure defined by (9) and using the interval arithmetic calculation resulted in following bounding IVPs

$$\begin{aligned} \dot{\underline{x}}_1 &= -(k_1^U+k_3^U)\underline{x}_1^2 \quad \forall t\in[0,0.95] \\ \underline{x}_{10} &= 1 \\ \dot{\underline{x}}_2 &= k_1^L\,\underline{x}_1^2 - k_2^U\,\underline{x}_2 \quad \forall t\in[0,0.95] \\ \underline{x}_{20} &= 0 \end{aligned} \qquad (21)$$

$$\begin{aligned} \dot{\overline{x}}_1 &= -(k_1^L+k_3^L)\overline{x}_1^2 \quad \forall t\in[0,0.95] \\ \underline{x}_{10} &= 1 \\ \dot{\overline{x}}_2 &= k_1^U\overline{x}_1^2 - k_2^L\overline{x}_2 \quad \forall t\in[0,0.95] \\ \overline{x}_{20} &= 0 \end{aligned} \qquad (22)$$

Solutions of these ODEs represent a convex underestimator and concave overestimator of the relaxed problem solution space. The globally optimal parameters are $k_1$=12.2111, $k_2$=7.9764 and $k_3$=2.2259 with the corresponding value of the objective function equal to 2.655e-3, when convergence criterion is set to 1e-2. The experimental points and the state variable trajectories for the global optimum are shown in Fig. 2. Algorithm converged after 8497 iterations and 13637 seconds of CPU time. The upper bound calculation was performed once every 100 iterations. In this example with non-linear dynamic system and three parameters to be estimated, rise of the computational effort is significant. This can be attributed to the higher complexity of the problem and also to small amount of random error added to the data integrated for parameters k=[12, 8, 2], which is evident from value of the objective function in optimum.

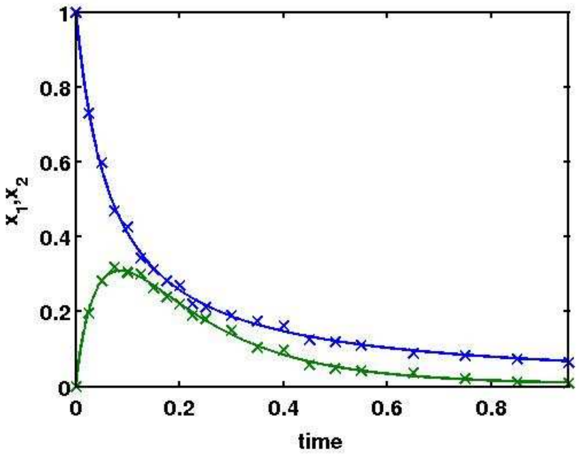


**Fig.2 Experimental points and state variable trajectories for the globally optimal parameters in Example 2**

### 3.3 Example 3: Reversible Liquid-phase Reaction of the First Order

The third example is a parameter estimation problem with four parameters and three differential equations. It appears in [4]. It involves a first-order reversible isothermal liquid-phase chain reaction.

$$A \xrightarrow{k_1} B \xrightarrow{k_3} C$$

$$C \xrightarrow{k_4} B \xrightarrow{k_2} A$$

The problem can be formulated as follows:

$$
\begin{aligned}
&\min_{\substack{k_1,k_2\\k_3,k_4}} \sum_{j=1}^{20}\sum_{i=1}^{3}(x_i(t=t_j,k_1,k_2,k_3,k_4)-x_i^{\exp}(t_j))^2\\
&s.t.\ \dot{x}_1=-k_1x_1+k_2x_2 \quad \forall t\in[0,1]\\
&\quad \dot{x}_2=k_1x_1-(k_2+k_3)x_2+k_4x_3 \quad \forall t\in[0,1]\\
&\quad \dot{x}_3=k_3x_2-k_4x_3 \quad \forall t\in[0,1]\\
&\quad x_1(t=0,k_1,k_2,k_3,k_4)=1\\
&\quad x_2(t=0,k_1,k_2,k_3,k_4)=0\\
&\quad x_2(t=0,k_1,k_2,k_3,k_4)=0\\
&\quad 0\le k_1\le 10\\
&\quad 0\le k_2\le 10\\
&\quad 10\le k_3\le 50\\
&\quad 10\le k_4\le 50
\end{aligned}
\tag{23}
$$

where x$_1$, x$_2$ and x$_3$ are the mole fractions of components A, B and, C, respectively. k$_1$, k$_2$ and k$_3$ are the rate constants of the first and second reaction, respectively. x$_i$(t$_j$) is the experimental point for the state variable i at time t$_j$. The points used are taken from [4]. Applying the procedure defined by (9) leads to the expressions which using the interval arithmetic calculation resulted in six bounding IVPs

$$
\begin{aligned}
&\dot{\underline{x}}_1=-k_1^U\underline{x}_1+k_2^L\underline{x}_2 \quad \forall t\in[0,1]\\
&\underline{x}_{10}=1\\
&\dot{\underline{x}}_2=k_1^L\underline{x}_1-(k_2^U+k_3^U)\underline{x}_2+k_4^L\underline{x}_3 \quad \forall t\in[0,1]\\
&\underline{x}_{20}=0\\
&\dot{\underline{x}}_3=k_3^L\underline{x}_2-k_4^U\underline{x}_3 \quad \forall t\in[0,1]\\
&\underline{x}_{30}=0
\end{aligned}
\tag{24}
$$

$$
\begin{aligned}
&\dot{\overline{x}}_1=-k_1^L\overline{x}_1+k_2^U\overline{x}_2 \quad \forall t\in[0,1]\\
&\underline{x}_{10}=1\\
&\dot{\overline{x}}_2=k_1^U\overline{x}_1-(k_2^L+k_3^L)\overline{x}_2+k_4^U\overline{x}_3 \quad \forall t\in[0,1]\\
&\overline{x}_{20}=0\\
&\dot{\overline{x}}_3=k_3^U\overline{x}_2-k_4^L\overline{x}_3 \quad \forall t\in[0,1]\\
&\underline{x}_{30}=0
\end{aligned}
\tag{25}
$$

Solutions of these ODEs represent a convex underestimator and concave overestimator of the relaxed problem solution space. The experimental points and trajectories of state variables for global optimum are shown in Fig. 3. Algorithm converged within the relative convergence criterion ε set to 1e-2. Global optimum was found with parameter values k$_1$=3.9990, k$_2$=1.9981, k$_3$=40.0000, k$_4$=20.0007 and the value of the objective function for the global optimum parameter was equal to 1.1856e-6. Algorithm converged after 44600 iterations and 142380 seconds of CPU time. The upper bound calculation was performed once every 100 iterations. Despite that there was no random error added to integrated data for parameters k=[4, 2, 40, 20] and system with linear dynamic embedded was studied, there is an enormous increase of computational time needed (almost 2 days). According to examples discussed previously the only explanation of this lies in augmented number of parameters together with extended parameter ranges.

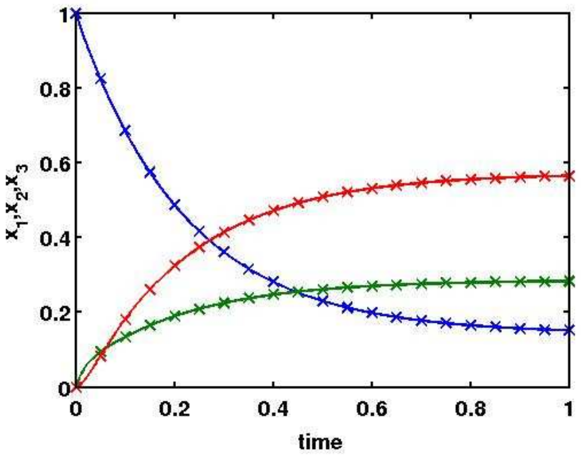


**Fig.3 Experimental points and state variable trajectories for the globally optimal parameters in Example 3**

## Conclusions

Main purpose of this work was to present application of a global optimization algorithm suitable for parameter estimation problems of dynamic systems. A deterministic sBB global optimization algorithm was employed. Local solutions produced using the sequential approach were used as an upper bound on the global minimum of the objective function value. Lower bounds were provided from the solution of a convex relaxation of the problem on subregions considered in the BB algorithm. This convex relaxation was achieved after defining a convex underestimation of the objective function and a convex overestimation of the feasible region. We implemented the algorithm proposed by [10] and used it to solve selected examples relevant to chemical engineering. The principle of constant bounds is very useful and also quite simple. Although there are some other few methods (affine bounds, αBB-bounds) already developed, we focused on this approach in the work. Explicit interval arithmetic calculations were used and solution of IVP (9) was performed once at each node of BB tree. Results show that the method of constant bounds for larger problems results in enormous rise of the number of iterations and computational time needed to obtain the global optimum. These results suggest that future work should be focused on larger systems with utilization of different bounding strategies.

### Acknowledgments

*The work has been supported by the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09 and 1/4055/07 and by the Slovak Research and Development Agency under the contract No. VV-0029-07. This support is very gratefully acknowledged.*

## References

[1] ADJIMAN, C. S., ANDROULAKIS, I. P., MARANAS, C.D., FLOUDAS. C. A.: A global optimization method, αBB for process design. Computers and Chemical Engineering, 20:419-424, 1996.

[2] BAGAJEWICZ, M., MANOUSIOUTHAKIS. V.: On the generalized Benders decomposition. Computers and Chemical Engineering, 15:691-700, 1991.

[3] ČIŽNIAR, M., PODMAJERSKÝ, M., HIRMAJER, T., FIKAR, M., LATIFI, M. A.: Global optimization for the parameter estimation of differential-algebraic systems. Chemical Papers, 2009. (accepted).

[4] ESPOSITO, W., FLOUDAS, C. A.: Global optimization for the parameter estimation of differential-algebraic systems. Industrial and Engineering Chemistry Research, 39:1291-1310, 2000.

[5] FLOUDAS, C. A.. VISWESWARAN. V.:A global optimization algorithm (GOP) for certain classes of non-convex NLPs - I theory. Computers and Chemical Engineering, 14:1397-1417, 1990.

[6] GEOFFRION, A. M.: Generalized Benders decomposition. Journal of Optimization Theory and Applications, 10:237-260, 1972.

[7] HAN, J. R., MANOUSIOUTHAKIS, V., CHOI. S. H.: Global optimization of chemical processes using the interval analysis. Korean Journal of Chemical Engineering, 14:270-276, 1997.

[8] NICKEL, K. I.: Using interval methods for numerical solution of ODE's. ZAMM, 66(11):513-523, 1986.

[9] PAPAMICHAIL, I., ADJIMAN, C. S.: A rigorous global optimization algorithm for problems with ordinary differential equations. Journal of Global Optimization, 24:1-33, 2002.

[10] PAPAMICHAIL, I., ADJIMAN, C. S.: Global optimization of dynamic systems. Computers and Chemical Engineering, 28:403-415, 2004.

[11] RATSCHEK, H., ROKNE, J.: New Computer Methods for Global Optimization. Ellis Horwood, Chichester, England, 1988.

[12] RYOO, H. S., SAHINIDIS, N. V.: Global optimization of nonconvex NLPs and MINLPs with applications in process design. Computers and Chemical Engineering, 19:551-566, 1995.

[13] SOLAND, R. M.: An algorithm for separable nonconvex programming problems II: Nonconvex constraints. Management Science, 17:759-773, 1971.

[14] VAIDYANATHAN, R., EL-HALWAGI, M.: Global optimization of nonconvex nonlinear programs via interval analysis. Computers and Chemical Engineering, 18:889-897, 1994.

[15] VASSILIADIS, V. S., SARGENT, R. W. H., PANTELIDES, C. C.: Solution of a class of multistage dynamic optimization problems. 1. problems without path constraints. Industrial and Engineering Chemistry Research, 33(9):2111-2122, 1994a.

[16] VASSILIADIS, V. S., SARGENT, R. W. H., PANTELIDES, C. C.: Solution of a class of multistage dynamic optimization problems. 2. problems with path constraints. Industrial and Engineering Chemistry Research, 33(9):2123-2133, 1994b.

**Ing. Radoslav Paulen**
**Prof. Dr. Ing. Miroslav Fikar, DrSc.**
**Ing. Michal Čižniar**
Slovak University of Technology
Faculty of Chemical and Food Technology
Institute of Information Engineering,
Automation and Mathematics
Department of Information Engineering and Process Control
Radlinského 9, 812 37 Bratislava
Tel.: 02/52 495 730
E-mail: radoslav.paulen@stuba.sk

**Prof. M. A. Latifi**
**Ing. Michal Čižniar**
Institut National Polytechnique de Lorraine
École Nationale Supérieure des Industries Chimiques
Laboratoire des Sciences du Genie Chimique
1 rue Grandville, 54 001 Nancy, France
E-mail: latifi@ensic.inpl-nancy.fr

# On-line neighboring-extremal controller design for setpoint transition in presence of uncertainty

Marián Podmajerský, Miroslav Fikar

**Abstract**

In this paper we present an approach suitable for optimal constrained control of processes subject to uncertainties. The controller follows a nominal solution of dynamic optimisation problem given by a theoretical model which needs not to be very accurate. The nominal optimal control trajectory is identified as a sequence of arcs and boundaries. Real output measurements are used to cancel model mismatch and to augment nominal inputs on-line using state-feedback law. Neighboring-extremal controller is designed to follow the nominal output trajectory along interior arcs using necessary conditions for optimality (NCO). Methodology will be implemented for setpoint-transition of van de Vusse reactor type. Finally, the performance of neighboring-extremal controller will be benchmarked using several perturbation scenarios.

**Key words:** NCO-tracking, dynamic optimisation problem, neighboring-extremal, necessary conditions for optimality

## Introduction

Batch and semi-batch plants are widely used in the industry and studied in academia for their non-linear behavior especially when consecutive and side reactions are presented. For these processes, the mathematical model is known with limited accuracy and controller design has to deal with variations. In the presence of model mismatches and uncertainties there are demands on advanced process control schemes.

In the last decade, the approaches which deal with limited model accuracy and with highly nonlinear behavior have been addressed. The presence of uncertainty can be solved using multiple approaches: Linear-Quadratic-Gaussian control [19], NCO-tracking [7, 13, 14, 15], robust $H_\infty$ loop-shaping [11, 19], adaptive control [16, 17], robust control [3, 4] or whole process re-optimisation: NMPC [1, 2, 8]. Most of these methods incorporate direct output measurements or reconstruct them with observers. There is a difference in the controller design and the implementation: if design is performed on-the-fly or can be done off-line; or if main controller implementation is simple and can be applied on commonly used hardware in the industry. Next limitation for controllers which perform on-line is sampling rate, especially for NMPC where whole optimisation process must be repeated, or for LQG and for adaptive control where controller parameters are also updated at each sampling period.

In this work, we apply NCO-tracking approach. Nominal optimal solution is used to calculate state-feedback gain matrices for state variations around the nominal trajectory. The augmented action is determined at each sampling time by adjustment of the nominal input profile with the pre-computed gain and the actual difference between the measurements and the nominal output profiles. The implementation then consist of the storage of the nominal input, output profiles, and the gain matrices for state-feedback at sampling periods in which the controls will be updated. In addition, a short sampling time allows to control the processes with the fast dynamic behavior.

The proposed NCO-tracking controller is applied on setpoint-transition of van de Vusse reactor model [10] in presence of parameter and initial state variations.

## 1. Theoretical Background

### 1.1 Optimal control problem

We assume the following dynamic optimisation problem with simple bound constraints:

$$\min J = \Phi(\mathbf{x}(t_f)) + \int_0^{t_f} L(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\theta})\mathrm{d}t \qquad (1)$$

s.t.

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\theta}); \quad \mathbf{x}(0) = \mathbf{x}_0 \qquad (2)$$

$$\mathbf{u}^L \le \mathbf{u}(t) \le \mathbf{u}^U \qquad (3)$$

where $t$ stands for the time variable, $t_f$ the fixed time, $\mathbf{u}$ the control vector, $\mathbf{x}$ the state vector with initial state $\mathbf{x}_0$, $\boldsymbol{\theta}$ the vector of uncertain time-invariant parameters, $\mathbf{F}$ the system dynamics, $J$ the scalar cost function to be minimised, $\Phi$ the terminal cost function, and $L$ integral cost function. All functions in (1)-(3) are assumed to be continuous and continuously differentiable with respect to their arguments. Then, there exists an unique optimal control solution $\mathbf{u}^*$ for given nominal parameter values $\bar{\boldsymbol{\theta}}$. This solution may consist of several arcs: boundary arcs (trajectories lie on the constraints) and interior arcs (trajectories within constraints).

### 1.2 Necessary Conditions for Optimality

According to [6], the Hamiltonian function $H$ is defined as follows

$$\begin{aligned} &H(\mathbf{x},\mathbf{u},\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{\mu}^L,\boldsymbol{\mu}^U) = L(\mathbf{x},\mathbf{u},\boldsymbol{\theta}) + \\ &+ \mathbf{F}(\mathbf{x},\mathbf{u},\boldsymbol{\theta})^T\boldsymbol{\lambda} + \boldsymbol{\mu}^{L^T}(\mathbf{u}^L - \mathbf{u}) + \boldsymbol{\mu}^{U^T}(\mathbf{u}^U - \mathbf{u}) \end{aligned} \tag{4}$$

where $\boldsymbol{\lambda}$ denotes adjoint vector function given by

$$\dot{\boldsymbol{\lambda}} = -\mathbf{H}_x = -\mathbf{F}_x^T\boldsymbol{\lambda} - \mathbf{L}_x; \quad \boldsymbol{\lambda}(t_f) = \boldsymbol{\Phi}_x(t_f) \tag{5}$$

The vectors and matrices with subscript $_x$ denote partial derivatives of the corresponding variable with respect to state $\mathbf{x}$.

Lagrange multiplier vector functions are denoted by $\boldsymbol{\mu}^L,\boldsymbol{\mu}^U$ and satisfy following conditions

$$\boldsymbol{\mu}^{L^T}(\mathbf{u}^L - \mathbf{u}) = \mathbf{0}; \qquad \mathbf{u}^L \geq \mathbf{0} \tag{6}$$

$$\boldsymbol{\mu}^{U^T}(\mathbf{u}^U - \mathbf{u}) = \mathbf{0}; \qquad \mathbf{u}^U \geq \mathbf{0} \tag{7}$$

Note that Lagrange multipliers are equal to zero $\boldsymbol{\mu}^L = \boldsymbol{\mu}^U = \mathbf{0}$ along an interior arc, while they are non-zero $\boldsymbol{\mu}_i{}^L \neq \boldsymbol{\mu}_i{}^U \neq 0, \quad i \in \{1,\ldots,n_u\}$ along a boundary arc. The first and second order necessary conditions of optimality for the problem described by (1)-(3) are of the form

$$\mathbf{H}_u = \mathbf{L}_u + \mathbf{F}_u^T\boldsymbol{\lambda} - \boldsymbol{\mu}^L + \boldsymbol{\mu}^U = \mathbf{0}; \quad \mathbf{H}_{uu} > 0 \tag{8}$$

where the positive definite matrix $\mathbf{H}_{uu} > 0$ denotes the second partial derivative of $H$ with respect to control $\left(\partial^2 H / \partial \mathbf{u}^2\right)$.

### 1.3 Neighboring-Extremal Control for Nonsingular Problems

Even a small disturbance in the model parameters results in changes of the optimal control trajectory $\mathbf{u}^*(t), 0 \leq t \leq t_f$. Let us consider the first-order approximation for augmented optimal trajectory of a perturbed control

$$\mathbf{u}(t;\zeta) = \mathbf{u}^*(t) + \zeta\delta\mathbf{u}(t) + \mathbf{o}(\zeta) \tag{9}$$

and use theory of neighboring extremal [6] for computing the correction $\delta\mathbf{u}$ in a such manner that the first-order variation of necessary conditions for optimality heads to zero along the augmented control $\mathbf{u}^*(t) + \zeta\delta\mathbf{u}(t)$. The correction of $\delta\mathbf{u}$ is computed as the solution to the variational LQ minimum problem [5, 9]

$$\begin{aligned} \min \delta^2 J(\delta\mathbf{u}) = \frac{1}{2}\delta\mathbf{x}(t_f)^T\boldsymbol{\Phi}_{\mathbf{xx}}\delta\mathbf{x}(t_f) + \\ + \frac{1}{2}\int_o^{t_f} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix}^T \begin{pmatrix} \mathbf{H}^*_{\mathbf{xx}} & \mathbf{H}^*_{\mathbf{xu}} \\ \mathbf{H}^*_{\mathbf{ux}} & \mathbf{H}^*_{\mathbf{uu}} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \end{pmatrix} \mathrm{d}t \end{aligned} \tag{10}$$

s.t.

$$\delta\dot{\mathbf{x}} = \mathbf{F}^*_x\delta\mathbf{x} + \mathbf{F}^*_u\delta\mathbf{u} + \mathbf{F}^*_\theta\delta\boldsymbol{\theta}; \tag{11}$$

$$\delta\mathbf{x}(0) = \delta\mathbf{x}_0 \tag{12}$$

$$\mathbf{u}^L - \mathbf{u}^*(t) \leq \delta\mathbf{u}(t) \leq \mathbf{u}^U - \mathbf{u}^* \tag{13}$$

that corresponds to minimisation of the second-order variation of the cost functional subject to the linearised dynamics. A superscript $^*$ (e.g. $\mathbf{H}^*_{\mathbf{uu}}$) means that the variable is evaluated upon nominal trajectories $\mathbf{u}^*(t)$, $\mathbf{x}^*(t)$, $\boldsymbol{\lambda}^*(t)$, for $0 \leq t \leq t_f$. A perturbed optimal control $\mathbf{u}(t;\zeta)$ exists in a neighbourhood of $\zeta = 0$, provided that the LQ problem (10)-(13) itself has an optimal solution [12]. The control variation $\delta\mathbf{u}$ satisfying the strengthened Legendre-Clebsch condition of positive definiteness $\mathbf{H}^*_{\mathbf{uu}} > \mathbf{0}$ and for unconstrained problems $\boldsymbol{\mu}^L(t) = \boldsymbol{\mu}^U(t) = \mathbf{0}$ is then given by

$$\delta\mathbf{u}(t) = -(\mathbf{H}^*_{\mathbf{uu}})^{-1}(\mathbf{H}^*_{\mathbf{ux}}\delta\mathbf{x}(t) + \mathbf{F}_{\mathbf{u}}^{*\mathrm{T}}\delta\boldsymbol{\lambda}(t) + \mathbf{H}^*_{\mathbf{u}\boldsymbol{\theta}}\delta\boldsymbol{\theta}) \tag{14}$$

where $\mathbf{x}(t)$ and $\boldsymbol{\lambda}(t)$ satisfy the following two-point boundary-value problem (TPBVP)

$$\begin{pmatrix} \delta\dot{\mathbf{x}}(t) \\ \delta\dot{\boldsymbol{\lambda}}(t) \end{pmatrix} = \begin{pmatrix} \mathbf{T}_{\mathbf{x}} & \mathbf{T}_{\boldsymbol{\lambda}} \end{pmatrix} \begin{pmatrix} \delta\mathbf{x}(t) \\ \delta\boldsymbol{\lambda}(t) \end{pmatrix} + \mathbf{T}_{\boldsymbol{\theta}}\delta\boldsymbol{\theta} \tag{15}$$

$$\delta\dot{\mathbf{x}}(0) = \delta\dot{\mathbf{x}}_0, \quad \delta\dot{\boldsymbol{\lambda}}(t_f) = \boldsymbol{\Phi}^*_{\mathbf{xx}}\delta\mathbf{x}(t_f) \tag{16}$$

where

$$\mathbf{T}_{\mathbf{x}} = \begin{pmatrix} \mathbf{F}^*_{\mathbf{x}} - \mathbf{F}^*_{\mathbf{u}}(\mathbf{H}^*_{\mathbf{uu}})^{-1}\mathbf{H}^*_{\mathbf{ux}} \\ -\mathbf{H}^*_{\mathbf{xx}} + \mathbf{H}^*_{\mathbf{xu}}(\mathbf{H}^*_{\mathbf{uu}})^{-1}\mathbf{H}^*_{\mathbf{ux}} \end{pmatrix} \tag{17}$$

$$\mathbf{T}_{\boldsymbol{\lambda}} = \begin{pmatrix} -\mathbf{F}^*_{\mathbf{u}}(\mathbf{H}^*_{\mathbf{uu}})^{-1}\mathbf{F}_{\mathbf{u}}^{*T} \\ -(\mathbf{F}^*_{\mathbf{x}} - \mathbf{F}^*_{\mathbf{u}}(\mathbf{H}^*_{\mathbf{uu}})^{-1}\mathbf{H}^*_{\mathbf{ux}})^T \end{pmatrix} \tag{18}$$

$$\mathbf{T}_{\boldsymbol{\theta}} = \begin{pmatrix} \mathbf{F}^*_{\boldsymbol{\theta}} - \mathbf{F}^*_{\mathbf{u}}(\mathbf{H}^*_{\mathbf{uu}})^{-1}\mathbf{H}^*_{\mathbf{u}\boldsymbol{\theta}} \\ -\mathbf{H}^*_{\mathbf{x}\boldsymbol{\theta}} + \mathbf{H}^*_{\mathbf{xu}}(\mathbf{H}^*_{\mathbf{uu}})^{-1}\mathbf{H}^*_{\mathbf{u}\boldsymbol{\theta}} \end{pmatrix} \tag{19}$$

Furthermore, a neighboring-extremal state-feedback law can alternatively be designed via backward sweep method [6], that assumes a linear relation between the state and adjoint variables and parameters $\delta\boldsymbol{\lambda}(t) = \mathbf{S}_{\mathbf{x}}(t)\delta\mathbf{x}(t) + \mathbf{S}_{\boldsymbol{\theta}}(t)\delta\boldsymbol{\theta}$

$$\delta u = -\mathbf{K}_{\mathbf{x}}(t)\delta\mathbf{x}(t) - \mathbf{K}_{\boldsymbol{\theta}}(t)\delta\boldsymbol{\theta}(t) \tag{20}$$

$$\mathbf{K}_{\mathbf{x}}(t) = (\mathbf{H}^*_{\mathbf{uu}})^{-1}(\mathbf{H}^*_{\mathbf{ux}} + \mathbf{F}_{\mathbf{u}}^{*\mathbf{T}}\mathbf{S}_{\mathbf{x}}(t)) \tag{21}$$

$$\mathbf{K}_{\boldsymbol{\theta}}(t) = (\mathbf{H}^*_{\mathbf{uu}})^{-1}(\mathbf{H}^*_{\mathbf{u}\boldsymbol{\theta}} + \mathbf{F}_{\mathbf{u}}^{*\mathbf{T}}\mathbf{S}_{\boldsymbol{\theta}}(t)) \tag{22}$$

$$\begin{aligned} &\dot{\mathbf{S}}_{\mathbf{x}}(t) = -\mathbf{H}^*_{\mathbf{xx}} - \mathbf{S}_{\mathbf{x}}(t)\mathbf{F}^*_{\mathbf{x}} - \mathbf{F}_{\mathbf{x}}^{*T}\mathbf{S}_{\mathbf{x}}(t) + \\ &+ (-\mathbf{H}^*_{\mathbf{xu}} + \mathbf{S}_{\mathbf{x}}(t)\mathbf{F}^*_{\mathbf{u}})\mathbf{K}_{\mathbf{x}}(t) \end{aligned} \tag{23}$$

$$\dot{\mathbf{S}}_{\mathbf{x}}(t_f) = \boldsymbol{\Psi}^*_{\mathbf{xx}} \tag{24}$$

$$\begin{aligned} &\dot{\mathbf{S}}_{\boldsymbol{\theta}}(t) = -\mathbf{H}^*_{\mathbf{x}\boldsymbol{\theta}} - \mathbf{S}_{\boldsymbol{\theta}}(t)\mathbf{F}^*_{\mathbf{x}} - \mathbf{F}_{\mathbf{x}}^{*T}\mathbf{S}_{\boldsymbol{\theta}}(t) + \\ &+ (-\mathbf{H}^*_{\mathbf{xu}} + \mathbf{S}_{\mathbf{x}}(t)\mathbf{F}^*_{\mathbf{u}})\mathbf{K}_{\boldsymbol{\theta}}(t) \end{aligned} \tag{25}$$

$$\dot{\mathbf{S}}_{\boldsymbol{\theta}}(t_f) = \mathbf{0} \tag{26}$$

It is implicitly assumed for constrained control sequence that the uncertainty is sufficiently small for the perturbed optimal control to have the same sequence of constrained and unconstrained arcs as the nominal solution. Neighboring-extremal is obtained similarly to unconstrained case: by solving either TPBVP or Riccati equation with possible discontinuities at junction times between constrained and unconstrained arcs. In practice, this assumption does not cause an apparent performance loss.

## 2. Design Example

### 2.1 Plant Model

We consider a chemical reactor with side and follow-up reactions of van de Vusse scheme [18], where desired cyclopentenol (B) is produced from cyclopentadiene (A) by acid-catalysed electrophilic addition of water in dilute solution. In addition, cyclopentanediol (C) is consecutive product of cyclopentenol (B) and addition of another water molecule, and dicyclopentadiene (D) is a side product of strong Diels-Alder reaction between the educt and the product.

The plant model presented in [10] consists of material balances of the reactant (A) and the product (B) as well as energy balances of the plant and the cooling jacket as follows

$$\dot{c}_A = -k_1(T)c_A - k_2(T)c_A^2 + (c_{in} - c_A)u_1, \tag{27a}$$

$$\dot{c}_B = k_1(T)(c_A - c_B) - c_B u_1, \tag{27b}$$

$$\dot{T} = h_r(c_A, c_B, T) + \alpha(T_c - T) + (T_{in} - T)u_1, \tag{27c}$$

$$\dot{T}_c = \beta(T - T_c) + \gamma u_2 \tag{27d}$$

with reaction enthalpy given as

$$h_r(c_A, c_B, T) = -\sigma[k_1(T)(c_A \Delta H_{AB} + c_B \Delta H_{BC}) + k_2(T)c_A^2 \Delta H_{AD}] \tag{28}$$

and kinetic rate constants are expressed as Arrhenius functions of temperature in °C

$$k_i(T) = k_{i0} e^{-\frac{\frac{E_i}{R}}{T+273.15}}, \quad i = 1,2. \tag{29}$$

We define states variables as $\mathbf{x} = \begin{bmatrix} c_A & c_B & T & T_c \end{bmatrix}^T$. The model parameters are defined in Tab.1.

The controlled inputs are input flow rate $q$ normalised by the volume of the plant $V_R$ and cooling system capacity $\dot{Q}$. Both inputs are constrained in the form of lower and upper bounds

$$u_1 = \frac{q}{V_R}, \quad 5h^{-1} \le u_1 \le 35h^{-1} \tag{30a}$$

$$u_2 = \dot{Q}, \quad -8500kJ.h^{-1} \le u_1 \le 0kJ.h^{-1} \tag{30b}$$

The product concentration and the plant temperature were chosen ascontrolled outputs

$$y_1 = c_B, \qquad y_2 = T. \tag{31}$$

The aim of the optimisation problem is to drive reactor's operational conditions from the original steady-state to another operational point. The particular numeric values of states and inputs at the operational points are summarised in Tab.1. The transition is performed with several scenarios, whereby the desired stationary point is always reached without violating input constraints. Thus, the performance index is defined as LQ integral functional where the normalised tracking error variations between original and new stationary point are driven to zero in a finite time $t_f = 20\,\text{min}$. The cost function then reads

$$\min_u J_0 = \int_0^{t_f} (\hat{\mathbf{y}}^T Q_I \hat{\mathbf{y}} + \hat{\mathbf{u}}^T R_I \hat{\mathbf{u}}) \mathrm{d}t \tag{32}$$

| |
|---|
| $\alpha = 30.8285 \quad [\mathrm{h}^{-1}]$ |
| $\beta = 86.688 \quad [\mathrm{h}^{-1}]$ |
| $\gamma = 0.1 \quad [\mathrm{K.kJ}^{-1}]$ |
| $\sigma = 3.556 \times 10^{-4} \quad [\mathrm{m}^3.\mathrm{K.kJ}^{-1}]$ |
| $k_{10} = 1.287 \pm 20\% \ \times 10^{12} \quad [\mathrm{h}^{-1}]$ |
| $\frac{E_1}{R} = 9758.3$ |
| $k_{20} = 9.043 \pm 20\% \ \times 10^{6} \quad [\mathrm{m}^3.\mathrm{mol}^{-1}.\mathrm{h}^{-1}]$ |
| $\frac{E_2}{R} = 8560$ |
| $\Delta H_{AB} = 4.2 \quad [\mathrm{kJ.mol}^{-1}]$ |
| $\Delta H_{BC} = -11 \quad [\mathrm{kJ.mol}^{-1}]$ |
| $\Delta H_{AD} = -41.85 \quad [\mathrm{kJ.mol}^{-1}]$ |
| $c_{in} = 5100 \pm 20\% \quad [\mathrm{mol.m}^{-3}]$ |
| $T_{in} = 104.9 \quad [\mathrm{K}]$ |
| $c_{A,sp1} = 3517.5 \quad [\mathrm{mol.m}^{-3}]$ |
| $c_{B,sp1} = 740 \quad [\mathrm{mol.m}^{-3}]$ |
| $T_{sp1} = 87 \quad [\mathrm{K}]$ |
| $T_{c,sp1} = 79.8 \quad [\mathrm{K}]$ |
| $u_{1,sp1} = 8.256 \quad [\mathrm{h}^{-1}]$ |
| $u_{2,sp1} = -6239 \quad [\mathrm{kJ.h}^{-1}]$ |
| $c_{A,sp2} = 2985 \quad [\mathrm{mol.m}^{-3}]$ |
| $c_{B,sp2} = 960 \quad [\mathrm{mol.m}^{-3}]$ |
| $T_{sp2} = 106 \quad [\mathrm{K}]$ |
| $T_{c,sp2} = 100.7 \quad [\mathrm{K}]$ |
| $u_{1,sp2} = 18.037 \quad [\mathrm{h}^{-1}]$ |
| $u_{2,sp2} = -4556 \quad [\mathrm{kJ.h}^{-1}]$ |

**Tab.1 Parameters for plant model and the main stationary setpoints**

where

$$\hat{\mathbf{y}} = \begin{bmatrix} \frac{c_B - c_{B,sp2}}{c_{B,sp2}} & \frac{T - T_{sp2}}{T_{sp2}} \end{bmatrix}^T \tag{33}$$

$$\hat{\mathbf{u}} = \begin{bmatrix} \frac{u_1 - u_{1,sp2}}{u_{1,sp2}} & \frac{u_2 - u_{2,sp2}}{u_{2,sp2}} \end{bmatrix}^T \tag{34}$$

and matrices $\mathbf{Q}_1$ and $\mathbf{R}_1$ are positive-definite and symmetric weight matrices

$$\mathbf{Q}_1 = \begin{bmatrix} q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & q_n \end{bmatrix}, \quad \mathbf{R}_1 = \begin{bmatrix} r_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_n \end{bmatrix}. \tag{35}$$

### 2.2 Open-loop Optimal Control

To find the optimal sequence of arcs, a numerical solution of dynamic optimisation problem (27)-(32) was obtained. We employed our in-house dynamic optimisation package based on CVP approach and implemented in MATLAB environment: SUNDIALS toolbox for the forward and backward numerical integration of differential equations and MATLAB version of SNOPT for NLP solution. For this particular case study, we parametrise both control inputs piecewise on 40 stages of equal width (30s).
The optimal control profiles of more aggressive control scenario are depicted in the third and fourth graph in

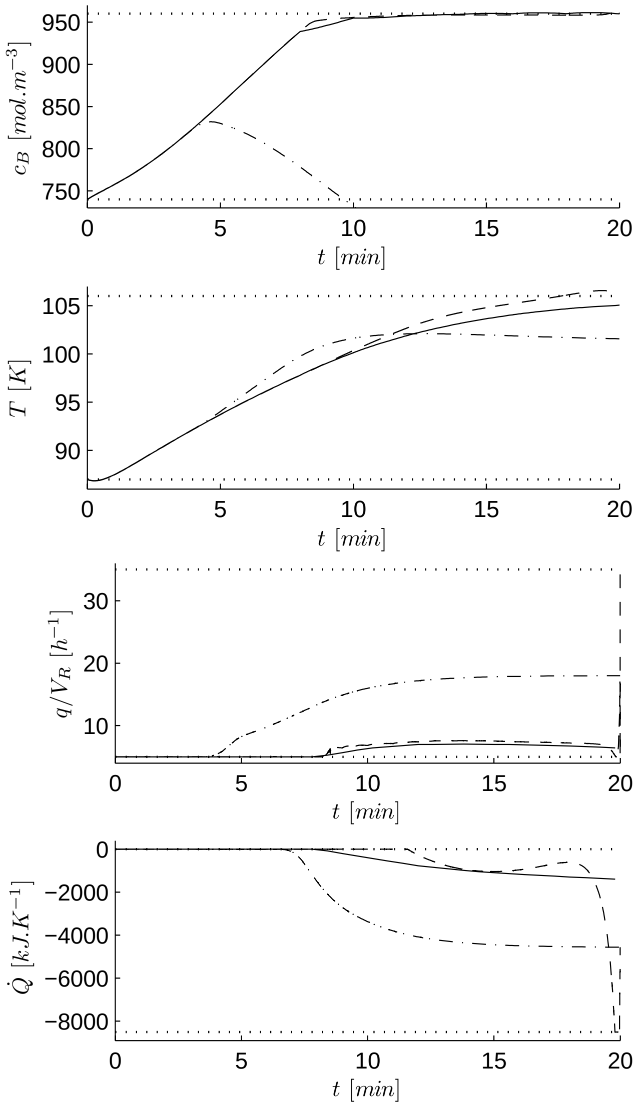


**Fig.1 More aggressive control with the perturbation scenario: $\Delta c_{in} = 20\%$, $\Delta k_{10} = -20\%$, $\Delta k_{20} = -20\%$. Dashed line: NCO tracking inputs; Solid line: optimal inputs to the perturbed problem; Dash-dotted line: open-loop nominal inputs**

Fig.1 (dash-dotted line). We can see that $u_1$ starts on upper bound and $u_2$ on lower bound and then they are followed by an interior arc. Similarly, the optimal control profiles of less aggressive control scenario are depicted in the third and fourth image in Fig 3. Observe, that upper bound of $u_1$ and lower bound for $u_2$ are shorter and the interior arcs are longer compared to more aggressive scenario. Along the interior arcs the following necessary conditions (see (8)) must hold for $u_1$ and $u_2$

$$H_{u1} = 2\left(\frac{u_1 - u_{1,sp2}}{u_{1,sp2}^2}\right) r_1 + \lambda_{cA}(c_{in} - c_A) - \lambda_{cB} c_B + \lambda_T (T_{in} - T) = 0 \quad (36)$$

$$H_{u2} = 2\left(\frac{u_2 - u_{2,sp2}}{u_{2,sú2}^2}\right) r_2 + \lambda_{T_c}\gamma = 0 \quad (37)$$

These equations give expressions for optimal control trajectories

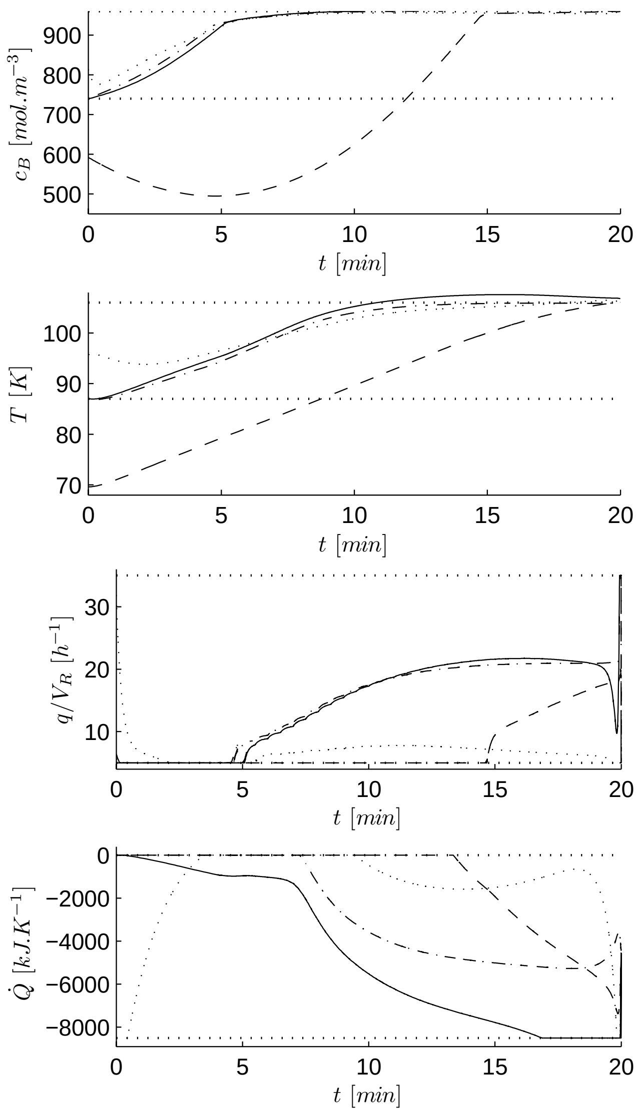


**Fig.2 Performance of NCO tracking with more aggressive control. Dashed line: C1, solid line: C2, dash-dotted line: C3, dotted line: C4**

$$u_1^* = \frac{\frac{2}{u_{1,sp2} r_1} - \lambda_{cA}(c_{in} - c_A)}{\frac{2r_1}{u_{1,sp2}^2}} + \frac{\lambda_{cB} c_B - \lambda_T (T_{in} - T)}{\frac{2r_1}{u_{1,sp2}^2}} \quad (38)$$

$$u_2^* = \frac{\frac{2}{u_{2,sp} 2r_2} - \lambda_{T_c}\gamma}{\frac{2r_2}{u_{2,sp2}^2}} \quad (39)$$

Note that optimal control trajectories $u_1^*$ and $u_2^*$ of the nominal problem are computed iteratively because adjoints become unstable during forward integration. The procedure is as follows. Dynamic process (27)-(29) is integrated forward, the controls are explicitly given from (38)-(39). The unknown adjoint variables $\boldsymbol{\lambda}^*$ are taken from the nominal solution and then they are approximated during forward integration. Subsequently, in next iteration step $\boldsymbol{\lambda}$ are corrected during backward integration. At final stage, $\boldsymbol{\lambda}_{t_f}$ at final time must be equal to $\boldsymbol{\lambda}(t_f)$ given by the optimal problem.

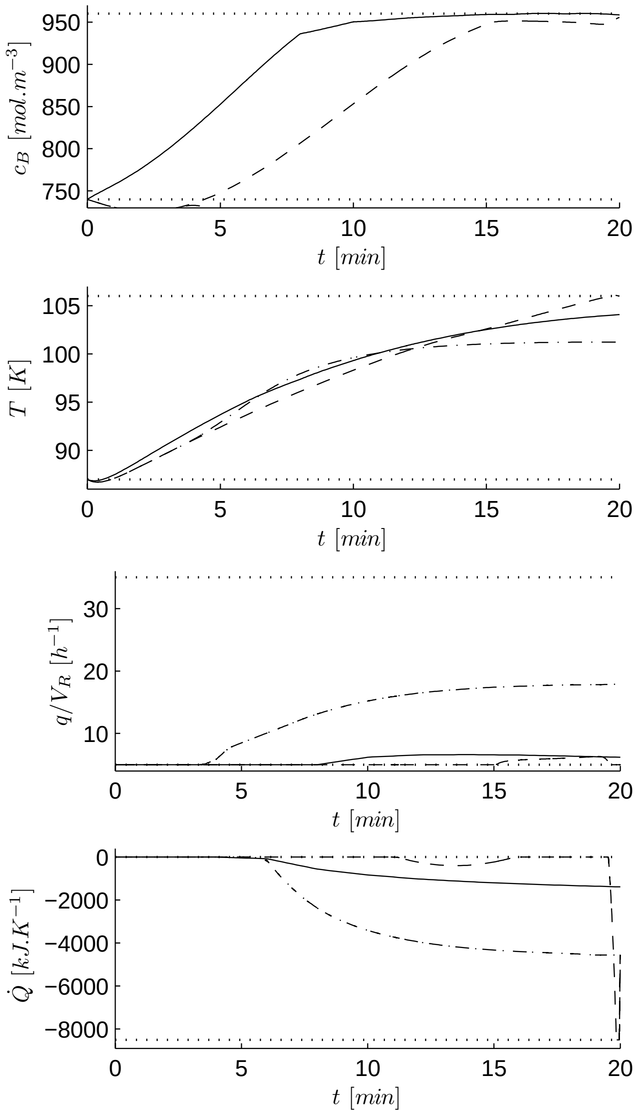


**Fig.3 More aggressive control with the perturbation scenario: $\Delta \mathbf{c}_{\text{in}} = -20\%$, $\Delta \mathbf{k}_{10} = -20\%$, $\Delta \mathbf{k}_{20} = -20\%$. Dashed line: NCO tracking inputs; Solid line: optimal inputs to the perturbed problem; Dash-dotted line: open-loop nominal inputs**

## 2.3 Neighboring-extremal Feedback Control

The standard approach of real-time optimisation consists of process model update using available measurements and followed by numerical re-optimisation that provides input to the plant. Instead of reoptimisation, the so called NCO-tracking approach is used in this work. The main idea is based on the fact that optimality requires meeting necessary conditions for optimality. NCO-tracking secures optimal operation via feedback without solving dynamic optimisation problem in real-time. The objective of NCO-tracking is to find zero gradients and to meet active constraints in presence of uncertainty that can be model mismatch or process disturbances. This will be handled on-line via neighboring-extremal controller.

### 2.3.1 NE controller design

In the section 2.2 we did analysis of the optimal control profiles of more and less aggressive scenario and we found a sequence of boundary and interior arcs that apply to the open-loop solution of the problem (1)-(3). Both inputs consist of a boundary arc followed by an interior arc, in more aggressive scenario and of the one interior arc, in less aggressive scenario. The optimal inputs $u_1^*$ and $u_2^*$) along the interior arcs, are given by (38)-(39). The switching times, between particular boundary arcs and interior arcs are taken from nominal solution. We assume that they are fixed and they are perturbated minimally. Only the interior arcs are updated in presence of uncertainty.

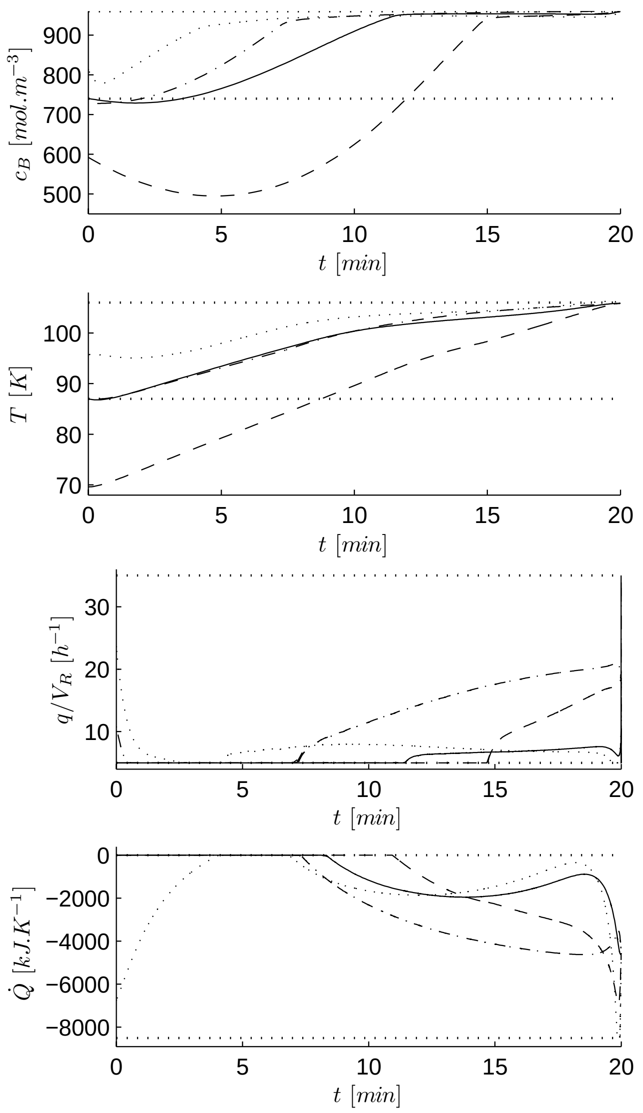


**Fig.4 Performance of NCO tracking with less aggressive control. Dashed line: C1, solid line: C2, dash-dotted line: C3, dotted line: C4**

The NE controller can be designed in two different ways (i) by solving TPBVP described by (15), (ii) by solving matrix Riccati equation (20). From both we get the gain matrices $\mathbf{K}_x$ and $\mathbf{K}_\theta$ that determine optimal control response. This drives the perturbed system towards original optimal output trajectory.

### 2.3.2 Performance of NE controller

To assess the performance of proposed NE controller two scenarios are studied: more and less aggressive control. Performance will be demonstrated with parameter uncertainty of inlet concentration $c_{in}$, vector of initial conditions $\mathbf{x}_0$, and kinetic rate constants $k_{10}$, $k_{20}$ that may vary in range of $\pm 20\%$. The weight coefficients are $r_1 = r_2 = 1$ and state penalisation is $q_1 = q_2 = 500$ for more aggressive and $q_1 = q_2 = 200$ for less aggressive scenario, respectively.

The corresponding control and response for various coefficient disturbances can be found in Fig. 1-2 for more aggressive, and in Fig. 3-4 for less aggressive scenario. Note that open-loop nominal controller is clearly unable to deal with presence of uncertainties, the desired setpoint is not reached in any case. In contrary, proposed NCO-tracking controller recovers influence of uncertainty  nd the reactor ends up at desired setpoint independent on controller aggressiveness. Same behavior can be observed in other simulations with various combinations of uncertainty, where the following cases were simulated:

$$\mathbf{C}_1 : \Delta c_{in} = 20\%, \quad \Delta k_{10} = -20\%, \quad \Delta k_{20} = 10\%, \quad \Delta x_0 = -20\%$$
$$\mathbf{C}_2 : \Delta c_{in} = -10\%, \quad \Delta k_{10} = -20\%, \quad \Delta k_{20} = 10\%$$
$$\mathbf{C}_3 : \Delta c_{in} = 10\%, \quad \Delta k_{10} = -10\%, \quad \Delta k_{20} = -20\%$$
$$\mathbf{C}_4 : \Delta c_{in} = -20\%, \quad \Delta x_0 = 10\%$$

In these cases, optimality loss is fully recovered while the input constraints remain satisfied and performance follow closely copy the original one.

## Concluding Remarks

Design of controller which tracks necessary conditions for optimality was presented and applied to the van de Vusse reactor model [10] with uncertainties which may occur under realistic conditions. The dynamic optimisation problem was transformed to the control problem through nominal input decomposition into sequence of boundary and interior arcs.

Neighboring-extremal controller was introduced to track the necessary conditions for optimality along interior arcs. The nominal optimal control as well as state-feedback law were calculated offline for all variations of states and parameters and for a small neighbourhood around nominal trajectories. The simulation results shown in Fig. 1-4 confirmed attractivity of proposed solution whereas desired setpoint was reached and inputs were within limits for both more and less aggressive control criterion. Opimality loss was successfully recovered in resence of parameter uncertainties. The approach is well-suited especially for the real-time optimisation with short sampling times.

### Acknowledgments

*The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09 and 1/4055/07. This work was supported by the Slovak Research and Development Agency under the contract No. VV-0029-07.*

## References

[1] ABEL, O., MARQUARDT, W.: A model predictive control scheme for safe and optimal operation of exothermic semi-batch reactors. In IFAC DYCOPS-5, pages 761–766, Corfu, Greece, 1998

[2] ALLGÖWER, F., ZHENG, A.: Nonlinear Model Predictive Control. Birkhäuser Verlag, 2000

[3] BAKOŠOVÁ, M., PUNA, D.: Selected Topics in Modelling and Control, volume 5, chapter Control of a Continuous-Time Stirred Tank Reactor via Robust Static Output Feedback, pages 38–44. Slovak University of Technology Press, 2007

[4] BAKOŠOVÁ, M., PUNA, D., VAŠIČKANINOVÁ A.: Robust control of chemical reactors. Acta Chimica Slovaca, 1(1):12–23, 2008

[5] BREAKWELL, J.V., SPEYER, J., BRYSON, A.E.: Optimization and control of nonlinear systems using the second variation. SIAM J. Control Ser. A 1 2, pages 193–223, 1963

[6] BRYSON, A.E., Yu-Chi Ho: Applied Optimal Control - Optimization, Estimation and Control. Hemisphere publishing corporation, 1975

[7] FRANÇOIS, G., SRINIVASAN, B., BONVIN, D.: Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. Journal of Process Control, 15:701–712, 2007.

[8] GARCIA, C.E., PRETT, D.M., MORARI, M.: Model predictive control: theory and practice – a survey. Automatica, 25(3):335–348, 1989

[9] KELLY, H.J., KOPP, R.E., MOYER, G.: A trajectory optimization technique based upon the theory of the second variation. AIAA Astrodynamics Conference, pages 193–223, 1963

[10] KLATT, K.U., ENGELL, S.: Gain scheduling control of a non-minimum-phase CSTR. In European Control Conference, pages 2323–2328, Groning, Netherlands, 1993

[11] MCFARLENE, D.C., GLOVER, K.: Robust Controller Design Using Normalized Coprime Factor Plant Descriptions (LNCIS). Springer, New York, 1989

[12] PESH, H.J.: The accessory minimum problem and its importance for the numerical computation of closed-loop controls. In Conference on Decision and Control, pages 952–953, Honolulu, HI, 1990

[13] SRINIVASAN, B., BONVIN, D.: Real-time optimization of batch processes by tracking the necessary conditions of optimality. Indus. Eng. Chem. Res., 46(2):492–504, 2007.

[14] SRINIVASAN, B., BONVIN, D.: Dynamic optimization under uncertainty via NCO tracking: A solution model approach. Technical report. In Journal of Process Control, Switzerland, 2004. Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne

[15] SRINIVASAN, B., BONVIN, D., VISSER, E.: Dynamic optimization of batch processes: II. role of measurements in handling uncertainty. Computing and Chemical Engineering, 44:27–44, 2003

[16] ÅSTRÖM, K.J., WITTENMARK, B.: Theory and application of adaptive control - A survey. Automatica, (19):471–486, 1983

[17] ÅSTRÖM, K.J., WITTENMARK, B.: Adaptive Control. Addison-Wesley, Massachusetts, 1989

[18] VAN DE VUSSE, J.G.: Plug-flow type reactor versus tank reactor. Chem. Eng. Sci, (19):994–998, 1964.

[19] ZHOU, K., DOYLE, J.C., GLOVER, K.: Robust and Optimal Control. Prentice Hall, Englewood Cliffs, New Jersey, 1995

**Ing. Marián Podmajerský**

Slovak University of Technology in Bratislava
Faculty of Chemical and Food Technology
Institute of Information Engineering, Automation, and Mathematics
Randlinského 9
812 37 Bratislava
E-mail: marian.podmajersky@stuba.sk

**Prof. Ing. Miroslav Fikar, DrSc.**

Slovak University of Technology in Bratislava
Faculty of Chemical and Food Technology
Institute of Information Engineering, Automation, and Mathematics
Randlinského 9
812 37 Bratislava
E-mail: miroslav.fikar@stuba.sk

# Control of a laboratory chemical reactor using robust PI controller

Jana Závacká, Monika Bakošová, Katarína Vaneková

**Abstract**
The paper presents a method for design of robust PI controllers for systems with interval uncertainty. The method is based on plotting the stability boundary locus in the ($k_p$, $k_i$)-plane. Then parameters of stabilizing PI controllers are determined. The designed robust PI controller is used for control of a laboratory chemical continuous stirred tank reactor. Armfield PCT40. The reactor is used for preparing of NaCl solution with desired concentration. The conductivity of the solution is the controlled variable and the volumetric flow rate of water is the control variable.

**Keywords:** interval uncertainty, PI controller, robust control, chemical reactor

## Introduction

Chemical reactors are ones of the most important plants in chemical industry, see e.g. [7]. Their operation, however, is corrupted with various uncertainties. Some of them arise from varying or not exactly known parameters, as e.g. reaction rate constants, reaction enthalpies or heat transfer coefficients. In other cases, operating points of reactors vary or reactor dynamics is affected by various changes of parameters of inlet streams. All these uncertainties can cause poor performance or even instability of closed-loop control systems. Application of robust control approach can be one of ways for overcoming all these problems, which may seriously influence control design for chemical reactors and other chemical processes, see e.g. [1], [6].

In this paper, a simple method for design of robust PI controllers is [8]. The method is based on plotting the stability boundary locus in the ($k_p$,$k_i$)-plane and then parameters of a stabilizing PI controller are determined from the stability region. The PI controller stabilizes a controlled system with interval parametric uncertainties, when the stability region is found for sufficient number of Kharitonov plants.

The approach is used for design of a robust PI controller for a laboratory continuous stirred tank reactor, which can be modeled in the form of a transfer function with parametric interval uncertainty. The reactor serves for preparing of the NaCl solution with required concentration. Composition of the solution is determined by measurement of the solution conductivity and the conductivity is the controlled variable. The volumetric flow rate of water which is used for adulterating of NaCl solution, is the control variable.

## 1. Description of the Laboratory Continuous stirred tank reactor

Multifunctional process control teaching system - The Armfield PCT40 ([2], [9]) is the system which enables to test a wide class of technological processes, as a tank, a heat exchanger, a continuous stirred tank reactor and their combinations ([3], [4]).

PCT40 unit consists of two process vessels, several pumps, sensors and connection to the computer. Additional equipments PCT41 and PCT42 (Figure 1) represent a chemical reactor with a stirrer and a cooling/heating coil.

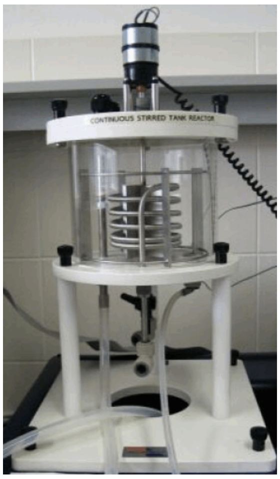


**Fig.1 Laboratory continuous stirred tank reactor**

Inlet streams of reactants can be injected into the reactor via a normally closed solenoid valve or by a proportional solenoid valve (PSV). The third possibility for feeding water into the reactor is using one of two peristaltic pumps. The technological parameters of the reactor are shown in Table 1.

| Parameter | Value |
|---|---|
| Vessel diameter | 0.153 m |
| Maximum vessel depth | 0.108 m |
| Maximum operation volume | 2 l |
| Minimum vessel depth | 0.054 m |
| Minimum operation volume | 1 l |

**Tab.1 Technological parameters of the reactor**

The connection to the computer is realized via an I/O connector, which is connected to the PCL card. The card used is the MF624 multifunction I/O card from Humusoft. This card has 8 inputs and 8 outputs. The whole system provides 9 inputs and 17 outputs, hence two MF624 cards were used. This connection enables use of Matlab Real-time Toolbox and Simulink or data entry from the Matlab command window.

NaCl solution with the concentration 0.8555 mol/dm$^3$ is fed into the reactor by a peristatic pump. The performance of the pump may be theoretically set in the range 0-100%. But for the pump performance less than 20%, revolutions of the rotor are very small and the produced force is not high enough to transport the fluid from the barrel. The volumetric flow rate of the NaCl solution for all measurements was 0.2222 dm$^3$/min, which represents the pump performance 30%.

The water was dosed into the reactor by the PSV. Application of the PSV allowed flow measurements by the adjoint flow meter. The PSV opening could be again done in the range 0-100%, but the volumetric flow rate of water for the PSV opening in the range 0-15% was negligible.

For control purposes, the laboratory continuous stirred tank reactor is a SISO system. The control variable is the volumetric flow rate of water (F) and the controlled variable is the conductivity of the NaCl solution (G) inside the reactor. Used water was cold water from the standard water distribution. The volume of the solution in the reactor was kept constant with the value 1 dm$^3$ during all experiments.

## 2. Process identification

Identification of the controlled laboratory reactor was done from measured step responses. The constant flow rate 0.2222 dm$^3$/min of NaCl solution dosed into the reactor was assured by the peristaltic pump with performance 30% in all experiments. Three various step changes of water flow rate were realized: 0-0.1804 dm$^3$/min, 0-1.3 dm$^3$/min and 0-1.78 dm$^3$/min which represented the PSV opening 0-20%, 0-50% and 0-100%. The step responses were measured repeatedly. The resultant transfer function of the laboratory reactor was identified ([5]) in the form of a transfer function (1) with the parametric interval uncertainty. The values of the uncertain parameters are shown in Table 2. Nominal values of the uncertain parameters are the mean values of intervals.

$$G(s)=\frac{b_0}{a_2 s^2+a_1 s+a_0} \tag{1}$$

| Parameter | Minimal value | Maximal value | Nominal value |
|---|---|---|---|
| $b_0$ | 0.00405 | 0.178 | 0.091 |
| $a_2$ | 130 | 500 | 315 |
| $a_1$ | 36.5 | 148 | 92.25 |
| $a_0$ | 1 | 1 | 1 |

**Tab. 2 Uncertain parameters**

The measured step response of the laboratory reactor and the simulated step response of the reactor with the identified transfer function (1) are compared in Figure 2 for the maximal step change of input variable.

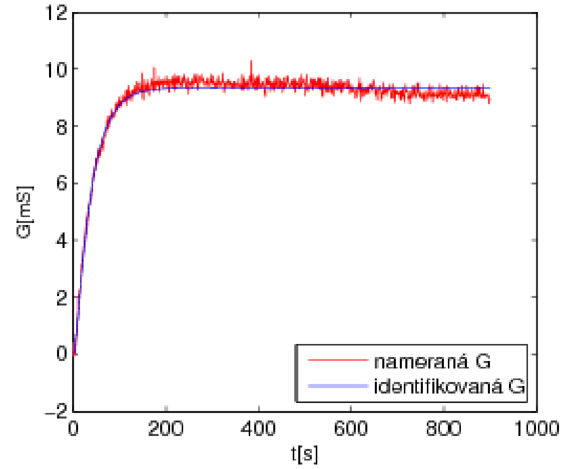


**Fig.1 Comparision of the measured and the simulated step responses of the reactor**

## 3. Design of a robust PI controller

A simple method based on plotting the stability boundary locus in the ($k_p$,$k_i$)-plane is used for robust PI controller design, [8], [10]. Parameters of a stabilizing PI controller are determined from the stability region of the ($k_p$,$k_i$) – plane. The PI controller stabilizes a controlled system with interval parametric uncertainties, when the stability region is found for sufficient number of Kharitonov plants.

For the controlled system in the form of the transfer function (1) with interval uncertainty (Table 2), the Kharitonov polynomials $N_i(s)$, i=1, 2 for the numerator and $D_j(s)$, j=1, 2, 3, 4 for the denominator can be created, as it is seen in (2), (3).

$$\begin{aligned} N_1(s)&=b_0^- \\ N_2(s)&=b_0^+ \end{aligned} \tag{2}$$

$$\begin{aligned} D_1(s)&=a_2^- + a_1^- + a_0 \\ D_2(s)&=a_2^+ + a_1^+ + a_0 \\ D_3(s)&=a_2^+ + a_1^- + a_0 \\ D_4(s)&=a_2^- + a_1^+ + a_0 \end{aligned} \tag{3}$$

where $b_0^-$ and $b_0^+$ are lower and upper bounds of the $b_0$ interval and $a_k^-$ and $a_k^+$, k= 1, 2, are lower and upper bounds of intervals of denominator parameters. 8 Kharitonov systems (4) can be obtained using polynomials (2), (3)

$$G_{ij}(s)=\frac{N_i(s)}{D_j(s)} \tag{4}$$

Substituting s=jω into (4) and decomposing the numerator and the denominator polynomials of (4) into their even and odd parts one obtains

$$G_{ij}(j\omega)=\frac{N_{ie}\left(-\omega^2\right)+j\omega N_{io}\left(-\omega^2\right)}{D_j e\left(-\omega^2\right)+j\omega D_{jo}\left(-\omega^2\right)} \tag{5}$$

The closed loop characteristic polynomial is as follows

$$\Delta(j\omega)=\left[k_i \mathrm{N}_{ie}\left(-\omega^2\right)-k_p\omega^2 N_{io}\left(-\omega^2\right)-\omega^2 D_{jo}\left(-\omega^2\right)\right] + j\left[k_p\omega N_{ie}\left(-\omega^2\right)+k_i\omega N_{io}\left(-\omega^2\right)+\omega D_{je}\left(-\omega^2\right)\right] \qquad (6)$$

Then, equating the real and imaginary parts of Δ(jω) to zero, one obtains

$$k_p\left(-\omega^2 N_{io}\left(-\omega^2\right)\right)+k_i\left(N_{ie}\left(-\omega^2\right)\right)=\omega^2 D_{jo}\left(-\omega^2\right) \qquad (7)$$

and

$$k_p\left(N_{ie}\left(-\omega^2\right)\right)+k_i\left(N_{io}\left(-\omega^2\right)\right)=-D_{je}\left(-\omega^2\right) \qquad (8)$$

After denoting

$$\begin{aligned} F_i(\omega)&=-\omega^2 N_{io}\left(-\omega^2\right)\\ G_i(\omega)&=N_{ie}\left(-\omega^2\right)\\ H_i(\omega)&=N_{ie}\left(-\omega^2\right)\\ I_i(\omega)&=N_{io}\left(-\omega^2\right)\\ J_j(\omega)&=\omega^2 D_{jo}\left(-\omega^2\right)\\ K_j(\omega)&=-D_{je}\left(-\omega^2\right) \end{aligned} \qquad (9)$$

(7), (8) and (9) can be written as

$$\begin{aligned} k_p F_i(\omega)+k_i G_i(\omega)&=J_j(\omega)\\ k_p H_i(\omega)+k_i I_i(\omega)&=K_j(\omega) \end{aligned} \qquad (10)$$

From these equations, parameters of the PI controller are expressed in the form

$$k_p=\frac{J_j(\omega)I_i(\omega)-K_j(\omega)G_i(\omega)}{F_i(\omega)I_i(\omega)-G_i(\omega)H_i(\omega)} \qquad (11)$$

and

$$k_i=\frac{K_i(\omega)F_i(\omega)-J_j(\omega)H_i(\omega)}{F_i(\omega)I_i(\omega)-G_i(\omega)H_i(\omega)} \qquad (12)$$

Consider one of the systems (4), where i=2 and j=2

$$G_{22}(s)=\frac{0.178}{500s^2+148s+1} \qquad (13)$$

The closed loop characteristic polynomial has according to (6) the form

$$\Delta(j\omega)=\left[-a_1^+\omega^2+b_0^+k_i\right]+j\left[-a_2^+\omega^3+b_0^+k_p\omega+\omega\right] \qquad (14)$$

and then

$$\begin{aligned} k_p&=2890\ \omega^2-5.6180\\ k_i&=831\ .4607\ \omega^2 \end{aligned} \qquad (15)$$

The stability boundary of the closed loop characteristic polynomial in the $(k_p,k_i)$-plane for ω=[0:0.001:0.5] is plot in the Figure 3. It is seen in the figure that the region is split in two parts, stable and unstable ones. Parameters $k_p$ and $k_i$ of the stabilizing controller are chosen from the stable region.

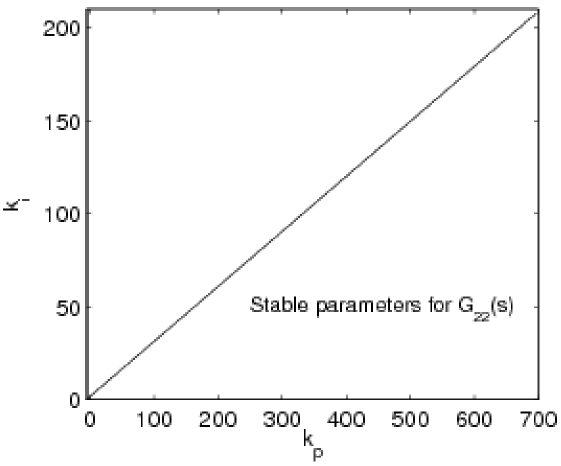


**Fig.3 Stability region of parameters $k_p$, $k_i$ for the system $G_{22}$(s)**

Stable regions for all 8 Kharitonov systems are obtained alike. In the Figure 4 are shown stable regions for 8 Kharitonov systems (4). The controller which stabilizes all 8 Kharitonov systems has to be found in the intersection of all stable regions, which is in detail displayed in the Figure 5.

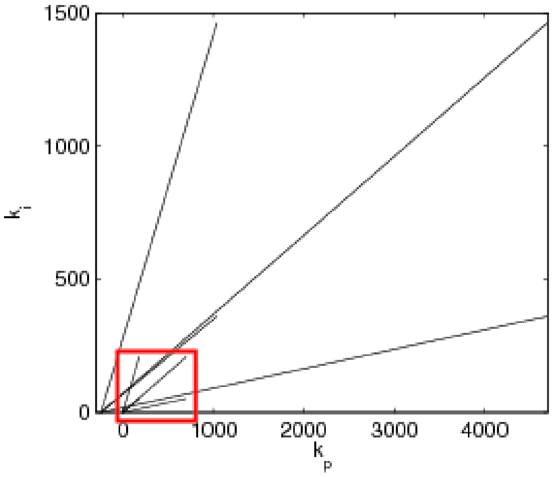


**Fig.4 Stability regions for 8 Kharitonov plants**

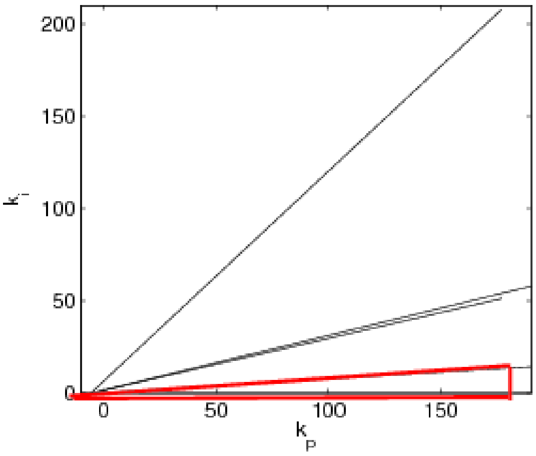


**Fig.5 Detail of the stability region for 8 Kharitonov plants**

The parameters of the robust PI controller for control of the laboratory reactor (16) were chosen from the stable region of parameters $k_p$, $k_i$ according to simulation results obtained for various choices of PI controllers.

$$C(s) = \frac{k_p s + k_i}{s} = \frac{110s + 12}{s} \qquad (16)$$

The designed PI controller was used for control of the laboratory reactor. The controlled variable y(t) was the conductivity G [mS] of the NaCl solution, control variable u(t) was the water flow rate F [dm$^3$/min] and the reference w(t) was the conductivity of the NaCl solution which corresponded to the required concentration of the NaCl solution.

Obtained experimental results are presented in the Figures 6 and 7. Robustness of the designed PI controller (16) was tested by setting the reference value in a wider area. In the Figure 6 are control responses of the reactor for ω є[8; 16] mS and in the Figure 7 for ω є[11; 21] mS.

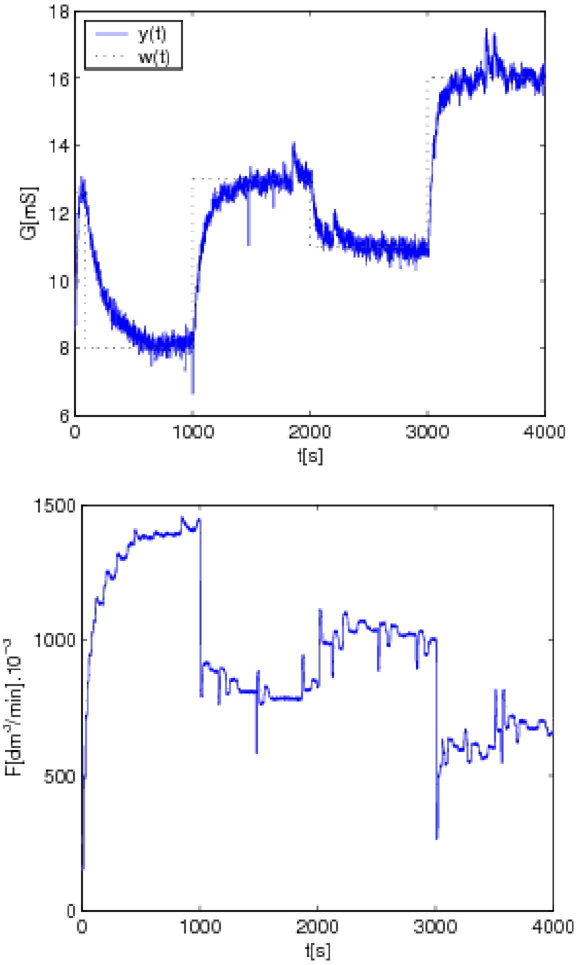


**Fig.6 Control of the reactor with robust PI controller**

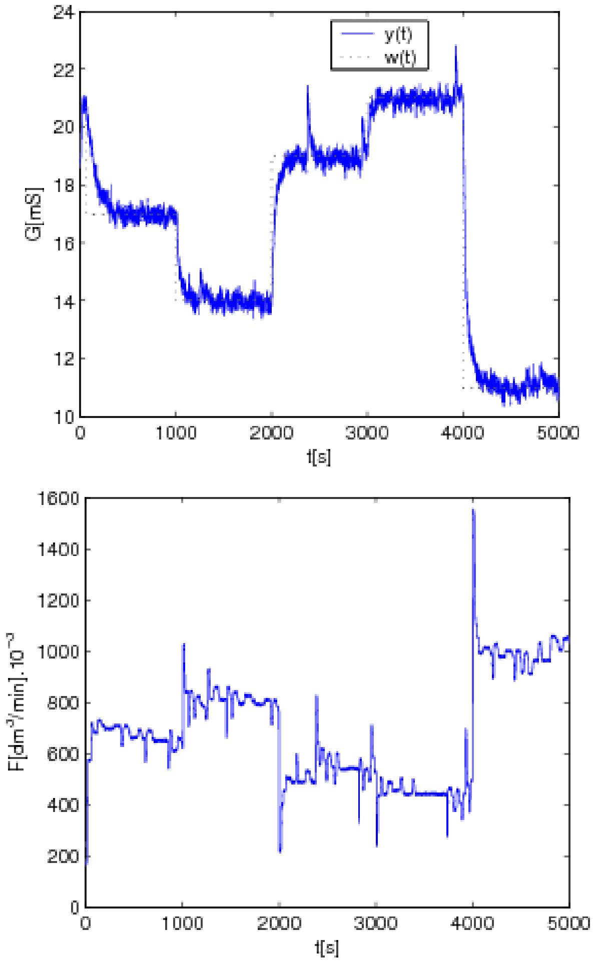


**Fig.7 Control of the reactor with robust PI controller**

## Conclusion

The robust PI controller was designed for control of the laboratory continuous stirred tank reactor. A simple robust synthesis was used which was based on plotting the stability boundary locus in the ($k_p$,$k_i$)-plane. The stabilizing PI controller was chosen from the stable region of the ($k_p$,$k_i$)-plane. The designed controller was tested experimentally by control of a laboratory reactor. Obtained experimental results confirm that the designed robust PI controller successfully controlled the laboratory reactor. The varying reference was always reached. The control responses were without overshoots and fast enough. The future work will be focused on improvement the choice of a stabilizing controller so that also the quality of control will be assured.

### Acknowledgements

*The work has been supported by the Scientific Grant Agency of the Slovak Republic under grants 1/4055/07, 1/0071/09 and by the Slovak Research and Development Agency under the project APVV-0029-07.*

## References

[1] ALVAREZ-RAMIREZ, J., FEMAT, R.: Robust PI stabilization of a class of chemical reactors. System Control Letter, 38, 219-225, 1999.

[2] ARMFIELD: Instruction manual PCT40, 4 edition, 2005.

[3] ARMFIELD: Instruction manual PCT41, 3 edition, 2006a.

[4] ARMFIELD: Instruction manual PCT42, 2 edition, 2006b.

[5] ČIRKA, Ľ., FIKAR, M.: A Dynamical System Identification Toolbox, Selected Topics in Modelling and Control. STU Press, 58-62, Bratislava, 2007.

[6] GERHARD, J., MONNINGMANN, M., MARQUARDT, W: Robust stable nonlinear control and design of a CSTR in a large operating range. In Proc. 7th Int. Symp Dynamics and Control of Process Systems, pages Cambridge, Massachusset, USA, July 5-7. CD-ROM 92, 2004.

[7] MIKLEŠ, J., FIKAR, M: Process modelling, identification and control II. STU Press, Bratislava, 2004.

[8] TAN, N., KAYA, I.: Computation of stabilizing PI controllers for interval systems. In Meditteranean Conference on Control and Automation, Rhodes, Greece, June 2003.

[9] VOJTEŠEK, J., DOSTAL, J., MATUŠŮ: Multifunctional process control teaching system pct40. In 16th Int. Conferences Process Control, page 15f.pdf, Štrbské Pleso, Slovakia, 2007.

[10] ZÁVACKÁ, J., BAKOŠOVÁ, M., VANEKOVÁ, K.: Design of robust PI, PID controller for control of systems with parametric uncertainty. In V 18th International Congress of Chemical and Process Engineering, pages 0457-1-0457-10, Prague, Czech Republic, June 2008.

**Ing. Jana Závacká,**
**Doc. Ing. Monika Bakošová, CSc.**
**Ing. Katarína Vaneková**
Slovak University of Technology in Bratislava
Faculty of Chemical and Food Technology
Department of Information Engineering and Process Control
Radlinského 9
812 37 Bratislava
Tel.: (02)59325 349
E-mail: {zavacka, bakosova, vanekova}@stuba.sk

# Global Asymptotically Stable Control Design for Time-delay Systems

**Anna Filasová, Dušan Krokavec**

**Abstract**

The purpose of this paper is to present an improved version of time-delay system state feedback control methods and any extension over the one concerning the output and input variables constraint. Based on the standard Lyapunov-Krasovskii functional and norm-bounded constraints, delayed--independent stability condition is derived using linear matrix inequalities. The results obtained with a numerical example are presented to compare limitation in system structure for defined onstraints. Since presented method is based on convex optimization techniques it is computationally very efficient.

**Keywords:** Constraints, linear matrix inequality, state feedback, time-delay systems, asymptotic stability

## Introduction

The continuous-time control systems are used in many industrial applications, where time delays can take a deleterious effect on both the stability and the dynamic performance in open and closed-loop systems. The presence of delays (especially, long delays) makes system annalysis and control design much more complex, the system is very likely to overshoot or even became unstable. Therefore the stability and control of dynamical systems involving time-delayed states is a problem of very large theoretical and practical interest where the intensive activities are done to eliminate fixed time delays, to compensate for the uncertain ones or to develop the control for time-delay systems stabilization, especially for the uncertain time-delay systems.

Number of techniques for the time-delay linear systems control design as well as for the stability analysis have been reported in the literature over past decades. Usually for the stability issue of the time delay systems the Lyapunov-Krasovskii functional is used and the results based on this functional are applied to the controller synthesis and to the observer design. This time-delay independent methodology, as well as used bounded inequality techniques are sources of conservatism that can cause higher norm of the state feedback gain (see e.g. [20]). Some progres review in this research field one can find in [5], [12], [21], and the references therein.

This paper is concerned with the problem of the asymptotically stable control design of the continuous-time linear systems with delayed state, where the case of single, possibly varying time delay is considered and attention is focused on the methods based on the linear matrix inequalities (LMIs). As by-products, new formulae for the input/output constrained state feedback control of the system with the uncontroled time-delayed stay vectors is derived. The presented LMI approach is computationally efficient as it can be solved numerically using the interior point methods (see e.g. [11], [17], [18]) in LMI formulation, and is based on the norm bounded approximation for the Lyapunov-Krasovskii functional (see e.g. [3], [7], [21]), as well for the new defined constrained extension of this functional. Numerical examples are presented to compare limitation in system structure for defined onstraints

## 1. Problem Description

Through this paper the task is concerned with the computation of the state feedback $\boldsymbol{u}(t)$ which control a time-delay linear dynamic system given by the set of equations

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{A}_2\boldsymbol{q}(t-\tau) + \boldsymbol{B}\boldsymbol{u}(t) \qquad (1)$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{q}(t) \qquad (2)$$

with initial condition

$$\boldsymbol{q}(\vartheta) = \varphi(\vartheta), \qquad \vee\,\vartheta \in \langle -\tau, 0\rangle \qquad (3)$$

where $\tau > 0$ is the delay, $\boldsymbol{q}(t) \in \mathbf{R}^n$, $\boldsymbol{u}(t) \in \mathbf{R}^r$ and $\boldsymbol{y}(t) \in \mathbf{R}^m$ are vectors of the state, input and measurable output variables, respectively, and the nominal system matrices $\boldsymbol{A} \in \mathbf{R}^{n\times n}$, $\boldsymbol{A}_2 \in \mathbf{R}^{n\times n}$, $\boldsymbol{B} \in \mathbf{R}^{n\times r}$ and $\boldsymbol{C} \in \mathbf{R}^{m\times n}$ are real matrices. Problem of the interest is to design the asymptotically stable closed-loop system with the memory less linear state feedback controller of the form

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{q}(t) \qquad (4)$$

Here matrix $\boldsymbol{K} \in \mathbf{R}^{r\times n}$ is the controller gain matrix. It is supposed that all eigenvalues of the matrix

$$\boldsymbol{A}_{c2} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}) + \boldsymbol{A}_2 \qquad (5)$$

lie in the open left-half plane. The above assumption, which corresponds to the asymptotic stability of the closed-loop system without time delay, is indeed necessary for the asymptotic stability of closed-loop system with time delays.

## 2. Basic Preliminaries

### 2.1 Schur Complement

The nonlinear convex inequalities can be converted to LMI form using Schur complements. Let a linear matrix inequality takes form

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & -\boldsymbol{R} \end{bmatrix} < 0, \quad \boldsymbol{Q} = \boldsymbol{Q}^T > 0, \, \boldsymbol{R} = \boldsymbol{R}^T > 0 \tag{6}$$

Using Gauss elimination it yields

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}\boldsymbol{R}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & -\boldsymbol{R} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{R}^{-1}\boldsymbol{S}^T & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q} + \boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}^T & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{R} \end{bmatrix} \tag{7}$$

Since

$$\det \begin{bmatrix} \boldsymbol{I} & \boldsymbol{S}\boldsymbol{R}^{-1} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = 1 \tag{8}$$

and $\boldsymbol{I}$ is the identity matrix of appropriate dimension, with this transform the negativity of (7) is not changed, i.e. this follows as a consequence

$$\begin{gathered} \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{S} \\ \boldsymbol{S}^T & -\boldsymbol{R} \end{bmatrix} < 0 \Leftrightarrow \begin{bmatrix} \boldsymbol{Q} + \boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}^T & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{R} \end{bmatrix} < 0 \\ \Updownarrow \\ \boldsymbol{Q} + \boldsymbol{S}\boldsymbol{R}^{-1}\boldsymbol{S}^T < 0, \boldsymbol{R} > 0 \end{gathered} \tag{9}$$

respectively. As one can see, this complement offer possibility to rewrite the nonlinear inequalities in a closed matrix LMI form (see e.g. [1], [8]).

### 2.2 Symmetric upper-bounds inequality

Let $\boldsymbol{p} \in \mathbf{R}^n$, $\boldsymbol{r} \in \mathbf{R}^n$, are vectors of equal dimension. Then the next equality is satisfied

$$-\boldsymbol{p}^T\boldsymbol{r} - \boldsymbol{r}^T\boldsymbol{p} \le \boldsymbol{p}^T\boldsymbol{S}^{-1}\boldsymbol{p} + \boldsymbol{r}^T\boldsymbol{S}\boldsymbol{r} \tag{10}$$

where $\boldsymbol{S} = \boldsymbol{S}^T > 0$, $\boldsymbol{S} \in \mathbf{R}^{n \times n}$, is any symmetric positive definite matrix. (see e.g. [9], [10]).

## 3. Time-delay System Control with Output Constraint

Defining the Lyapunov--Krasovskii functional with output constraint as follows

$$\mathrm{v}(\boldsymbol{q}(t)) = \boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{q}(t) + \int_{t-\tau}^{t} (\boldsymbol{q}^T(r)\boldsymbol{R}\boldsymbol{q}(r) + \varepsilon\boldsymbol{y}^T(t)\boldsymbol{y}(t))\,\mathrm{d}r \tag{11}$$

where $\boldsymbol{P} = \boldsymbol{P}^T > 0$, $\boldsymbol{R} = \boldsymbol{R}^T > 0$, $0 \le \varepsilon < 1$, and evaluating derivative of $\mathrm{v}(\boldsymbol{q}(t))$ gives

$$\begin{gathered} \dot{\mathrm{v}}(\boldsymbol{q}(t)) = \dot{\boldsymbol{q}}^T(t)\boldsymbol{P}\boldsymbol{q}(t) + \boldsymbol{q}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t) + \\ +(\boldsymbol{q}^T(r)\boldsymbol{R}\boldsymbol{q}(r) + \varepsilon\boldsymbol{y}^T(t)\boldsymbol{y}(t))\Big|_{t-\tau}^{t} < 0 \end{gathered} \tag{12}$$

$$\begin{gathered} \dot{\mathrm{v}}(\boldsymbol{q}(t)) = (\boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{A}_2\boldsymbol{q}(t-\tau) + \boldsymbol{B}\boldsymbol{u}(t))^T\boldsymbol{P}\boldsymbol{q}(t) + \\ +\boldsymbol{q}^T(t)\boldsymbol{P}(\boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{A}_2\boldsymbol{q}(t-\tau) + \boldsymbol{B}\boldsymbol{u}(t)) + \\ +\boldsymbol{q}^T(t)(\boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C})\boldsymbol{q}(t) - \boldsymbol{q}^T(t-\tau)(\boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C})\boldsymbol{q}(t-\tau) < 0 \end{gathered} \tag{13}$$

respectively. Using inequality (10) it can be written

$$\begin{gathered} \boldsymbol{q}^T(t-\tau)\boldsymbol{A}_2^T\boldsymbol{P}\boldsymbol{q}(t) + \boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{A}_2\boldsymbol{q}(t-\tau) \le \\ \le \boldsymbol{q}^T(t-\tau)\boldsymbol{S}\boldsymbol{q}(t-\tau) + \boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2^T\boldsymbol{P}\boldsymbol{q}(t) \end{gathered} \tag{14}$$

and considering (14) it is possible to rewrite (13) in the form

$$\begin{gathered} \dot{\mathrm{v}}(\boldsymbol{q}(t)) \le \boldsymbol{u}^T(t)\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{q}(t) + \boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{B}\boldsymbol{u}(t) + \\ +\boldsymbol{q}^T(t)(\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{P}\boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2^T\boldsymbol{P} + \boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C})\boldsymbol{q}(t) + \\ +\boldsymbol{q}^T(t-\tau)(\boldsymbol{S} - \boldsymbol{R} - \varepsilon\boldsymbol{C}^T\boldsymbol{C})\boldsymbol{q}(t-\tau) < 0 \end{gathered} \tag{15}$$

Then, with (4), inequality (15) implies

$$\begin{gathered} \boldsymbol{q}^T(t)(\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{P}\boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2\boldsymbol{P} + \boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C} - \boldsymbol{K}^T\boldsymbol{B}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{K})\boldsymbol{q}(t) + \\ +\boldsymbol{q}^T(t-\tau)(\boldsymbol{S} - \boldsymbol{R} - \varepsilon\boldsymbol{C}^T\boldsymbol{C})\boldsymbol{q}(t-\tau) < 0 \end{gathered} \tag{16}$$

or, in the matrix form

$$\begin{bmatrix} \boldsymbol{q}^T(t) & \boldsymbol{q}^T(t-\tau) \end{bmatrix} \begin{bmatrix} \Phi_1 & 0 \\ 0 & \Phi_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{q}(t) \\ \boldsymbol{q}(t-\tau) \end{bmatrix} < 0 \tag{17}$$

where

$$\Phi_1 = \boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{P}\boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2\boldsymbol{P} + \boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C} - \boldsymbol{K}^T\boldsymbol{B}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{K} < 0 \tag{18}$$

$$\Phi_2 = \boldsymbol{S} - (\boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C}) < 0 \tag{19}$$

Since $\boldsymbol{P} = \boldsymbol{P}^T > 0$, pre-multiplying (18) from the left side and the right side by $\boldsymbol{P}^{-1} > 0$ leads to the next inequality

$$\begin{gathered} \Psi_1 = \boldsymbol{P}^{-1}\Phi_1\boldsymbol{P}^{-1} = \boldsymbol{P}^{-1}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{P}^{-1} + \boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2 + \\ +\boldsymbol{P}^{-1}(\boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C})\boldsymbol{P}^{-1} - \boldsymbol{P}^{-1}\boldsymbol{K}^T\boldsymbol{B}^T - \boldsymbol{B}\boldsymbol{K}\boldsymbol{P}^{-1} < 0 \end{gathered} \tag{20}$$

If both design parameters are prescribed and fixed and $\boldsymbol{R} = \boldsymbol{R}^T > 0$, $0 < \varepsilon < 1$, then the design condition takes form of LMIs

$$\begin{bmatrix} \boldsymbol{Y}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{Y} + \boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2^T - \boldsymbol{B}\boldsymbol{Z} - \boldsymbol{Z}^T\boldsymbol{B}^T & \boldsymbol{Y} \\ * & -\boldsymbol{X} \end{bmatrix} < 0 \tag{21}$$

$$\begin{bmatrix} \boldsymbol{S} & \boldsymbol{I} \\ * & \boldsymbol{X} \end{bmatrix} < 0 \tag{22}$$

with the LMI variables

$$\boldsymbol{Y} = \boldsymbol{P}^{-1} > 0, \qquad \boldsymbol{Z} = \boldsymbol{K}\boldsymbol{P}^{-1} = \boldsymbol{K}\boldsymbol{Y} \tag{23}$$

and with the design parameter

$$\boldsymbol{X} = (\boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C})^{-1} \tag{24}$$

If design parameters are free, it is possible to be chosen

$$\boldsymbol{X}^{-1} = \boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C} = (1+\eta)\boldsymbol{I} = \delta^{-1}\boldsymbol{I}, \quad 0 < \delta < 1 \tag{25}$$

Then the structures of (21) and (22) are

$$\begin{bmatrix} \boldsymbol{Y}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{Y} + \boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2^T - \boldsymbol{B}\boldsymbol{Z} - \boldsymbol{Z}^T\boldsymbol{B}^T & \boldsymbol{Y} \\ * & -\delta\boldsymbol{I} \end{bmatrix} < 0 \tag{26}$$

$$\begin{bmatrix} \boldsymbol{S} & \boldsymbol{I} \\ * & \delta\boldsymbol{I} \end{bmatrix} < 0 \tag{27}$$

Especially, if it is possible to be chosen $\eta = 0$, i.e. $\delta = 1$, and $\boldsymbol{S} = \boldsymbol{I}$, in limit can be obtained $\Phi_2 = \boldsymbol{0}$, and

$$\begin{bmatrix} \boldsymbol{Y}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{Y} + \boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2^T - \boldsymbol{B}\boldsymbol{Z} - \boldsymbol{Z}^T\boldsymbol{B}^T & \boldsymbol{Y} \\ * & -\boldsymbol{I} \end{bmatrix} < 0 \tag{28}$$

Condition (28) with $\delta = 1$ and $\boldsymbol{S} = \boldsymbol{I}$ set this design task to be independent on the system state time delay.

Solving for $\boldsymbol{Y}$ and $\boldsymbol{Z}$, the gain matrix of the state feedback control law for the all above mentioned modifications can be designed as

$$\boldsymbol{K} = \boldsymbol{Z}\boldsymbol{P} = \boldsymbol{Z}\boldsymbol{Y}^{-1} \tag{29}$$

To compute $\boldsymbol{Y}$ and $\boldsymbol{Z}$ any LMI solver is necessary to use.

## 4. Time-delay System Control with Output and Input Constraint

Generally it is possible to extend constraint in the Lyapunov--Krasovskii functional (11) as follows

$$\mathrm{v}_e(\boldsymbol{q}(t)) = \mathrm{v}(\boldsymbol{q}(t)) + \int_{t-\tau}^{t} \gamma \boldsymbol{u}^T(t)\boldsymbol{u}(t)\,\mathrm{d}\,r \tag{30}$$

where $\mathrm{v}(\boldsymbol{q}(t))$ is defined in (11) with $\varepsilon = 1,$ and $\gamma > 0$ .

Then time derivative of Lyapunov--Krasovskii functional (30) takes form

$$\dot{\mathrm{v}}_e(\boldsymbol{q}(t)) = \dot{\mathrm{v}}(\boldsymbol{q}(t)) + \gamma \boldsymbol{u}^T(t)\boldsymbol{u}(t) - \gamma \boldsymbol{u}^T(t-\tau)\boldsymbol{u}(t-\tau) < 0 \tag{31}$$

where $\dot{\mathrm{v}}(\boldsymbol{q}(t))$ is given in (15). Then, with (4) and (15) inequality (31) implies

$$\begin{bmatrix} \boldsymbol{q}^T(t) & \boldsymbol{q}^T(t-\tau) \end{bmatrix} \begin{bmatrix} \Phi_{1e} & 0 \\ 0 & \Phi_{2e} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}(t) \\ \boldsymbol{q}(t-\tau) \end{bmatrix} < 0 \tag{32}$$

$$\begin{aligned} \Phi_{1e} = {} & \boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{P}\boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2\boldsymbol{P} + \boldsymbol{R} + \boldsymbol{C}^T\boldsymbol{C} - \\ & -\boldsymbol{K}^T\boldsymbol{B}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{K} + \gamma\boldsymbol{K}^T\boldsymbol{K} < 0 \end{aligned} \tag{33}$$

$$\Phi_{2e} = \boldsymbol{S} - (\boldsymbol{R} + \boldsymbol{C}^T\boldsymbol{C}) - \gamma\boldsymbol{K}^T\boldsymbol{K} < 0 \tag{34}$$

Therefore, using symmetric properties of ***P*** one can write for $\Psi_{1e}, \Psi_{2e}$

$$\begin{aligned} \Psi_{1e} = {} & \boldsymbol{P}^{-1}\Phi_{1e}\boldsymbol{P}^{-1} = \boldsymbol{P}^{-1}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{P}^{-1} + \boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2 - \boldsymbol{P}^{-1}\boldsymbol{K}^T\boldsymbol{B}^T + \\ & + \boldsymbol{P}^{-1}(\boldsymbol{R} + \boldsymbol{C}^T\boldsymbol{C})\boldsymbol{P}^{-1} + \gamma\boldsymbol{P}^{-1}\boldsymbol{K}^T\boldsymbol{K}\boldsymbol{P}^{-1} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{P}^{-1} < 0 \end{aligned} \tag{35}$$

$$\Psi_{2e} = \boldsymbol{P}^{-1}\Phi_{2e}\boldsymbol{P}^{-1} = \boldsymbol{P}^{-1}(\boldsymbol{S} - (\boldsymbol{R} + \boldsymbol{C}^T\boldsymbol{C}) - \gamma\boldsymbol{K}^T\boldsymbol{K})\boldsymbol{P}^{-1} < 0 \tag{36}$$

and using (23), (24), to do it as follows

$$\begin{bmatrix} \boldsymbol{Y}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{Y} + \boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2^T - \boldsymbol{B}\boldsymbol{Z} - \boldsymbol{Z}^T\boldsymbol{B}^T & \boldsymbol{Y} & \gamma\boldsymbol{Z}^T \\ * & -\boldsymbol{X} & \boldsymbol{0} \\ * & * & \gamma\boldsymbol{I}_m \end{bmatrix} < 0 \tag{37}$$

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{Y} & \boldsymbol{Y} & \gamma\boldsymbol{Z} \\ * & -\boldsymbol{S}^{-1} & \boldsymbol{0} & \boldsymbol{0} \\ * & * & \boldsymbol{X} & \boldsymbol{0} \\ * & * & * & \gamma\boldsymbol{I} \end{bmatrix} < 0 \tag{38}$$

$$\boldsymbol{Y} = \boldsymbol{P}^{-1} > 0, \qquad \boldsymbol{Z} = \boldsymbol{K}\boldsymbol{P}^{-1} = \boldsymbol{K}\boldsymbol{Y}, \qquad \boldsymbol{X} = (\boldsymbol{R} + \boldsymbol{C}^T\boldsymbol{C})^{-1} \tag{39}$$

If design parameters are free, it is possible to choose $\boldsymbol{X}$ as in (25), i.e.

$$\boldsymbol{X}^{-1} = \boldsymbol{R} + \boldsymbol{C}^T\boldsymbol{C} = (1+\eta)\boldsymbol{I} = \delta^{-1}\boldsymbol{I}, \quad 0 < \delta < 1 \tag{40}$$

Then the structures of matrix inequalities (37) and (38) take forms

$$\begin{bmatrix} \boldsymbol{Y}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{Y} + \boldsymbol{A}_2\boldsymbol{S}^{-1}\boldsymbol{A}_2^T - \boldsymbol{B}\boldsymbol{Z} - \boldsymbol{Z}^T\boldsymbol{B}^T & \boldsymbol{Y} & \gamma\boldsymbol{Z}^T \\ * & -\delta\boldsymbol{I} & \boldsymbol{0} \\ * & * & \gamma\boldsymbol{I}_m \end{bmatrix} < 0 \tag{41}$$

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{Y} & \boldsymbol{Y} & \gamma\boldsymbol{Z} \\ * & -\boldsymbol{S}^{-1} & \boldsymbol{0} & \boldsymbol{0} \\ * & * & \delta\boldsymbol{I} & \boldsymbol{0} \\ * & * & * & \gamma\boldsymbol{I} \end{bmatrix} < 0 \tag{42}$$

Solving for ***Y*** and ***Z***, the gain matrix of the state feedback control law for these modifications can be designed using (29), too.

## 5. Illustrative example

The numerical example is provided below to illustrate the main results. It is assumed that the parameters of the delay system (1), (2) are given by

$$\boldsymbol{A} = \begin{bmatrix} -7.36 & -2.76 & -13.80 \\ 19.58 & 8.96 & 31.80 \\ -5.68 & -3.88 & -5.40 \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} -2.88 & -0.96 & -0.96 \\ 9.48 & 3.16 & 3.16 \\ -4.44 & -1.48 & -1.48 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 0.4 & -0.6 \\ -0.4 & 2.6 \\ 0.2 & 0.2 \end{bmatrix}, \qquad \boldsymbol{C} = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 1 & 0 \end{bmatrix}$$

and the design parameters of Lyapunov--Krasovskii functional with output constraint (11) satisfy equalities

$$\boldsymbol{R} + \varepsilon\boldsymbol{C}^T\boldsymbol{C} = \delta^{-1}\boldsymbol{I}, \quad 0 < \delta < 1, \quad \boldsymbol{S} = \boldsymbol{I}$$

Solving (26), (27) for LMI matrix variables (23) using the Self-Dual-Minimization (SeDuMi) package for Matlab [13]), the feedback gain matrix design problem was solved as feasible with these matrices

$$\boldsymbol{Y} = \begin{bmatrix} 3.1236 & -1.8626 & 2.9324 \\ -1.8626 & 5.9776 & 2.1982 \\ 2.9324 & 2.1928 & 6.3620 \end{bmatrix}$$

$$\boldsymbol{Z} = \begin{bmatrix} 14.0738 & 11.5758 & -92.5781 \\ 43.9268 & 72.1630 & 72.1595 \end{bmatrix}$$

Substituting for $\boldsymbol{Y}$ and $\boldsymbol{Z}$ into (29) there was computed the feedback gain matrix as follows

$$\boldsymbol{K} = \begin{bmatrix} 437.3844 & 249.4492 & -302.4663 \\ 198.5989 & 118.5257 & -121.2061 \end{bmatrix}$$

It is possible easily to verify, that the closed loop is stable, with the system matrices satisfying given stability condition for

$$\boldsymbol{A}_c = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K} = \begin{bmatrix} -63.1544 & -31.4243 & 34.4629 \\ -321.8434 & -199.4270 & 225.9493 \\ -132.8767 & -77.4750 & 79.3345 \end{bmatrix}$$

$$\rho(\boldsymbol{A}_c) = \{-8.5632 \ \ -13.9219 \ \ -160.7619\}$$

$$\rho(\boldsymbol{A}_c + \boldsymbol{A}_2) = \{-8.5572 \ \ -23.7091 \ \ -152.1807\}$$

where $\rho(\cdot)$ denotes the eigenvalue spectrum of any square matrix.

Using Lyapunov--Krasovskii functional (30), with the output and the input variable constraint parameters setting as $\varepsilon = 1,$ and $\gamma = 0.1$ there were no feasible solutions for LMI matrix variables $\boldsymbol{Y}$ and $\boldsymbol{Z}$ if matrix $\boldsymbol{A}_2$ was specified as given above. Taking in computation another time-delay states system matrix $\boldsymbol{A}_2$ chosen as follows

$$\boldsymbol{A}_2^{\bullet} = 0.2\boldsymbol{A}_2$$

the problem was feasible with these results

$$\boldsymbol{Y} = \begin{bmatrix} 0.5303 & -0.8413 & 0.0603 \\ -0.8413 & 1.6117 & 0.0815 \\ 0.0603 & 0.0815 & 0.2798 \end{bmatrix}$$

$$\boldsymbol{Z} = \begin{bmatrix} 0.4340 & 0.1380 & -1.0476 \\ 1.1634 & 2.6166 & 2.5129 \end{bmatrix}$$

$$\boldsymbol{K} = \begin{bmatrix} 17.0074 & 9.4774 & -10.1687 \\ 31.0407 & 17.9748 & -2.9407 \end{bmatrix}$$

$$\boldsymbol{A}_c = \begin{bmatrix} 4.4615 & 4.2339 & -11.4669 \\ -54.3430 & -33.9835 & 35.3782 \\ -15.2896 & -9.3704 & -2.7781 \end{bmatrix}$$

respectively. Since both eigenvalue spectrum of the system matrices

$$\rho(\boldsymbol{A}_c) = \{-5.1736 \;\; -10.7180 \;\; -16.4086\}$$

$$\rho(\boldsymbol{A}_c + \boldsymbol{A}_2) = \{-4.0292 \;\; -14.2555 \pm \mathrm{j}\, 3.7255\}$$

lie in the open left-half plain, designed control is stable.

## 5. Concluding Remarks

In this paper there was developed a constructive method based on the classical memory-less feedback control for the stabilization of the time-delay systems with constraints given on the output and the input variables. The method ensures that the closed-loop system is internally stable in the sense of the global uniform asymptotic stability in the presence of a state time delay. The validity of the proposed method is demonstrated by the numerical examples with the asymptotically stable closed-loop system variables.

### Acknowledgments

*The work presented in this paper was supported by Grant Agency of Ministry of Education and Academy of Science of Slovak Republic VEGA under Grant No. 1/0328/08. This support is very gratefully acknowledged.*

## References

[1] BOYD, D., EL GHAOUI, L., PERON, E., BALAKRISHNAN, V.: *Linear Matrix Inequalities in System and Control Theory*. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[2] DORÁK, S., ROSINOVÁ, D.: Decentralized control design using LMI model reduction. *Journal of Electrical Engineering*, 58:6, 307-312, 2007.

[3] FRIDMAN, E.: New Lyapunov-Krasovskii functionals for stability of linear retarded and neutral type systems. *Systems and Control Letters*, **43**:4, 309-319, 2001.

[4] GAHINET, P., NEMIROVSKI, A., LAUB, A.J., CHILALI, M.: *LMI Control Toolbox User's Guide*. The MathWorks, Inc., Natick, 1995.

[5] GU, K., KHARITONOV, V.L., CHEN, J.: *Stability of Time-Delay Systems*. Birkhäuser, Boston, 2003.

[6] HALÁSZ, M., HUBA, M., ŽILKA, V.: Constrained discrete-time nonlinear controller for a fluid tank system. *AT&P Journal Plus*, 2, 2007.

[7] KOLMANOVSKII, V.B., NICULESCU, S., RICHARD, J.: On the Lyapunov Krasovskii functionals for stability analysis of linear delay systems. *International Journal of Control*, **72**, 374-384, 1999.

[8] KROKAVEC D., FILASOVÁ, A.: *Dynamic Systems Diagnosis*. Elfa, Košice, 2007. (in Slovak)

[9] KROKAVEC D., FILASOVÁ, A.: *Discrete-Time Systems*. Elfa, Košice, 2008. (in Slovak)

[10] LI, X., de SOUZA, C.E.: Delay-dependent robust stability and stabilization of uncertain time-delayed systems: A linear matrix inequality approach. *IEEE Transactions on Automatic Control*, **42**, 1144-1148, 1997.

[11] NESTEROV, Y., NEMIROVSKY, A.: *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*. SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1994.

[12] NICULESCU, S.I., VERIEST, E.I., DUGARD, L., DION, J.M.: Stability and robust stability of time-delay systems: A guided tour. In *Stability and Control of Time-delay Systems*. Springer-Verlag, Berlin, 1998.

[13] PEAUCELLE, D., HENRION, D., LABIT, Y., TAITZ, K.: *User's Guide for SeDuMi Interface 1.04*. LAAS-CNRS, Toulouse, 2002.

[14] PROKOP, R., DOSTÁL, P., BAKOŠOVÁ, M.: Control of continuous stirred tank reactor based on delta model representation. *Hungarian Journal of Industrial Chemistry*, **23**:4, 263-269, 1995.

[15] SÁS, M., FILASOVÁ, A., KLACIK, J.: K-nearest neighbor algorithm in multi-model switching control. Annals of the University of Craiova, **2(29)**:1, 104-108, 2005.

[16] SHAKED, U., YAESH, I., de SOUZA, C.E.: Bounded real criteria for linear time systems with state-delay. *IEEE Transactions on Automatic Control*, **43**, 1116-1121, 1998.

[17] SKELTON, R.R., IVASAKI, T., GRIGORIADIS, K.: A *Unified Algebraic Approach to Linear Control Design*. Taylor & Francis, London, 1998.

[18] VESELÝ, V., KOZÁKOVÁ, A., PAPADOPOULOS, D.: Design of robust output feedback controller via LMI approach. Journal of Electrical Engineering, 52:9-10, 273-277, 2001.

[19] VÍTEČKOVÁ, M., VÍTEČEK A.: Modulus optimum for digital controllers. Acta Montanistica Slovaka. **8**:4, 214-216, 2003.

[20] WANG, D.J.: A new approach to delay-dependent $H_{\mathrm{inf}}$ control of time-delayed system. *ASME Journal of Dynamic Systems, Measurements, and Control*, **126**, 201-204, 2004.

[21] ZHONG, Q.C.: *Robust Control of Time-delay Systems*. Springer- Verlag, London, 2009.

**Anna Filasová, Prof. Assoc. Ing. Ph.D.**

**Dušan Krokavec, Prof. Ing. Ph.D.**

Technical University of Košice
Faculty of Electrical Engineering and Informatics
Department of Cybernetics and Artificial Intelligence
Letná 9
042 00 Košice
Tel.: ++421 55 602 4389, 2564
E-mail: anna.filasova@tuke.sk, dusan.krokavec@tuke.sk

# An Effective Robust Controller Algorithm Design

**Ján Cigánek, Štefan Kozák**

**Abstract**

The paper deals with simple and robust discrete controller design using new approach using the reflection vectors techniques. We assume that the controller structure consists of feed-forward and feedback parts. Developed robust algorithms were tested on the case study examples for different dynamical processes stable, unstable and oscillating type. Simulations were realized in MATLAB-Simulink. Obtained numerical and simulation results confirm applicability of the theoretical principles for robust control of processes subject to parametric model uncertainty.

**Keywords:** robust control, robust stability, parametrical uncertainty, pole-placement, quadratic programming

## Introduction

During last ten years, development of robust control elementary principles and evolution of new robust control methods for different model uncertainty types are visible. Progress in new techniques and theories in control of processes with model uncertainty is necessary because of performance requirements on control of complex processes containing large number of loops, activities coordination of a many agents in hybrid and stochastic control of systems containing large plant model uncertainties. Based on theoretical assumptions, modeling and simulation methods, an effective approach to the control of processes with strong and undefined uncertainties is designed. Such uncertainties are typical for biotechnology processes, chemical plants, automobile industry, aviation etc. For such processes is necessary to design robust and practical algorithms which ensures the high performance and robust stability using proposed mathematical techniques with respect the parametric and unmodelled uncertainties.  Solution to such problems is possible using robust predictive methods and „soft-techniques“ which include fuzzy sets, neuron networks and genetic algorithms.

Robust control is used to guarantee stability of plants with parameter changes. The robust controller design consists of two steps:

- analysis of parameter changes and their influence for closed-loop stability,
- robust control synthesis.

In hybrid control structures that combine the discrete controller and continuous plant, it is difficult to assess the closed-loop stability. One possibility is transformation of the controller and the continuous plant to the discrete-time region and specifying requirements for the discrete controller design. The problem of the robust controller design can be solved as:

- Time-optimal robust controller design
- Design of the robust controller based on the pole-placement

In both parts of the robust controller design it is possible to evolve from the solution of Diophantine equations.

## 1. Problem Formulation

Consider the robust control synthesis of a scalar discrete-time control loop. Transfer function of the original continuous-time system is described by the transfer function

$$G_P(s)=\frac{\bar{B}(s)}{\bar{A}(s)}e^{-Ds}= \qquad (1)$$
$$=\frac{\bar{b}_m s^m+\bar{b}_{m-1}s^{m-1}+\ldots+\bar{b}_0}{\bar{a}_n s^n+\bar{a}_{n-1}s^{n-1}+\ldots+\bar{a}_0}e^{-Ds}$$

Transfer function of (1) can be converted to its discrete-time counterpart

$$G_P\left(z^{-1}\right)=\frac{b_0+b_1z^{-1}+\ldots+b_nz^{-n}}{1+a_1z^{-1}+\ldots+a_nz^{-n}}z^{-d} \qquad (2)$$

For the plant (2) a discrete-time controller is to be designed in form

$$G_R\left(z\right)=\frac{q_0+q_1z^{-1}+\ldots+q_\upsilon z^{-\upsilon}}{1+p_1z^{-1}+\ldots+p_\mu z^{-\mu}}=\frac{Q(z)}{P(z)} \qquad (3)$$

The corresponding closed-loop characteristic equation is

$$1+G_P(z^{-1})G_R(z^{-1})=0 \qquad (4)$$

Substituting (3) and (2) in (4) after a simple manipulation yield the characteristic equation

$$\begin{aligned}&1+G_PG_R=\\&=(1+p_1z^{-1}+\ldots+p_\mu z^{-\mu})(1+a_1z^{-1}+\ldots+a_nz^{-n})+\\&+(q_0+q_1z^{-1}+\ldots+q_\upsilon z^{-\upsilon})(b_1z^{-1}+\ldots+b_nz^{-n})z^{-d}=0\end{aligned} \qquad (5)$$

Unknown coefficients of the discrete controller can be designed using various methods. In this paper robust controller design method based on reflection vectors is used.

The pole assignment problem is as follows: find a controller $G_R(z)$ such that *C(z)=e(z)* where *e(z)* is a given (target) polynomial of degree *k*. It is known [8] that when $\mu = n-1$, the above problem has a solution for arbitrary *e(z)* whenever the plant has no common pole-zero pairs. In general for $\mu < n-1$ exact attainment of a desired target polynomial *e(z)* is impossible.

Let us relax the requirement of attaining the target polynomial *e(z)* exactly and enlarge the target region to a polytope *V* in the polynomial space containing the point *e* representing the desired closed-loop characteristic polynomial. Without any restriction we can assume that $a_n = p_0 = 1$ and deal with monic polynomials *C(z)*, i.e. $\alpha_0 = 1$.

Let us introduce the stability measure as *ρ* = *c*$^T$ *c*, where

$$c = S^{-1}C \tag{6}$$

and *S* is a matrix of dimensions *(n + μ + 1) x (n + μ + 1)* representing vertices of the target polytope *V*. For monic polynomials holds

$$\sum_{i=1}^{k+1} c_i = 1 \tag{7}$$

where *k* = *n* + *μ*. If all coefficients are positive, i.e. *c$_i$ > 0, i = 1,..., k + 1,* then the point *C* is placed inside the polytope *V*.

The minimum *ρ* is attained if

$$c_1 = c_2 = \ldots = c_{k+1} = \frac{1}{k+1} \tag{8}$$

Then the point *C* is placed in centre of the polytope *V*.

In matrix form we have

$$C = Gx \tag{9}$$

where *G* is the Sylvester matrix of the plant

$$G=\begin{bmatrix}
0 & 0 & \ldots & 0 & 0 & b_{n+\mu-\upsilon} & b_{n+\mu-\upsilon+1} & \ldots & 0 & 0\\
0 & 0 & \ldots & 0 & 0 & b_{n+\mu-\upsilon-1} & b_{n+\mu-\upsilon} & \ldots & 0 & 0\\
. & . & . & . & . & . & . & . & . & .\\
0 & 0 & \ldots & 0 & 0 & b_{n+\mu-\upsilon-d+1} & b_{n+\mu-\upsilon-d+2} & \ldots & 0 & 0\\
a_n & 0 & \ldots & 0 & 0 & b_{n+\mu-\upsilon-d} & b_{n+\mu-\upsilon-d+1} & \ldots & 0 & 0\\
a_{n-1} & a_n & \ldots & 0 & 0 & b_{n+\mu-\upsilon-d-1} & b_{n+\mu-\upsilon-d} & \ldots & 0 & 0\\
. & . & . & . & . & . & . & . & . & .\\
a_1 & a_2 & \ldots & a_\mu & a_{\mu+1} & b_{\mu-\upsilon-d+1} & b_{\mu-\upsilon-d+2} & \ldots & b_{\mu-d} & b_{\mu-d+1}\\
1 & a_1 & \ldots & a_{\mu-1} & a_\mu & b_{\mu-\upsilon-d} & b_{\mu-\upsilon-d+1} & \ldots & b_{\mu-d-1} & b_{\mu-d}\\
0 & 1 & \ldots & a_{\mu-2} & a_{\mu-1} & b_{\mu-\upsilon-d-1} & b_{\mu-\upsilon-d} & \ldots & b_{\mu-d-2} & b_{\mu-d-1}\\
. & . & . & . & . & . & . & . & . & .\\
0 & 0 & \ldots & a_{d+1} & a_{d+2} & 0 & 0 & \ldots & b_1 & b_2\\
0 & 0 & \ldots & a_d & a_{d+1} & 0 & 0 & \ldots & 0 & b_1\\
0 & 0 & \ldots & a_{d-1} & a_d & 0 & 0 & \ldots & 0 & 0\\
. & . & . & . & . & . & . & . & . & .\\
0 & 0 & \ldots & 1 & a_1 & 0 & 0 & \ldots & 0 & 0\\
0 & 0 & \ldots & 0 & 1 & 0 & 0 & \ldots & 0 & 0
\end{bmatrix}$$

with dimensions $(n+\mu+d+1)\times(\mu+\upsilon+2)$ and *x* is the $(\mu+\upsilon+2)$-vector of controller parameters: $x = [p_\mu, \ldots, p_1, 1, q_\upsilon, \ldots, q_0]^T$.

Now we can formulate the following control design problem: find a discrete controller $G_R(z)$ such that the closed-loop characteristic polynomial *C(z)* is placed:

- In a stable target polytope *V*, $C(z) \in V$ (to guarantee stability)
- As close as possible to a target polynomial *e(z)*, $e(z) \in V$ (to guarantee performance).

Let the polytope *V* denote the $(k+1)\times N$ matrix composed of coefficient vectors $v_j$, $j = 1, \ldots, N$ corresponding to vertices of the polytope *V*.

Then we can formulate the above controller design problem as an optimization task: Find *x* that minimizes the cost function

$$J_1 = \min_x x^T G^T Gx - 2e^T Gx = \min_x \|Gx - e\|^2 \tag{10}$$

subject to the linear constraints

$$G\,x = V\,w(x), \tag{11}$$

$$w_j(x) > 0, \quad j = 1, \ldots, N, \tag{12}$$

$$\sum_j w_j(x) = 1. \tag{13}$$

Here *w(x)* is the vector of weights of the polytope *V* vertices to obtain the point *C* = *G x*. Fulfillment of the latter two constraints (12), (13) guarantees that the point *C* is indeed located inside the polytope *V*. Then, finding the robust pole-placement controller coefficients represents an optimization problem that can be solved using the Matlab Toolbox OPTIM (quadprog) with constraints [9].

Generally *J$_1$* is a kind of distance to the centre of the target polytope *V*. Is it better to use another criterion *J$_2$*, which measures the distance to the Schur polynomial *E(z)*

$$J_2 = (C-E)^T (C-E) = (Gx-E)^T (Gx-E). \tag{14}$$

It is possible to use the weighted combination of *J$_1$* and *J$_2$*

$$J = (1-\alpha)J_1 + \alpha J_2, \quad 0 \le \alpha \le 1 \tag{15}$$

and to solve the following quadratic programming task

$$J = \min_x \left\{ x^T G^T \left[ (1-\alpha)(SS^T)^{-1} + \alpha I_{k+1} \right] Gx - 2\alpha E^T Gx \right\}, \tag{16}$$
$$S^{-1}Gx < 0.$$

Assume the discrete robust controller design task with parametrical uncertainties in system description. Let us also assume that coefficients of the discrete-time system transfer functions *a$_n$* , ..., *a$_1$* and *b$_n$* , ..., *b$_1$* are placed in polytope *W* with the vertices $d^j = [a_n^j, \ldots a_1^j, b_n^j, \ldots, b_1^j]$:)

$$W = conv\{d_j, j = 1, \ldots, M\} \tag{17}$$

As (9) is linear in system parameters, it is possible to claim that for arbitrary vector of the controller coefficients *x* is the vector of the characteristic polynomial coefficients *C(z)* placed in the polytope *A* with vertices $a^1, \ldots, a^M$:

$$A = conv\{a^j, j = 1, \ldots, M\} \tag{18}$$

where $a^j = D^j x$ and $D^j$ is the Sylvester matrix of dimensions *(n + μ + d + 1) x (μ + υ + 2)*, composed of vertices set *d$^j$*, as in case of the exact model (9).

### 1.1 Problem Formulation

The digital controller $x = \left[p_{\mu}, \dots, p_1, 1, q_{\upsilon}, \dots, q_0\right]^T$ is to be designed such that all its vertices *a $^j$, j = 1, ..., M* are placed inside a stable desired target polytope *V*.

This problem can be effectively solved using quadratic programming procedure. It is necessary to find the controller *x* by minimization of the cost function

$$J = \min_x \left\{ x^T \bar{D}^T \begin{bmatrix} (1-\alpha)(I_M \otimes (S^T)^{-1})(I_M \otimes S^{-1}) + \\ + \alpha I_{(k+1)M} \end{bmatrix} \bar{D} x - \right. \tag{19}$$

$$\left. - 2\alpha E^T \bar{D} x \right\}, \quad S^{-1} D^j x > 0, \; j = 1, \dots, M$$

where IM is identity matrix of dimension M, $\otimes$ is the Kronecker product and $\bar{D}^T = \left[D_1^T, \dots, D_M^T\right]$.

### 1.2 Stable Region Computation via Reflection Coefficients

Polynomials are usually specified by their coefficients or roots. They can be characterized also by their reflection coefficients using Schur-Cohn recursion.

Let $C_k(z^{-1})$ be a monic polynomial of degree *k* with real coefficients $c_i \in$ R, *i = 0, ..., k,*

$$C(z^{-1}) = 1 + c_1 z^{-1} + \dots + c_k z^{-k}. \tag{20}$$

Reciprocal polynomial of the polynomial $C_k(z^{-1})$ is defined in [11] as follows

$$C_k^*(z^{-1}) = c_k + c_{k-1} z^{-1} + \dots + c_1 z^{-k+1} + z^{-k} \tag{21}$$

Reflection coefficients *r$_i$, i = 1, ..., k*, can be obtained from the polynomial $C_k(z^{-1})$ using backward Levinson recursion [12]

$$z^{-1} C_{i-1}(z^{-1}) = \frac{1}{1 - \left| r_i^2 \right|} \left[ C_i(z^{-1}) - r_i C_i^*(z^{-1}) \right] \tag{22}$$

where $r_i = -c_i$ and $c_i$ is the last coefficient of $C_i(z^{-1})$ of degree *i*. From (22) we obtain in a straightforward way:

$$C_i(z^{-1}) = z^{-1} C_{i-1}(z^{-1}) + r_i C_{i-1}^*(z^{-1}). \tag{23}$$

Expressions for polynomial coefficients $C_{i-1}(z^{-1})$ and $C_i(z^{-1})$ result from equations (22) and (23):

$$C_{i-1}(z^{-1}) = \frac{1}{1 - \left| r_i^2 \right|} \left[ \sum_{j=0}^{i-1} \left( c_{i,j+1} - r_i c_{i,i-j-1} \right) z^{-j} ) \right] \tag{24}$$

$$C_i(z^{-1}) = \sum_{j=0}^{i} \left( c_{i-1,j-1} + r_i c_{i-1,i-j-1} \right) z^{-j}. \tag{25}$$

The reflection coefficients *r$_i$* are also known as Schur-Szegö parameters [11], partial correlation coefficients [6] or *k*-parameters [13]. Presented forms and structures were effectively used in many applications of signal processing [13] and system identification [6]. A complete characterization and classification of polynomials using their reflection coefficients instead of roots (zeros) of polynomials is given in [11].

The main advantage of using reflection coefficients is that the transformation from reflection to polynomial coefficients is very simple. Indeed, according to (23) and (25), polynomial coefficients *c$_i$* depend multilinearly on the reflection coefficients *r$_i$*. If the coefficients $c_i \in R$ are real, then also the reflection coefficients $r_i \in R$ are real.

Transformation from reflection coefficients *r$_i$, i = 1, ..., k*, to polynomial coefficients *c$_i$, i = 1, ..., k*, is as follows

$$c_i = c_i^{(k)}, \qquad c_i^{(i)} = -r_i,$$

$$c_j^{(i)} = c_j^{(i-1)} - r_i c_{i-j}^{(i-1)}, \tag{26}$$

*i = 1, ..., k;  j = 1, ..., i – 1*

or in the matrix form

$$C = R(r) c^{(t)}, \quad t = 1, \dots, k-1,$$

$$C^{(t)} = \begin{bmatrix} 0^T \\ R_t(r_t) \end{bmatrix} C^{(t-1)}, \tag{27}$$

where

$$C = \left[c_k, \dots, c_1, 1\right]^T,$$

$$C^{(t)} = \left[0, c_t^{(t)}, \dots, c_1^{(t)}, 1\right]^T,$$

$$C^{(0)} = \left[0, 1\right]^T,$$

$$R(r) = R_k(r_k) \begin{bmatrix} 0^T \\ R_{k-1}(r_{k-1}) \end{bmatrix} \cdots \begin{bmatrix} 0^T \\ R_t(r_t) \end{bmatrix},$$

$$R_j(r_j) = I_{j+1} - r_j E_{j+1},$$

where $I_k$ is a *k* x *k* identity matrix, $E_k = \begin{bmatrix} 0 \dots 1 \\ . \;\; . \;\; . \\ 1 \dots 0 \end{bmatrix}$ and $0^T$ is a row vector of zeros.

*Lemma 1.* A linear discrete-time dynamic system is stable if its characteristic polynomial is Schur stable, i.e., if all its poles lie inside the unit circle.

The stability criterion in terms of reflection coefficients is as follows [11].

*Lemma 2*. A polynomial $C(z^{-1})$ has all its roots inside the unit disk if and only if $|r_i| < 1$, *i = 1, ..., k*.

A polynomial $C(z^{-1})$ lies on the stability boundary if some $r_i = \pm 1$, *i = 1, ..., k*. For monic Schur polynomials there is a one-to-one correspondence between $C = \left[c_k, \dots, c_1\right]^T$ and $r = \left[r_1, \dots, r_k\right]^T$.

Stability region in the reflection coefficient space is simply the *k*-dimensional unit hypercube $R = \{r_i \in (-1, 1), i = 1, \dots, k\}$. The stability region in the polynomial coefficient space can be found starting from the hypercube *R*.

### 1.3 Stable Polytope of Reflection Vectors

It will be shown that for a family of polynomials the linear cover of the so-called reflection vectors is Schur stable.

*Definition 1.* The reflection vectors of a Schur stable monic polynomial *C(z$^{-1}$)* are defined as the points on stability boundary in polynomial coefficient space generated by changing a single reflection coefficient $r_i$ of the polynomial *C(z$^{-1}$)*.

Let us denote the positive reflection vectors of *C(z$^{-1}$)* as $v_i^+(C) = (C|r_i = 1), i = 1,\dots,k$, and the negative reflection vectors of *C(z$^{-1}$)* as $v_i^-(C) = (C|r_i = -1), i = 1,\dots,k$.

The following assertions hold:

1. every Schur polynomial has *2k* reflection vectors $v_i^+(C)$ and $v_i^-(C), i = 1,\dots,k;$

2. all reflection vectors lie on the stability boundary $(r_i^v = \pm 1);$

3. the line segments between reflection vectors $v_i^+(C)$ and $v_i^-(C)$ are Schur stable.

In the following theorem a family of stable polynomials is defined such that the polytope generated by reflection vectors of these polynomials is stable.

*Theorem 1.* Consider $r_1^C \in (-1,1)$, $r_k^C \in (-1,1)$ and $r_2^C = \dots = r_{k-1}^C = 0$. Then the inner points of the polytope *V(C)* generated by the reflection vectors of the point *C*

$$V(C) = conv\left\{v_i^{\pm}(C), i = 1,\dots,k\right\} \tag{28}$$

are Schur stable.

### 1.4 Roots of Reflection Vectors

In this section we study the root placement of reflection vectors. It is useful for selecting a stable target simplex to solve the robust output control problem.

By definition, at least one root of a reflection vector $v_i(C)$ (i.e. root of $V_i(z^{-1}) = \left[z^{-k} \dots z^{-1}\ 1\right]\begin{bmatrix} v_i(C) \\ 1 \end{bmatrix}$) must lie on the unit circle, and the number of unit circle roots is determined by the number *i* of the reflection vector $v_i(C)$ and the character of the roots (real or complex) is determined from the sign of the boundary reflection coefficient $(r_i^V = \pm 1)$.

### 1.5 Robust Controller Design

A robust controller is to be designed such that the closed-loop characteristic polynomial is placed in the stable polytope (linear cover) of reflection vectors. It means that the following problems have to be solved:

1. choice of initial polynomial *C(z-1)* for generating the polytope *V(C),*

2. choice of *k + 1* most suitable vertices of *V(C)* to build a target simplex *S*,

3. choice of a target polynomial *E(z-1).*

In the following section some “thumb rules” are given for choosing a stable target simplex *S*.

To choose *k + 1* vertices of the target simplex *S* we use the well known fact that poles with positive real parts are preferred to those with negative ones [1]. The positive reflection vectors $v_i^+(C)$ with *i* odd and negative reflection vectors $v_i^-(C)$ with *i* even are chosen yielding *k* vertices. The *(k+1)*th vertex of the target simplex *S* is chosen as the mean of the remaining reflection vectors.

The target polynomial *E(z$^{-1}$)* of order *k* is reasonable to be chosen inside the stable polytope of reflection vectors *V(C)*. A common choice is *E(z$^{-1}$)=C(z$^{-1}$)*.

For higher-order polynomials the size of the target simplex *S* is considerably less than the volume of the polytope of reflection vectors *V*. That is why the above quadratic programming method with a preselected target simplex *S* works only if uncertainties are sufficiently small. Otherwise it is reasonable to use some search procedure to find a robust controller such that the polytope of closed-loop characteristic polynomial is placed inside the stable polytope of reflection vectors *V(C).*

## 2. Illustrative example – Controller Design via Reflection Coefficients

Consider the approximate model of a PUMA 762 robotic disk grinding process [4]. From the results of identification and because of the nonlinearity of the robot, the coefficients of the numerator of the plant transfer function change for different positions of the robot arm:

$$G_p(z^{-1}) = \frac{(0.0257 \pm 0.0005)z^{-1} - (0.0764 \pm 0.01528)z^{-2} - \\ -(0.1619 \pm 0.03238)z^{-3} - (0.1688 \pm 0.03376)z^{-4}}{1 - 1.9140\,z^{-1} + 1.7790\,z^{-2} - 1.0265\,z^{-3} + 0.2508\,z^{-4}}. \tag{29}$$

The task is to design a discrete-time controller (3), *υ=μ=3*.

From the transfer function (29) and matrix form of (9) results

$$C = \begin{bmatrix} 0.2508 & 0 & 0 & 0 & -0.1688 & 0 & 0 & 0 \\ -1.0265 & 0.2508 & 0 & 0 & -0.1619 & -0.1688 & 0 & 0 \\ 1.7790 & -1.0265 & 0.2508 & 0 & -0.0764 & -0.1619 & -0.1688 & 0 \\ -1.9140 & 1.7790 & -1.0265 & 0.2508 & 0.0257 & -0.0764 & -0.1619 & -0.1688 \\ 1 & -1.9140 & 1.7790 & -1.0265 & 0 & 0.0257 & -0.0764 & -0.1619 \\ 0 & 1 & -1.9140 & 1.7790 & 0 & 0 & 0.0257 & -0.0764 \\ 0 & 0 & 1 & -1.9140 & 0 & 0 & 0 & 0.0257 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_3 \\ p_2 \\ p_1 \\ 1 \\ q_3 \\ q_2 \\ q_1 \\ q_0 \end{bmatrix}$$

Let us choose the initial polynomial *C(z$^{-1}$)* for generating the polytope *V(C)* as follows

$$C(z) = [1 - 0.3z^{-1}][1 - 0.2z^{-1}]^2 \\ [1 - 0.1z^{-1}][1 + 0.2z^{-1}]^2[1 + 0.3z^{-1}] \tag{30}$$

with reflection coefficients $r_1 = 0.1$, $r_2 = 0.17$, $r_3 = -0.017$, $r_4 = -0.0088$, $r_5 = 0.00088$, $r_6 = 0.000144$, $r_7 = -0.0000144$

Now we can find the reflection vectors $v_i(C)$ of the initial polynomial *C(z$^{-1}$)* leading to the matrix form of the target simplex *S* (vertex polynomial coefficients)

$$S(E) = \begin{bmatrix} 0.7 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 \\ 0 & -0.2 & -0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 & 0 & -0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0.3 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

The discrete-time controller design task for the nominal transfer function (29) has been solved via quadratic programming taking *α=0.1* in the cost function *J* (16).

For the selected target simplex *S* we have obtained the following discrete-time feedback controller

$$G_{FB}\left(z^{-1}\right)=\frac{-1.765+1.857z^{-1}-2.241z^{-2}+0.169z^{-3}}{1+1.959z^{-1}+1.788z^{-2}+0.843z^{-3}} \qquad (31)$$

with control law

$$\begin{aligned}u_2(k)&=-1.96u_2(k-1)-1.79u_2(k-2)-0.84u_2(k-3)-\\&-1.77y(k)+1.86y(k-1)-2.24y(k-2)+0.17y(k-3)\end{aligned}. \qquad (32)$$

Corresponding closed-loop step responses under the feedback controller (31) are in Fig.1.

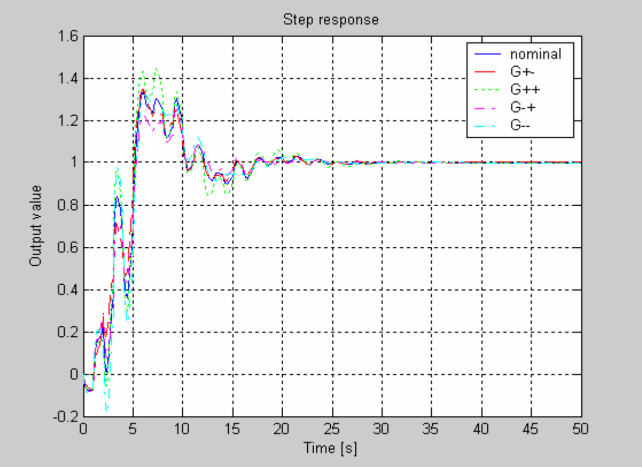


**Fig.1 Closed-loop step responses under robust controller**

## Conclusion

The paper deals with the development of robust methods based on reflection vectors methodology for computation of control law coefficients guaranteeing stability, robustness and high performance with respect to parameter uncertainties. Theoretical results were verified on the examples for feedback and feedforward control structures. Proposed methods were tested for both stable and unstable processes.

The paper proposes theoretical principles and design methodology of robust discrete-time controllers for systems with parametric uncertainties.

The illustrative example was solved using quadratic programming for suitably defined cost function. Simulation results prove applicability of the proposed robust controller design theory for systems with parametric uncertainty. The proposed robust techniques present conservative solution without performance analysis. For better performance is necessary to prove different values of the matrix *S* and poles of initial polynomial *C*($z^{-1}$).

### Acknowledgement

*This paper was supported by Slovak Scientific Grant Agency -VEGA No 1/0822/08.*

## References

[1] ACKERMANN, J.: “Robust Control: Systems with Uncertain Physical Parameters“. Springer-Verlag. London, UK, 1993.

[2] BUDINSKY, M., KOZAK, S., ZALMAN, M.: “Design of hybrid control systems based on the robust pole region approach”, in Preprints 2nd *IFAC Workshop on New Trends in Design of Control Systems* Smolenice, Slovak Republic, 1997.

[3] CIGANEK, J., KOZAK. S.: “Robust controller design using algebraic theory”, in: *Int. conf. Cybernetics and informatics*, Zdiar, Slovak Republic, 2008.

[4] HENRION, D., SEBEK, M., KUCERA, V.: „Positive polynomials and rébusy stabilization with fixed-order controllers“. *IEEE Transactions on Automatic Control,* Vol. 48, pp.1178-1186, 2003.

[5] HENRION, D., ARZELIER, D., PEAUCELLE, D., SEBEK, M.: „An LMI Condition for Robust Stability of Polynomial Matrix Polytopes“. *Automatica,* Vol. 37, pp.461-468, 2001.

[6] KUČERA, V.: „Discrete linear control: The polynomial equation approach“. Chichester, J. Wiley & Sons, 1979.

[7] KAY, S. M.: „Modern Spectral Estimation“. Prentice Hall, New Jersey. 1988.

[8] NURGES, U.: “Robust pole assignment via reflection coefficients of polynomials”. *Automatica*, Vol. 42, pp.1223-1230, 2006.

[9] KEEL, L. H., BHATTACHARYYA, S. P.: „A linear programming approach to controller design“. *Automatica*, Vol. 35, pp. 1717–1724, 1999.

[10] NURGES, U.: „New stability conditions via reflection coefficients of polynomials“. *IEEE Trans. AC*, Vol. 50, pp. 1354–1360, 2005.

[11] DIAZ-BARRERO, J. L., EGOZCUE, J. J.: „Characterization of polynomials using reflection coefficients“. *Appl. Math. E-Notes*, Vol. 4, 114–121, 2004.

[12] PICINBONO, B., BENIDIR, M., „Some properties of lattice autoregressive filters“, *IEEE Trans. Acoust. Speech Signal Process*, Vol. 34, pp. 342–349, 1986.

[13] OPPENHEIM, A. M., SCHAFFER, R. W., „Discrete-Time Signal Processing“. Prentice-Hall, Englewood Cliffs, 1989.

**Ing. Ján Cigánek**
**prof. Ing. Štefan Kozák PhD.**

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ústav aplikovanej informatiky
Ilkovičova 3
842 16 Bratislava
Tel.: +421 2 654 29 502
Fax: +421 2 654 20 587
E-mail: info@fiit.stuba.sk

# Robust control of a laboratory process

Katarína Vaneková , Monika Bakošová, Radek Matušů, Jana Závacká

**Abstract**

The paper presents robust control of a laboratory process with a transport delay using the industrial control system SIMATIC. The controlled process is identified in the form of a transfer function of a higher order with a transport delay at first and then the transport delay is approximated by the first order Taylor series expansion of the numerator or the denominator. Because the transport delay can vary, the controlled laboratory process is modelled in the form of a transfer function with interval parametric uncertainty. Robust PI controllers are designed for the laboratory process. The method for synthesis of robust controllers is based on plotting the stability boundary locus in the ($k_p$, $k_i$) - plane and the subsequent choice of a stabilizing PI controller using the pole-placement method so that the prescribed behavior of the closed-loop is achieved.

**Keywords**: PI controller, robust control, interval uncertainty, control system SIMATIC, transport delay

## 1. Introduction

The field of robust control has experienced a large number of breakthroughs over last decades. The primary focal points have been robustness analysis [5] and robustness synthesis involving structured real parametric uncertainty, see e.g. [3]. Numerous interests have grown in various problems of analysis, synthesis, and design for interval plants.

There has been done a great amount of research work on tuning of PID controllers since these types of controllers have been widely used in industrial applications [7, 11]. PID controller design in classical control engineering is based on a plant with fixed parameters and the latest approaches can be found e.g. [8]. In the real world, however, most process models are not known exactly and so, models contain uncertainties. Hence control system design for both, stability and performance robustness always requires taking uncertainties into account.

In this paper, a method for design of robust PI controllers is used, see [10]. The method is based on plotting the stability boundary locus in the (*kp*, *ki*)-plane and the subsequent choice of a stabilizing PI controller using the pole-placement method so that the prescribed behavior of the closed-loop is achieved.

## 2. SIMATIC S7-300

SIMATIC S7-300 is an industrial control system, which is used in many applications of process control. SIMATIC includes programming (STEP7) and visualization (WinCC) software, which are used for programming of programmable logic controllers (PLCs), for data accessing to users and they are simply applicable for monitoring and control of real processes.

The structure of the user's program is created by the organization block OB35, witch represents the main program that works cyclic with the sample time 100ms. The organization block OB35 includes a function block of a PID controller (FB41). Before the blocks are programmed [6, 9], it is necessary to create a project, configure a network, define input and output modules and define connections between input and output modules. Visualization of the project is realized in the Graphics Designer. Visualization software WinCC gives to users a possibility to define their own visualization for controlled processes. The component of WinCC is a graphic editor. WinCC allows choice of manipulating elements, I/O fields and monitoring windows according to demands of users. WinCC processes all important data from the program STEP7 and the connection between WinCC and STEP7 is linked by tags. For visualization of the controlled laboratory process a visualization screen has been created. All measured data and their graphic trends are displayed. User can design any objects for creating of a visualization screen individually or objects from a library can be chosen.

## 3. Laboratory process

Controlled laboratory process (Fig. 1) is an electronic model of a linear 2$^{nd}$ order system with a transport delay varying from 6s to 30s [1]. The process was identified by Strejc method [4] in the form of a transfer function

$$G_s(s) = \frac{K}{(T\,s+1)^2}\,e^{-[D_{\min},D_{\max}]s} \qquad (1)$$

where *K* is the gain, *T* is the time constant and $D_{min}$, $D_{max}$ are the minimun and the maximum transport delays of the process. The process was identified from step responses measured in various working areas and the identified parameters are collected in Table 1.

| *K* | *T(s)* | *D(s)* | *n* |
|---|---|---|---|
| 0.97 | 10.0 | 12.6 | 2 |
| 0.97 | 10.1 | 17.1 | 2 |
| 0.97 | 10.4 | 22.4 | 2 |
| 0.97 | 10.7 | 35.4 | 2 |

**Tab.1 Identified parameters**

## 4. Robust controller design

### 4.1 Destription of an uncertain system

Consider a system with real parametric uncertainty described by the transfer function

$$G_s(s,q) = \frac{b(s,q)}{a(s,q)} \qquad (2)$$

where *q* is a vector of uncertain parameters and *b*, *a* are polynomials in s with coefficients which depend on a parameter *q*.

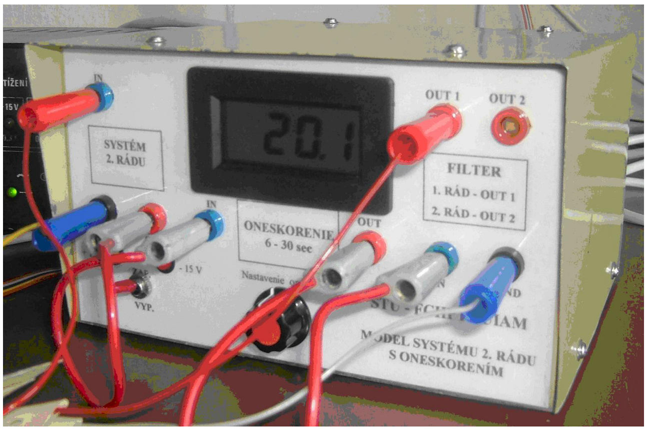


**Fig.1 Controlled laboratory process**

An uncertain polynomial

$$a(s,q)=\sum_{i=1}^{n} a_i(q)s^i \tag{3}$$

is said to have an independent uncertainty structure if each component $q_i$ of *q* enters into only one coefficient $a_i$.

A family of polynomials

$$A=\{a(s,q):q\in Q\} \tag{4}$$

is said to be an interval polynomial family if *a(s, q)* has an independent uncertainty structure, each coefficient depends continuously on *q* and *Q* is a box. An interval polynomial family *A* arises from the uncertain polynomials described by *a(s, q)* with uncertainty bounds $|q_i| \leq 1$ for $i = 0, . . . , n$. When dealing with an interval family, the shorthand notation

$$a(s,q)=\sum_{i=1}^{n}\left[q_i^{=},q_i^{+}\right]s^i \tag{5}$$

may be used with $[q_i^-, q_i^+]$ denoting the bounding interval for the $i^{th}$ of uncertainty component of uncertainty $q_i$.

### 4.2 Analysis of robust stability

In order to use the Kharitonov theorem [5] for robust stability analysis, polynomials associated with an interval polynomial family A have to be defined at first. In the definition below the polynomials are fixed in the sense that only the bounds $q_i^-$ and $q_i^+$ enter into the description but not the $q_i$ themselves. The number of polynomials is four and they are independent on the degree of *a(s, q)*. Associated with the interval polynomial family (5) are four fixed Kharinov polynomials [5]

$$\begin{aligned}
K_1(s)&=q_0^- + q_1^- s + q_2^+ s^2 + q_3^+ s^3 + \ldots\\
K_2(s)&=q_0^+ + q_1^+ s + q_2^- s^2 + q_3^- s^3 + \ldots\\
K_3(s)&=q_0^+ + q_1^- s + q_2^- s^2 + q_3^+ s^3 + \ldots\\
K_4(s)&=q_0^- + q_1^+ s + q_2^+ s^2 + q_3^- s^3 + \ldots
\end{aligned} \tag{6}$$

The interval polynomial family A with invariant degree is robustly stable if and only if its four Kharitonov polynomials (6) are stable.

### 4.3 Decription of PI controller synthesis

The method of a robust PI controller synthesis [10] is based on plotting the stability boundary locus in the ($k_p$, $k_i$)-plane and subsequent finding of stabilizing PI controllers. The method locates all PI controllers, which stabilize the controlled system with interval uncertainty. The stability boundary divides the parameter plane (($k_p$, $k_i$)-plane) into stable and unstable regions. The stable ones can be determined by the choice of a test point within each region.

### 4.4 Robust PI controller synthesis I

Consider the control system in Figure 2, where $G_s(s)$ represents the controlled process with the transfer function

$$G_s(s)=\frac{-b_1 s+b_0}{a_2 s^2+a_1 s+1} \tag{7}$$

and *C(s)* represents the feedback stabilizing PI controller

$$C(s)=k_p+\frac{k_i}{s} \tag{8}$$

with $k_p = Z_R$ and $k_i = Z_R/T_I$, where $Z_R$ is the gain of the controller and $T_I$ is the reset time of the controller. The closed loop characteristic equation can be written by substituting $s=j\omega$ in the form

$$\begin{aligned}
&\left[-a_1\omega^2+b_1k_p\omega^2+b_0k_i\right]+\\
&\qquad +j\left[-a_2\omega^3+\left(b_0k_p-b_1k_i+1\right)\omega\right]=0
\end{aligned} \tag{9}$$

The parameters of PI controller can be easily obtained by equating the real and the imaginary parts of the characteristic equation (9) to zero, for details see [12]. Equating the real and imaginary parts of (9) to zero leads following expressions for calculating of $k_p$, $k_i$ in the dependence on the frequency *ω*

$$k_p=\frac{\omega^2\left(a_l b_0+a_1b_1\right)-b_0}{b_0^2+b_1^2\omega^2} \tag{10a}$$

$$k_i=\frac{\omega^2\left(a_1-k_pb_1\right)}{b_0} \tag{10b}$$

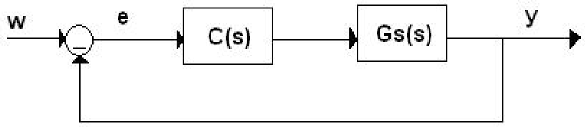


**Fig.2 Control system**

### 4.5 Robust PI controller synthesis II

Consider further the control system in Figure 2, where $G_s(s)$ represents the controlled process with the transfer function

$$G_s(s)=\frac{b_0}{a_3s^3+a_2s^2+a_1s+a_0} \tag{11}$$

and *C(s)* is the feedback stabilizing PI controller (8). The closed loop characteristic equation after the substitution $s=j\omega$ is

$$\left[a_3\omega^4 + a_{21}\omega^2 + b_0 k_i\right] + \\ + j\left[-a_2\omega^3 + \left(b_0 k_p + a_0\right)\omega\right] = 0 \qquad (12)$$

Equating the real and imaginary parts of the characteristic equation to zero gives following expressions for calculating of $k_p$, $k_i$ in the dependence on the frequency $\omega$

$$k_p = \frac{a_2\omega^2 + a_0}{b_0} \qquad (13a)$$

$$k_i = \frac{\omega^2\left(-a_3\omega^2 + a_1\right)}{b_0} \qquad (13b)$$

## 5. Pole-placement method

For the control system in Figure 2, where $G_s(s)$ represents the controlled system of the 2nd or the 3rd order and $C(s)$ represents the PI controller (8), the closed loop characteristic equation can be

$$\left(s + c_1\right)\left(s^2 + 2\,\xi\,\omega_0\ s + \omega_0^2\right) = 0 \qquad (14a)$$

$$\left(s + c_1\right)^2\left(s^2 + 2\,\xi\,\omega_0\ s + \omega_0^2\right) = 0 \qquad (14b)$$

where $\xi$ is the relative damping, $\omega_0$ is the natural undamped frequency and $-c_1$ is the pole of the closed-loop system [2]. The closed loop characteristic equation for the considered controlled systems (7) or (11) and the PI controller has the form

$$s^3 + \frac{a_1 - b_1 k_p}{a_2} s^2 + \\ + \frac{a_0 + b_0 k_p - b_1 k_i}{a_2} s \ + \frac{b_0 k_i}{a_2} = 0 \qquad (15a)$$

or

$$s^4 + \frac{a_2}{a_3} s^3 + \frac{a_1}{a_3} s^2 + \\ + \frac{a_0 + b_0 k_p}{a_3} s \ + \frac{b_0 k_i}{a_3} = 0 \qquad (15b)$$

After the suitable choice of $\xi$, $\omega_0$, $c_1$ and comparison of coefficients in (14a) and (15a), the parameters of the PI controller can be computed as follows

$$k_p = \frac{a_2\left(c_1 + 2\,\xi\,\omega_0\right) - a_1}{-b_1} \qquad (16a)$$

$$k_i = \frac{c_1 a_2 \omega_0^2}{b_0} \qquad (16b)$$

Simularly, after the suitable choice of $\xi$, $\omega_0$, $c_1$ and comparison of coefficients in (14b) and (15b), the parameters of the PI controller can be computed as follows

$$k_p = \frac{a_3\left(2\,\xi\,\omega_0 c_1^2 + 2\,c_1\,\omega_0^2\right) - a_0}{b_0} \qquad (17a)$$

$$k_i = \frac{c_1^2 a_2 \omega_0^2}{b_0} \qquad (17b)$$

pole-placement method was used for the nominal model of the controlled process with following values of identified parameters: the gain $K = 0.97$, the time constant $T = 10.3s$ and the transport delay $D_{nom} = 24.0s$.

## 6. Results

### 6.1 Application of robust controller synthesis I and pole-placement method

The identified transfer function (1) of the laboratory process was modified by approximation of the transport delay. The term representing the transport delay in (1) was aproximated by its 1st order Taylor series expansion of the numerator. So, the modified transfer function has the form (7) where

$$\begin{aligned} a_2 &= T^2 \\ a_1 &= 2T \\ a_0 &= 1 \\ b_1 &= \left[KD^-, KD^+\right] \\ b_0 &= K. \end{aligned} \qquad (18)$$

It can be stated according to (18) that the controlled process is a system with parametric interval uncertainty.

The parameters of robust PI controllers were found by the method described in the part 4.4 and 5.

In the stability region (Fig. 3) were found parameters of PI controllers (16) for following choice of characterictic equation parameters: $\xi = 0.77$, $c1 \in [0.42 : 0]$ and $\omega_0 \in [0 : 0.10]$. Found $k_p$ and $k_i$ lie on the magenta line in Figure 4. From designed $k_p$, $k_i$ parameters, two PI controllers were chosen: *C1* with $\xi = 0.77$, $c_1 = 0.12$ and $\omega_0 = 0.03$, and *C2* with $\xi = 0.77$, $c_1 = 0.11$ and $\omega_0 = 0.02$ (Tab. 2) (green stars in Figure 4).

| | $Z_R$ | $T_I$ |
|---|---|---|
| *C1* | 0.10 | 7.0 |
| *C2* | 0.20 | 29.1 |

**Tab.2 The parameters of PI controlers I**

The robust stability of the designed feedback control loop was also tested and the Kharitonov theorem was used. The characteristic equation of the feedback control loop is

$$1 + C(s)G_s(s) = 0 \qquad (19)$$

where parameters of $G_s(s)$ are given in (18) and parameters of $C(s)$ are given in Table (2). Four fixed Kharitonov polynomials for the characteristic equation were created according to (6) and their stability was tested. The command kharit from the Polynomial Toolbox was used and the result of this test is, that the polynomial on the left side of (19) is robustly stable. It means that the feedback control loop with designed controllers is robustly stable.

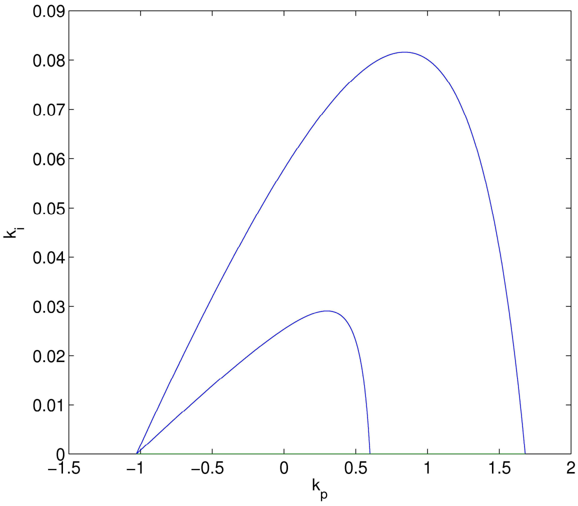


**Fig.3 Stability region for (18)**

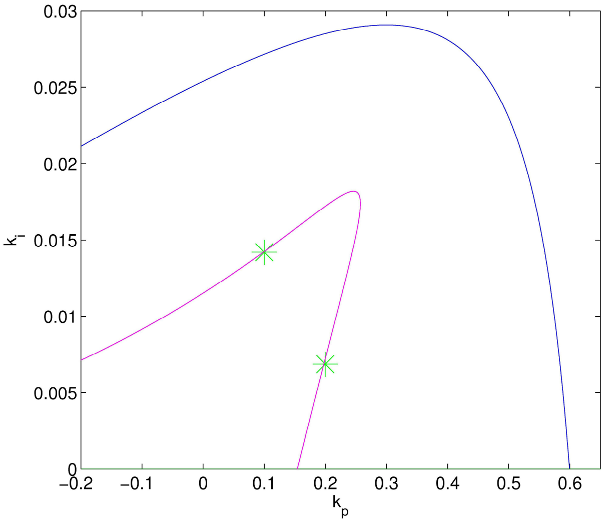


**Fig.4 The position of controllers in the stability region for (18)**

### 6.2 Application of robust controller synthesis II and pole-placement method

The identified transfer function (1) was modified by approximation of the transport delay. The term representing the transport delay in (1) was substituted by its 1st order Taylor series expansion of the denominator. So the transfer function has the form (11) where

$$\begin{aligned} a_3 &= \left[T^2 D^-, T^2 D^+\right] \\ a_2 &= \left[T^2 + 2TD^-, T^2 + 2TD^+\right] \\ a_1 &= \left[2T + D^-, 2T + D^+\right] \\ a_0 &= 1 \\ b_0 &= K. \end{aligned} \qquad (20)$$

According to (20), it can be stated, that the controlled process is a system with parametric interval uncertainty.

The parameters of robust PI controllers were found by the method described in the part 4.5 and 5.

In the stability region (Fig. 5), parameters of PI controllers (17) were found for *ξ = 1.1, $c_1$ ∈ [0.12 : 0.06]* and *$\omega_0$ ∈ [0 : 0.05]*. Found $k_p$ and $k_i$ lie on the magenta line in Figure 6.

From designed $k_p$, $k_i$ parameters was chosen the PI controller *C3* with *ξ = 1.1, $c_1$ = 0.09* and *$\omega_0$ = 0.03* (Tab. 3)(green star in Figure 6).

| | $Z_R$ | $T_I$ |
|---|---|---|
| *C3* | 0.50 | 34.3 |

**Tab.3 The parameters of PI controler II**

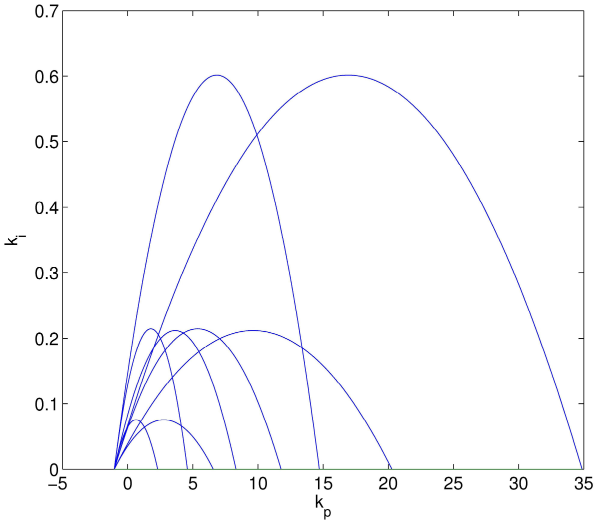


**Fig.5 Stability region for (20)**

The robust stability of the designed feedback control loop was also tested using the Kharitonov theorem by the way as it was described in the previous section. The result of the test confirmed that the polynomial (19) is robustly stable. It means the feedback control loop with the designed controller *C3* is robustly stable

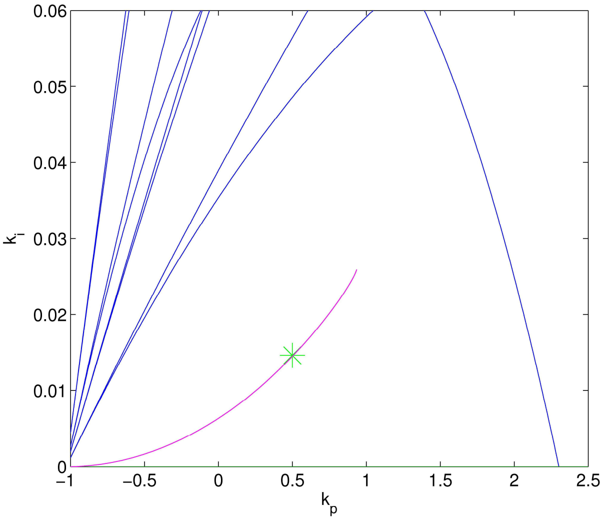


**Fig.6 The position of controllers in the stability region for (20)**

### 6.3 Control of laboratory process

The laboratory process was controlled using the robust PI controllers (Tab. 2) and (Tab. 3). These controllers were implemented via the control system SIMATIC.

Time responses of the closed loop with the controlled process with different values of the transport delay and the PI controller *C1* are shown in Figure 7, where *w* is the setpoint and $D_{min}$, $D_{max}$, $D_{nom}$ are minimal, maximal and nominal transport delays.

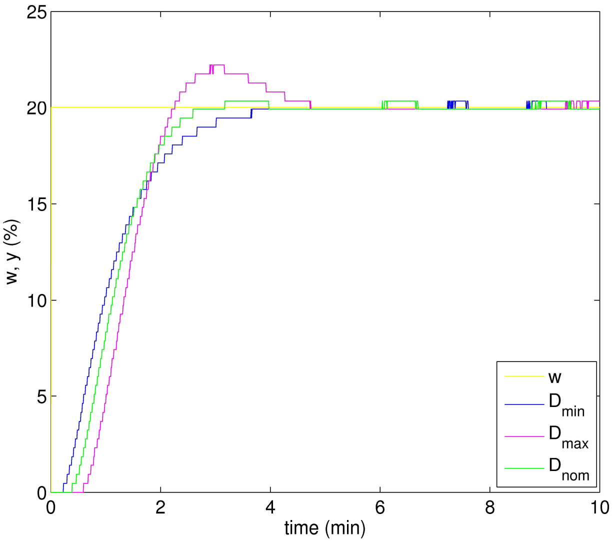


**Fig.7 Time responses for controller C1**

Time responses of the closed loop with the controlled process with different values of transport delay and the PI controller *C2* are shown in Figure 8, where *w* is the setpoint and $D_{min}$, $D_{max}$, $D_{nom}$ are minimal, maximal and nominal transport delays.

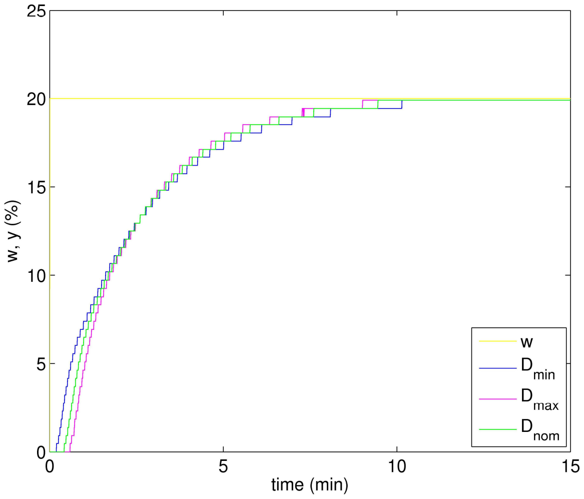


**Fig.8 Time responses for controller C2**

Time responses of the closed loop with the controlled process with different values of transport delay and the PI controller *C3* are shown in Figure 9 where *w* is the setpoint and $D_{min}$, $D_{max}$, $D_{nom}$ are minimal, maximal and nominal transport delays.

## Conclusions

The electronic model was identified as the process with interval uncertainty in the transport delay. For this process, robust PI controllers were designed by combination of two methods: the method based on the stability boundary locus in the $(k_p, k_i)$-plane and the pole-placement method. Adding the pole-placement method to the robust PI controller design offers the possibility to assure the prescribed behavior of the closed loop given by the choice of $\xi$, $c_1$ and $\omega_0$. Designed controllers were implemeted for control of the laboratory process using the control system SIMATIC. Obtained experimental results confirm that the designed robust PI controllers are suitable for control of real processes with varying transport delay.

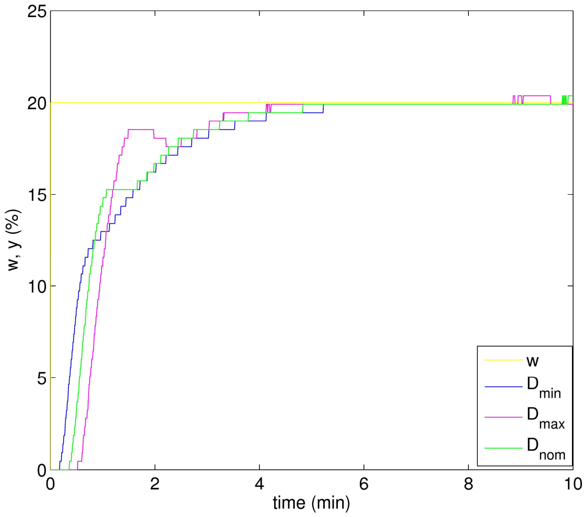


**Fig.9 Time responses for controller C3**

**Acknowledgments**

*The work has been supported by the Scientific Grant Agency of the Slovak Republic under grants 1/4055/07, 1/0071/09 and by the Slovak Research and Development Agency under the project APVV-0029-07, and the Ministry of Education, Youth and Sports of the Czech Republic under Research Plan No. MSM 7088352102.*

## References

[1] BABIRÁD, J.:Model of a 2nd order system with a transport delay (in Slovak). Technical documentation, Bratislava, 2006.

[2] BAKOŠOVÁ, M., FIKAR, M.: Process control (in Slovak). STU Press, Bratislava, 2008.

[3] BARMISH, B. R.: New Tools for Robustness of Linear Systems. Macmillan Publishing Company, New York, 1994.

[4] FIKAR, M.,MIKLEŠ, J.:: System Identification (in Slovak). STU Press, Bratislava, 1999.

[5] KHARITONOV, V.L.: Asymptotic stability of an equilibrium position of a family systems of linear differential equations. Differential Equations 14, pages 2086–2088, 1978.

[6] KOŽKA, Š, KVASNICA, M.: SIMATIC (in Slovak). DIEPC IIEAM FCFT STU, 2001.

[7] KROKAVEC, D., FILASOVÁ, A.: Diagnostic of the dynamic systems (in Slovak), Elfa, Košice, 2007.

[8] MATUŠŮ, R., PROKOP, R.: Single-parameter tuning of PI controllers: from theory to practice. In 17th IFAC World Congress, Soul, Korea, 2008.

[9] SIEMENS: Standard Software for S7–300 and S7–400 PID Control. Siemens User Manual, Siemens Automation Group, 1996.

[10] TAN, N., Kaya, I.: Computation of stabilizing pi controllers for interval systems. In 11th Mediterranean Conference on Control and Automation, Rhodes, Greece, 2003.

[11] VÍTEČKOVÁ, M., VÍTEČEK, A.: Automatic control fundamentals 2nd edition (in Czech), VŠB TU Ostrava, Ostrava 2008

[12] ZÁVACKÁ, J., BAKOŠOVÁ, M., VANEKOVÁ, K.: Robust PI controller design for control of a system with parametric uncertainties. In 34th Int. Conf. SSCHE, Tatranské Matliare, Slovakia, CDROM 038p, 2007.

**Ing. Katarína Vaneková**
**Doc. Ing. Monika Bakošová**
**CSc., Ing. Jana Závacká**

Slovak University of Technology in Bratislava
Faculty of Chemical and Food Technology
Institute of Information Engineering Automation and Mathematics
Department of Information Engineering and Process Control
Radlinského 9
812 37 Bratislava, Slovak Republic
e-mail:katarina.vanekova@stuba.sk,
monika.bakosova@stuba.sk, jana.zavacka@stuba.sk

**Ing. Radek Matušů, Ph.D.**

Tomas Bata University in Zlín,
Faculty of Applied Informatics
Department of Automation and Control Engineering
Nad Stráněmi 4511
76005 Zlín, Czech Republic
e-mail: rmatusu@fai.utb.cz

# About Grid Supported Learning Environments and Collaborative Virtual Control Laboratories

**Christian Schmid**

**Abstract**
In this article, grid technologies are introduced to build e-learning environments for virtual laboratories to be used in control education. Service-oriented grids open new fields of applications, the Learning Grids. The learning services concept and their deployment through grid technologies are excellent means to integrate virtual laboratories into collaborative e-learning environments for control engineering education. An introduction into this area, a review of the grid techniques and example applications from a virtual laboratory demonstrate grid based solutions.

**Keywords:** Collaboration, Engineering education, Learning Grid, Learning systems, Virtual laboratory.

## 1. Introduction

The current generation of electronic learning (e-learning) solutions has adopted the rather narrow pedagogic paradigm of information transfer, which features the teacher as someone who selects particular pieces of information and makes them available to students on the Web. However, there is no evidence that this approach to technology enhanced learning is in anyway effective. It has been adopted simply because it is an easy way to use the Web's facilities. As soon as the learner tries to mediate the personal knowledge with the others, to compare the results, and finally tries to use the synergy between personal and collective construction of knowledge, he has to fall back to conventional communication channels. The current solutions mainly offer textual, or conference tools at the most.

Remote or virtual laboratories with real or simulated experiments are becoming accepted in the engineering community for providing distance education and for augmenting traditional laboratories. Students have to modify instruments for a better understanding of the principle on which the plant operates. They even have to set their personal or within groups negotiated conditions in an environment for collaborative experimenting. From a pedagogical point of view, in this kind of environments the student has an active and central role in the learning process. In keeping them at the centre of the learning process, personalisation, individualisation and collaboration become relevant aspects to be supported by technologies through the creation of the right context. A learning grid can contribute to the achievement of these objectives through the definition of the learning services concept and their deployment through grid technologies.

In the following an introduction into grid computing and into the Virtual Control Laboratory (VCLab) is given. Section 2 describes the transition from computing to service-oriented grid and describes its main properties. Section 3 introduces into the specific properties of Learning Grids and Section 4 goes into details of the implementation of main features of a Learning Grid using VCLab examples. Section 5 extends this on the area of collaborative learning in a grid supported environment and Section 6 summarizes the results.

## 2. What is the Grid?

The term grid is used here as a paradigm of the commonly known concept of power grids, where the consumer is not aware where and how the power is exactly produced. He only receives the final product with a defined quality of service from the plug in the wall. In case of a computational grid, the client receives the computational power not knowing where it comes from and what the resources are. When it comes to a service-oriented grid, the user receives the functionality he needs with the desired quality of service.

Historically the term grid has been used describing a worldwide communication infrastructure for clustered computers, the nodes, that allows seamless transparent access to data and computing power on demand to solve large-scale computational problems. Such computing grids cost a fraction of what a supercomputer costs. They are commonly known from engineering, science and commerce. Grid is also a new paradigm for the information technology. The well known World Wide Web will be succeeded by the upcoming World Wide Grid. In this context a new type of grids, the service-oriented grids find applications in quite new areas not previously considered as the environments for a grid. An example of such a new area is education. This is the topic addressed in this article.

### 2.1 Grid basics

From a general point of view, a grid is considered as a collection of clustered computational machines, the nodes. In order to have a powerful supercomputer by a grid the computational problem has to be split into slices and assigned to these nodes. Each node processes its slice individually and after the completion of its slice the results are put back together. Grid nodes do not need to be placed in one geographic location; moreover, machines collaborating in the grid may have different architectures and operating systems. It is obvious that these nodes need to communicate with each other based on some standards. Therefore a vital topic of security is involved for the interchange of data between nodes. Depending on the application the data should be kept confidential and protected from undesired external changes. Also other issues must be addressed, e.g. redundancy of nodes, quality of service and scalability.

A grid shows some limitations and has to fulfil some requirements. The grid is applicable only for tasks that can be easily split into smaller slices and that do not require the characteristics of a real-time challenge. In order to reduce the complexity of a grid, a special layer is introduced that is for gluing the nodes on a logical level. This layer of software sandwiched between the operating system and the applications is commonly called middleware. Its spectrum ranges from execution environments responsible for the management of processes on nodes, to full development environments. What traditional grids lack, are the standards on that they are built. In most cases when considering computational grids, the methods of communication, the level of integrity between nodes and the architectures are each specially designed for a particular project.

### 2.2 Service oriented grids

During recent years a new approach for building grids has emerged. Instead of perceiving the grid nodes only as computational elements of an infrastructure they became providers of services [1]. This shift, from strict computational capabilities to service suppliers, opens new fields of applications for grids. The nodes, instead of only delivering their computational and storage capacity, are now regarded as providers of particular services. They may be parts of some code existing in multiple instances allowing the parallelization of the execution of an application. The nodes may offer individual services best suited to their own capabilities. Moreover, services developed for the usage in one application or grid may be reused in new applications. The service-oriented approach has additional advantages. It introduces well-defined standards, allows the creation of searchable catalogues of services. Further details are described in Section 2.3.

Figure 1 presents basic interactions between elements of a service-oriented grid. Services published into a Grid Registry are queried and when discovered then instantiated depending on the user request. Mainly for sake of efficiency the client’s communication with the service is direct but may also be virtualized.

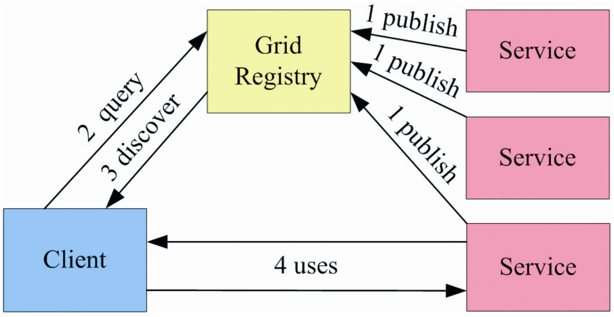


**Fig.1 Structure of a service-oriented grid environment**

### 2.3 Technology and standards

The realization of service-oriented grids needs clear standards to have that interoperability of grid elements and their reuse in other applications. The two main organizations involved in standardization of grid technologies are the Open Grid Forum [2] favouring the family of the Open Grid Services Architecture [3] standards and to some degree the competing Organization for the Advancement of Structured Information Standards [4] promoting the Web Services Resource Framework [5] standards. Both organizations adopt the currently widely recognized Web services and their extensions as their building blocks. These families of standards differ in the depth of the middleware integration, in the choice of the platform and in the programmatic languages of the implementation. But their general approach towards the grid is the same.

The main functionalities delivered by the middleware of a service-oriented grid are:

Location – allows the determination, whether the required service exists and at which locations it is accessible.

Instantiation – allows the instantiation of the service on that host, which matches the capabilities required for the service running with a given quality of service.

Orchestration – allows the dynamical composition of more complex services.

In the examples shown in this paper, the middleware called GrASP [6] is used, which was developed in an EU funded project. It follows the OASIS recommendations based on the implementation of the WSRF called WSRF.NET, which uses Microsoft’s .NET Framework as the implementation environment [7].

## 3. Learning Grids

Grids yield significant benefits to applications. The question to be answered here is what advantages may yield a grid particularly to educational systems.

### 3.1 Learning Objects (LOs) and Units of Learning (UoLs) in a grid environment

In the concept of using LOs the learning content is split into reusable elements. These elements are used to build complex learning resources. In the world of service-oriented grids the LOs are becoming fully functional services with their own user interface. They are independently interoperable blocks, which may be used as they are, or, moreover, are reused to build new more complex blocks using other grid services, e.g. orchestration. LOs themselves can be nested. For illustration consider the complex LO example from Figure 2. Delivering a nested LO for an experiment several components are necessary and each of them is implemented as a separate LO. The required components would be: the LO rendering the experiment environment, the LO displaying an Excel

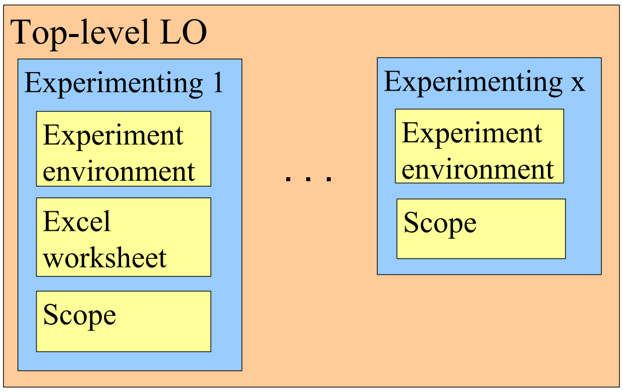


**Fig.2 Example showing an arbitrary nesting of LOs**

worksheet for evaluating results and a scope LO displaying the experiment signal histories. These components would be embedded into another LO, therefore constituting a new composed unit called e.g.: Experimenting 1, which itself could be nested in a more general LO. Due to the well-defined grid standards, the learning courses can be built from LOs delivered by different grid services. The grid techniques offer the capabilities of cataloguing and easy managing LOs by using metadata.

Metadata for describing LOs and ontologies for the semantic modelling of the learning domain can be used to build and execute distributed learning applications on a Learning Grid. They take the form of UoLs. Each UoL is described by an ontology, which defines the set of concepts to be taught. Each concept corresponds to a Learning Object (LO) that constitutes the learning material. The ontology describes the order in which the LO will be delivered to achieve the teaching goal for a learner. Each UoL realizes a certain learning model, which is modelled by the ontology. The LOs represented by UoLs and combined with the learner specific requirements are being delivered personalized by a suitable IMS-LD player [8].

### 3.2 Collaboration and communities

The use of a common platform allows a better collaboration, both in sense of interpersonal communication for collaborative learning, as well as collaboration between applications existing within the grid. A Learning Grid is a natural environment for its participants to create virtual learning communities for collaborative experimenting. All participants belong to the same community of grid users sharing the same tools, creating and sustaining professional relationships through time.

### 3.3 Scalability

An outstanding advantage of a Learning Grid environment is the approximately linear scalability inherited from its predecessor, the computing grid. When the number of students enrolled to a particular course gets larger, more instances of a particular service will be created on the hosts within the grid. When additional hosts are needed they do not have to belong to the same university or run the same operating system as long the services are implementing the same interface. Figure 3 illustrates the characteristics of the average response time observed when only one node or n nodes are available on the grid related to a particular service. Grids may grow from few resources to millions. In principle, there is no restriction in size, but the availability and latency of resources must be observed.

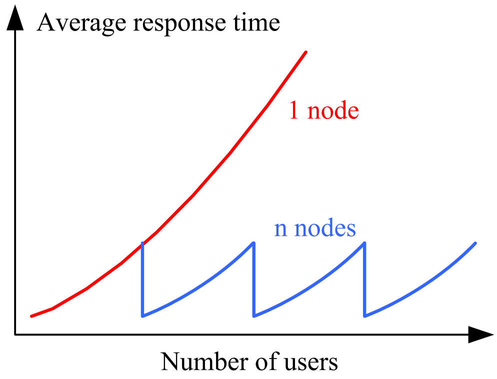


**Fig.3 Average response time characteristic depending on the number of users of a particular service available from one or more than three nodes on a grid**

### 3.4 Personalisation

A very important feature of a Learning Grid is the fact that it can deliver learning contents from heterogeneous resources in a unified fashion and personalised according to the profile of the learner. The following procedure is in analogy to the power grid. A learner with a well-defined profile introduces himself to the Learning Grid and requests some contents relevant to his learning needs. The Learning Grid starts here to find the best suitable service for the learner's needs. This would match closely as possible the user's profile taking in account the user's location, language skills, level of advancement in selected topic, preferred form of delivering content, etc.

### 3.5 Virtual Organizations (VOs)

One of the main advantages of the new grid technologies is their capability to integrate heterogeneous environments to an abstract entity. This property can be used to group resources of different universities to build a VO, e.g. a virtual university. Such an approach would allow specialization of universities in concrete areas and sharing the best offer with other universities.

## 4. VCLab as a Grid Supported Virtual Laboratory Environment

The VCLab [9] has been originally developed as a generic tool to support students in control engineering using professional design and simulations of process automation. It uses a 3D virtual user environment to recreate and to visualize experimenting plants. One can interact with a displayed scene in a similar fashion like with real devices. The dynamical behaviour of the plant is generated by a simulator driven by simulation models. VCLab has in its repository the generic components and services necessary for building LOs for experiments on a Learning Grid of the control engineering domain. The grid supported implementation of VCLab is an appropriate case study to present the introduced properties of service-oriented grids in a practical manner to the control community.

### 4.1 Architectural perspective of the grid implementation

VCLab is deployed on the ELeGI grid [10] by IWT [11], which is a content and user management system and includes a portal. For learners registered through its portal, it provides a personalized profile, which must be compatible with the IMS Learning Design specification [8]. These data describe the learner as well his preferences regarding the social context and learning styles. From the technical point of view, the profile is a standardized data structure. It is being used in two ways. First, it allows making a better choice regarding the content that should be provided to the learner; second, it also allows adjusting the selected content depending on the learner preferences.

The VCLab components are integrated as LOs into the IWT framework by using drivers. The procedure of the instantiation of VCLab components as LOs is presented in Figure 4. When the learner decides to take part in a course that uses VCLab resources, the portal receives a request to create the instance of the particular LO. The portal combines the original request with data withdrawn from the learner's profile and submits the created request document to the grid, in this instance to the GrASP middleware. GrASP searches for the best service and best service provider matching the request and instantiates the driver service. The driver service returns the portlet which renders

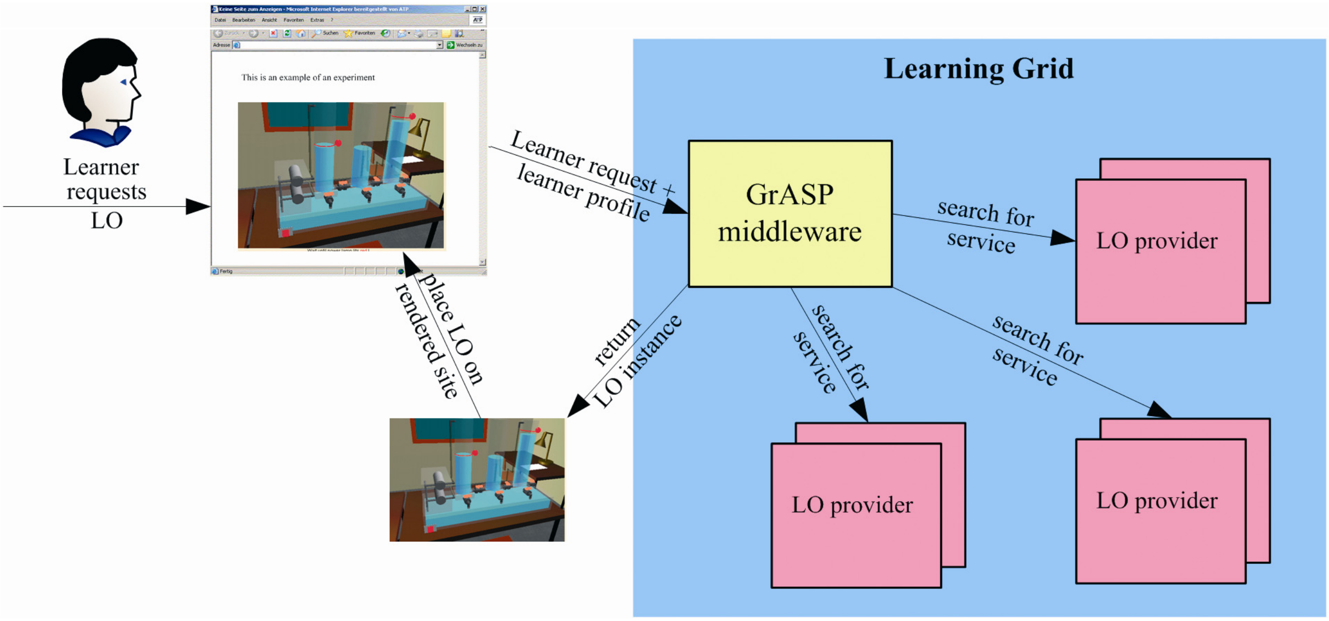


**Fig.4 Process of instantiation of a Learning Object from VCLab in a Learning Grid.**

the LO for the learner depending on the personalization definitions in the learner's profile.

In order to understand the benefits of the VCLab on the grid, the architectural perspective of implementation and the interactions between the services are presented. Figure 5 shows the services, on which the functionality of the VCLab is based and their interrelations with the elements of the grid infrastructure. VCLab consists of two services, Computation and Simulation. Their operation is orchestrated by the Driver Service, which implements that interface necessary for an e-learning application.

The Grid Catalogue plays a central role on the grid, which contains the metadata describing both, the services incorporated within the grid and the stored resources.

When a user is accessing a grid application, it identifies the user and obtains his profile. Then the application uses the Grid Catalogue to provide the list of the resources matching the learner queries and the profile. When the user chooses a UoL to work with, the e-learning application contacts the Grid Catalogue, which provides the list of services required to deliver the resource to the user. In the case of VCLab such a service is the Driver Service shown in Figure 5, which implements a particular interface understood by the e-learning application. Such a driver service takes the control over the execution of the resource and requests the required services from the catalogue. In the VCLab case these are the Simulation and Computation Service within a Simulation LO.

This grid architecture shows a high degree of separation of services during the implementation and favours the extensibility of the application itself. The e-learning application needs not to be aware of the services used by VCLab; it only needs to find a suitable driver for a particular LO to be delivered.

### 4.2 The Learning Model for Experimentation

Achieving a successful integration is not only performed on a technological level, but also in regard to pedagogical aspects. For this purpose a generic model for the delivery of virtual experiments has been developed, which can be applied to control engineering experiments. This model splits a UoL into four macro phases: Presentation, Practical Situation, Abstract Situation and Institutionalization phase, see Figure 6.

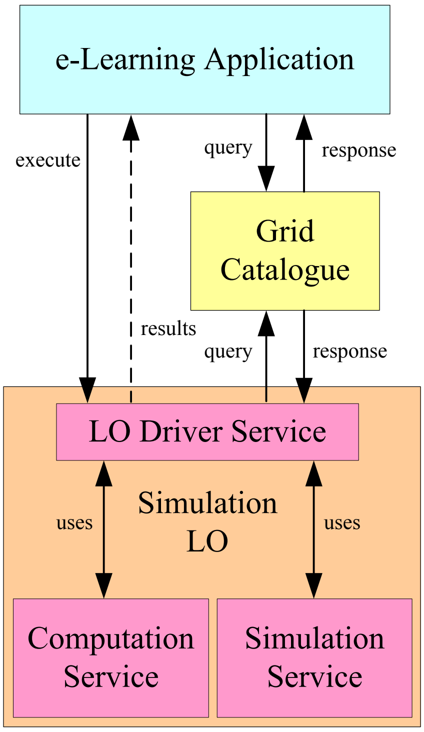


**Fig.5 VCLab grid architecture**

The phase of Presentation provides the description of the didactic experience that the learner is about to start. To such aim, the description of the different phases, the necessary information for the learner about the character and goals of the experiment and about the general reference regarding the operation of the software will be provided.

The Practical Situation represents the phase in which the learner live the concrete experience. Simulation and the presence of a collaborative environment are available in which the personal learners' experience can be mediated from the interaction with the other learners. This phase has an iterative character and consists of five micro phases:

Active Situation – A fascinating and interactive scene in 3D is proposed, inside of which the learner will be able to move and manipulate objects. Simulations are run by a series of controls that the learner can opportunely vary, modifying in real time the behaviour of the simulation, observing its response and actively gaining personal knowledge.

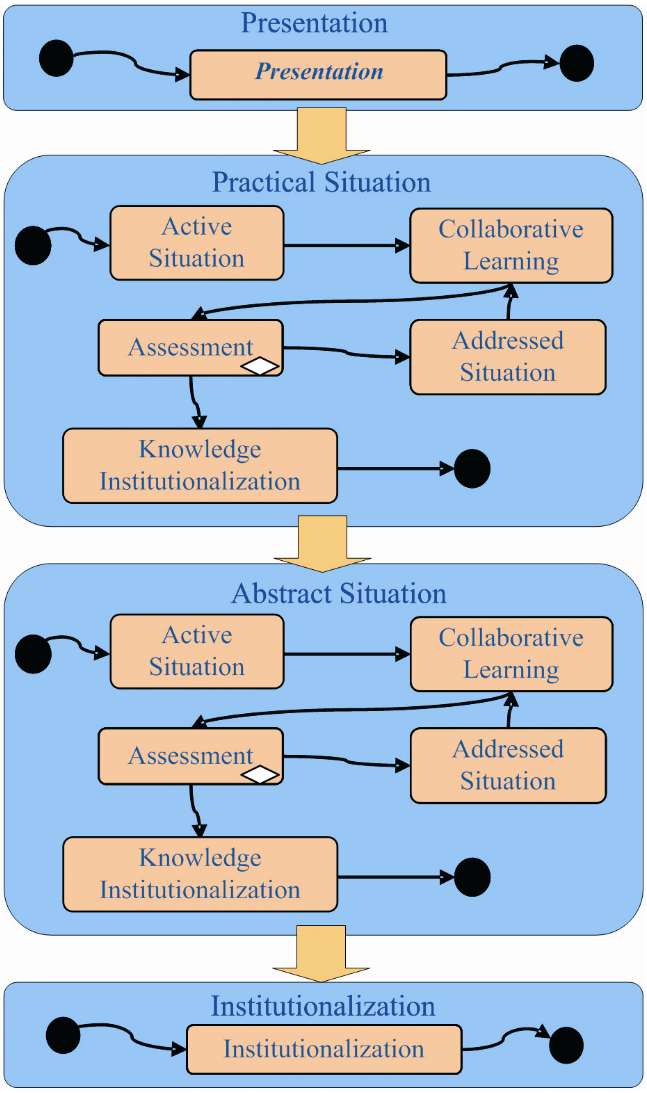


**Fig.6 Learning model for experimentation**

Collaboration – During this phase the learner has the possibility to mediate the personal knowledge with the others, to compare the results, and finally use the synergy between personal and collective construction of knowledge.

Assessment – This micro phase marks the transition from action to opinion by giving the learners a variety of questions to judge the current validity of the learning process. If the output is not adequate a possibility is offered to enter in a facilitated didactic situation, which leads to the phase of the Addressed Situation. The learner can enter again into the phase of Active Situation or Collaboration in order fill own gaps. This ends in a further assessment with a loop back if not successful. Otherwise the phase of Knowledge Institutionalization is entered.

Addressed Situation – This optional phase, to which learners may be redirected in case of an unsuccessful Assessment, may provide an altered version of the Active Situation and give additional hints which should allow a facilitated understanding of the experiment.

Knowledge Institutionalization – It is the last micro phase of the Practical Situation when the knowledge validity is shown to the learner with a correct solution and a list of concepts which should be known after completing this activity.

The Abstract Situation macro phase is to extrapolate from the previously context an abstract model. It consists of the same micro phases as the Practical Situation and its execution is governed by the same rules. But instead of the simulation of a concrete case the activities will be set up on a greater interaction between theory and practice to induce the learner to test knowledge in order to achieve new goals. For example instead of a 3D scene in the Active Situation, e.g. the learner has to deal with a set of equations describing the experiment.

Finally the macro phase of Institutionalization provides the means for organizing and formalizing the acquired knowledge.

### 4.3 UoLs for VCLab

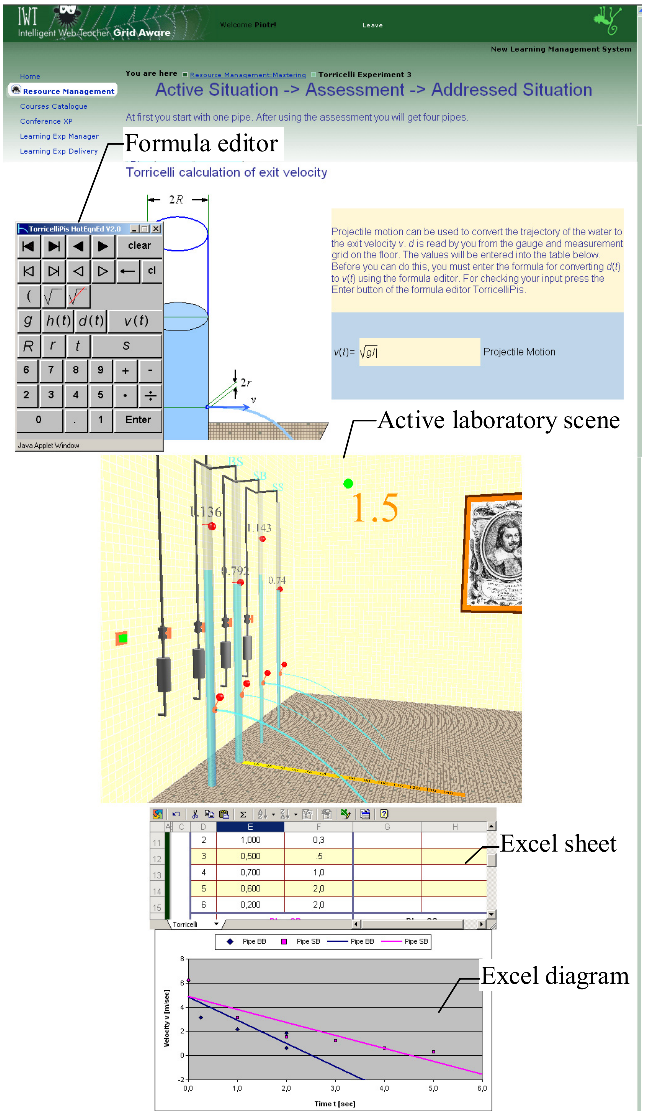
Implementing this learning model using the LO and UoL paradigm allows the building of a library of reusable learning units. Each object described by metadata is being easily catalogued and can even be dynamically bound for learning content delivery. UoLs are described by an ontology, which defines the set of concepts to be taught. Each concept contains LOs that constitute the learning material. In a situation, when a teacher has to explain to students how process control works, the plant is described by a LO. The teacher has a manuscript and makes a storyboard with items to be taught. To achieve a successful teaching process those items are structured according to phases of the learning model presented above.

The content managed by the IWT portal uses IMS-LD standardized forms of UoLs. This introduces an abstract layer over the technological aspects of resources in which authors can create their learning content in pedagogical instead of technical terms. This specification describes the recipients of a particular resource, the resource requirements in the sense of services required for executions, and the dependencies between resources.

For VCLab each UoL is described by an ontology, which defines the set of concepts to be taught. Each concept corresponds to a LO that constitutes the learning material. The macro phases from Section 4.2 are delivered as a single LO or as a series of LOs. The ontology describes the order in which the LO will be delivered to achieve the teaching goal for a learner. Each UoL realizes the learning model, which is modelled by the ontology. The LOs represented by UoLs and combined on the fly with the learner specific requirements are being delivered personalized by the IMS-LD player.

A section of an active learning session example with LOs from VCLab is shown in Figure 7. A web browser is used and the screenshot shows a section from the beginning of a nested Active Situation LO. Other sections, like assessment and addressed situation are not shown here. The example is taken from a beginner UoL for modelling Torricelli’s law.

The middle part contains the active 3D laboratory scene LO, where four pipes of different diameter can be filled by pumps with water to a given height. In the lower part of these pipes outlet valves of different diameter can be opened to let the water flow to the floor. The learner interacts with the experiment using this scene. Measurements are taken using a tape measure and a measurement grid on the floor and using a watch clock at the top of the scene. The scene is completely animated by the simulation service. The upper part deals with the relation between the trajectory of the water outflow and the exit velocity. The learner has to specify in this LO this relation in symbolic notation. The Formula editor is for entering the formula into the small window on the right-hand side. The symbolic user input is checked by the computation service and used below in the Excel sheet LO to covert the user measurement data into exit velocity data. The Excel diagram is for illustration purposes.



Welcome Piotr!

You are here ■ Resource Management:Mastering ■ Torricelli Experiment 3

Active Situation -> Assessment -> Addressed Situation

At first you start with one pipe. After using the assessment you will get four pipes.

Torricelli calculation of exit velocity

Projectile motion can be used to convert the trajectory of the water to the exit velocity *v*. *d* is read by you from the gauge and measurement grid on the floor. The values will be entered into the table below. Before you can do this, you must enter the formula for converting *d*(*t*) to *v*(*t*) using the formula editor. For checking your input press the Enter button of the formula editor TorricelliPis.

$v(t)=$ $\sqrt{g/l}$ Projectile Motion

| | D | E | F | G | H |
|---|---|---|---|---|---|
| 11 | 2 | 1,000 | 0,3 | | |
| 12 | 3 | 0,500 | .5 | | |
| 13 | 4 | 0,700 | 1,0 | | |
| 14 | 5 | 0,600 | 2,0 | | |
| 15 | 6 | 0,200 | 2,0 | | |

**Fig.7 Screen shot section of a VCLab session**

### 4.4 VCLab grid resource

The LOs realizing the macro-phases of the learning model in a UoL are generated using the VCLab Resource. Figure 8 presents the general structure of this very generic resource. It consists of a set of optional abstract elements whereby some of elements may occur in multiple instances. Customizing this resource in a proper way will yield the different LOs of the learning model. The appearance is also fully customizable by the sequence of elements in the resource as shown for the example in Figure 9.

The *Descriptive text* element is used to deliver information describing the elements that are placed above or below of it or may be used to present additional information. From the technical point of view this element is implemented as a HTML formatted text section. It may contain also active elements, e.g. Java applets, which are not standard VCLab Resource elements.

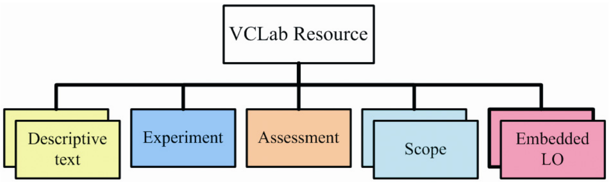


**Fig.8 VCLab Resource structure**

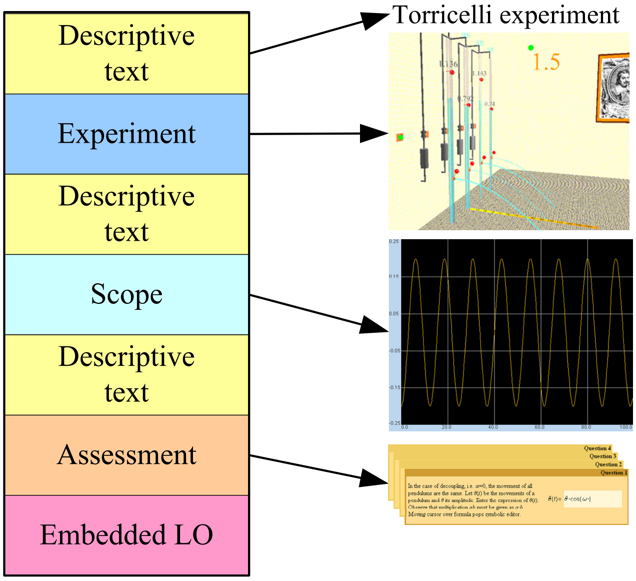


**Fig.9 Appearance of elements in a VCLab Resource**

The *Experiment* element realizes the 3D virtual user environment. It animates the behaviour of the experimental plant in the 3D scene and provides the means of altering the experiment parameters through a set of buttons or sliders rendered as 3D objects. For this purpose a 3D player is used combined with invisible controls that animates the scene and intercepts events generated by the learner. The current version of the VCLab Resource provides a LO with one experiment at one time. But as later shown this resource can be cloned and used in a collaborative manner simultaneously by many users.

The *Scope* element provides the possibility to display signal history plots in a similar manner like using an oscilloscope device. Several instances of *Scope* elements may be present showing in parallel different signals using different display modes and scaling parameters.

The *Assessment* element defines a set of multiple/single choice questions or questions that needs to be answered by entering the answer in symbolic notation, e.g. by mathematical expressions. Answers may be associated to events defining modifications of the experiment parameters.

Using the *Embedded LO* element the resource being created may also make use of already existing LOs to provide their functionality. Such an example may be LO for making notes of the measurements in an Excel sheet or a chat component for communication with other learners.

### 4.5 Authoring VCLab UoLs

Virtual laboratory models and its experiments require a large set of specifications for the related resources. Avoiding the error-prone and tedious work of authoring manually by using chains of several inhomogeneous tools an integrated authoring process of VCLab related grid resources has been developed starting from a manuscript or storyboard and supporting the process until to the final UoL.

Figure 10 presents a typical composition of a single LO, which is used to realize a Practical Situation macro-phase from Section 4.2. It contains metadata describing its content, the usability feature for the learner and the definition for the grid service to be used for its execution. It also contains the necessary data files for its execution and references to other LOs, which may be embedded within this LO.

All tasks to build a UoL are performed by using the VCLab authoring tools as shown in Figure 11. These tools are of generic type to describe the simulation model, the 3D visual representation, the assessment and to compose all resources to a UoL to be published on the ELeGI grid. In addition, these tools are itself grid resources published as grid services available for authors.

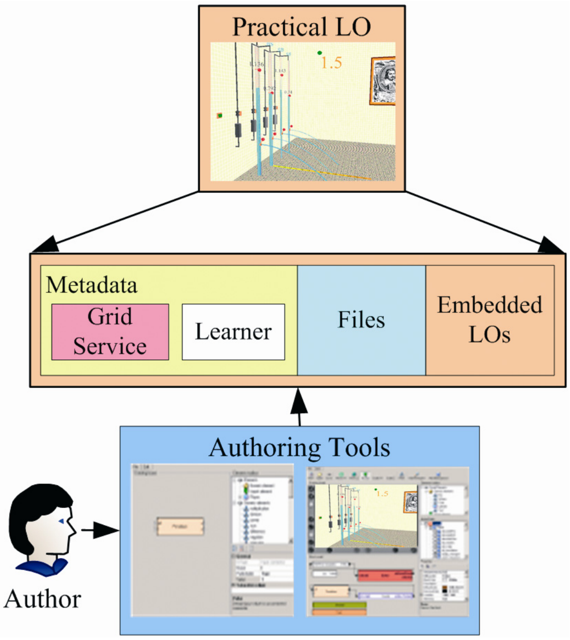


**Fig.10 A typical composition of a LO and its authoring**

The creation of the Experiment resource is more laborious and happens in two stages. First, the experiment has to be described by a simulation model in using hierarchical block structures consisting of a set of differential and algebraic equations. The simulation parameters can be defined such that they can be manipulated by the learner. The simulation results are directed for animation or plotting in real time. This stage is performed by the *Simulation Model Authoring Tool*, which contains a graphical user interface for constructing simulation diagrams.

Complex mathematical expressions may be added by embedding the HotEqn resource [12], which interprets LaTeX expressions dynamically and renders them accordingly.

Second, the *Visual Objects Authoring Tool* is used to generate the 3D model for the visual representation of the experimental plant. Then the elements of the 3D model are bounded to the inputs and outputs of the simulation model. The creation of the Assessment resource is supported also by the *Assessment Editor.* This gives direct access to visual elements, which are controlled depending on the answers to provide information for the Addressed Situation.

The implementation of the Active Situation in the Abstract macro-phase is performed by the Scope element, which is integrated with the simulation model. Signal outputs generated by the simulation model are directed to the Scope element.

At the final stage the *Resource Editor* is used to mount all the resources together into one LO and make it available for the direct deployment into the grid environment.

### 4.6 A process control example

In the following a short example of a UoL about experimenting with a thermo-fluid process of a chemical plant is shown taken from [13], which exists in reality at Ruhr-Universität Bochum, Germany, see Figure 12.

A chemical plant is described by its construction model and physical model. The construction model defines parts of the model such as a pump or a valve and the physical model defines how they interact with each other and describes their dynamical behaviour by differential and algebraic equations. As shown in Figure 13 the thermo-fluid subsystem consists of four storage tanks T1, T2, T3 and T4, two main tanks TM and TB, one reaction tank TS, waste tank TW, heaters, two pumps and several valves all together connected by a pipe system.

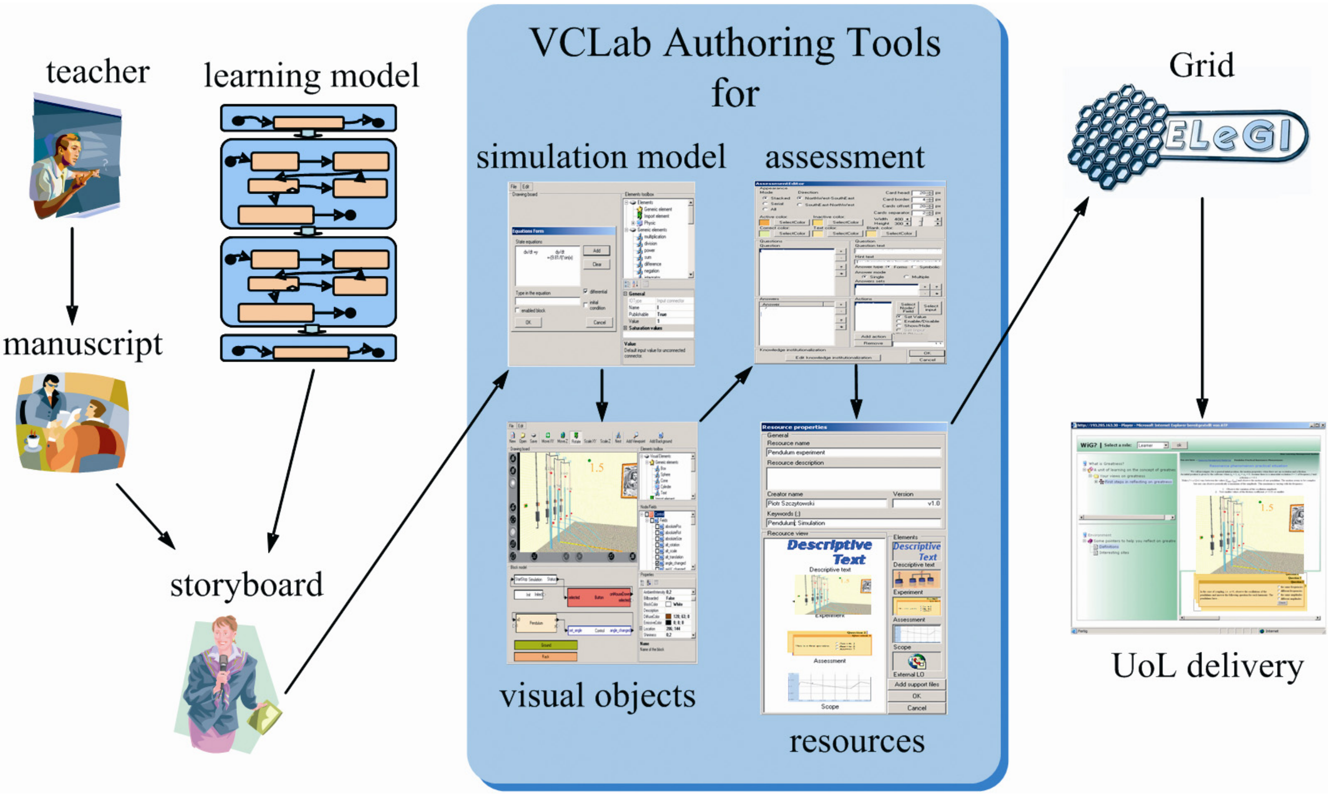


**Fig.11 From a virtual laboratory idea to a UoL.**

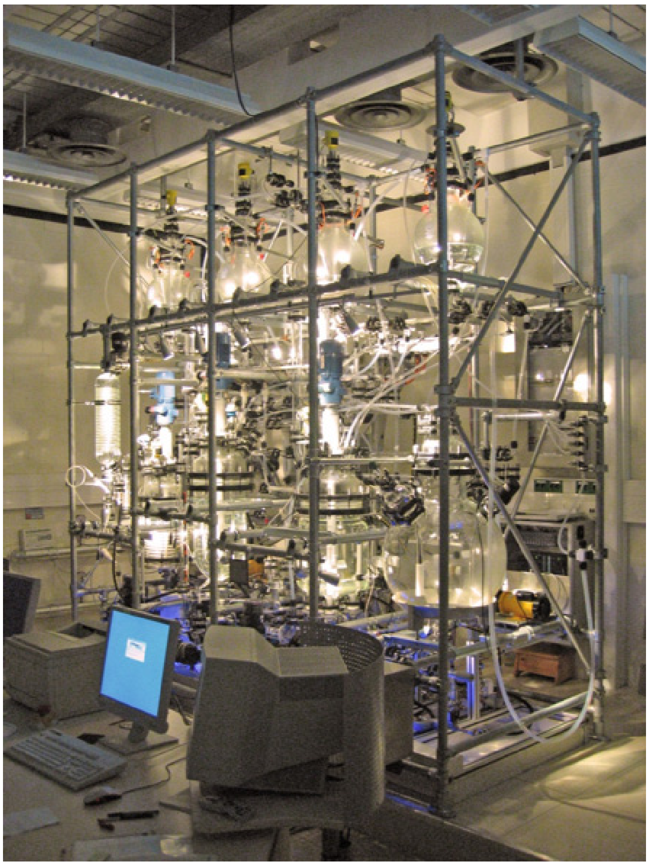


Fig.12 Chemical plant VERA at Ruhr-Universität Bochum, Germany

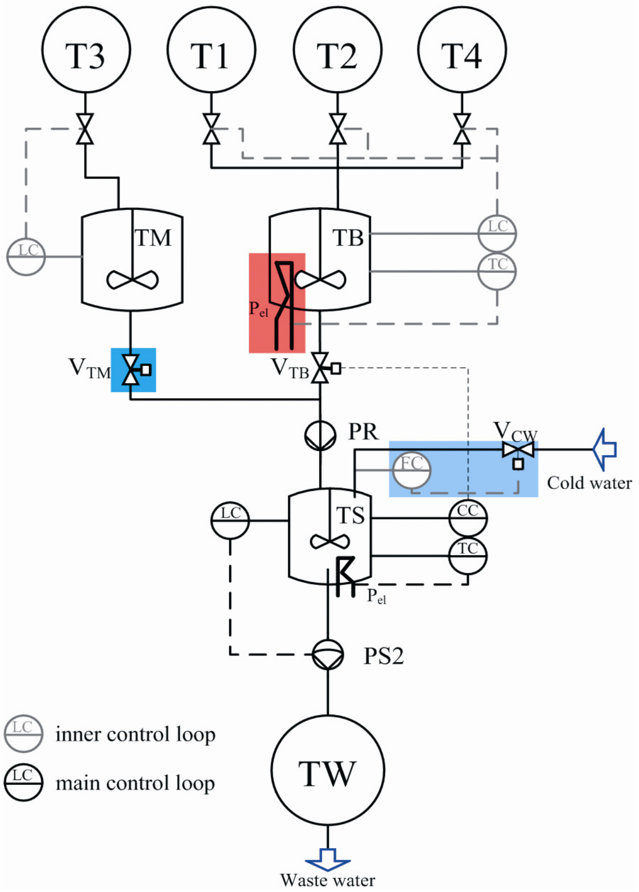


Fig.13 Flow chart of the thermo-fluid subsystem

The thermo-fluid subsystem including the control system and the physical constraints is described by a simulation model. It is built based on principles and laws for all the components as a fluid process (laminar, Torricelli flow, …) for tanks and pipes, valves and pumps. It is a dynamical system described by a system of differential and algebraic equations.

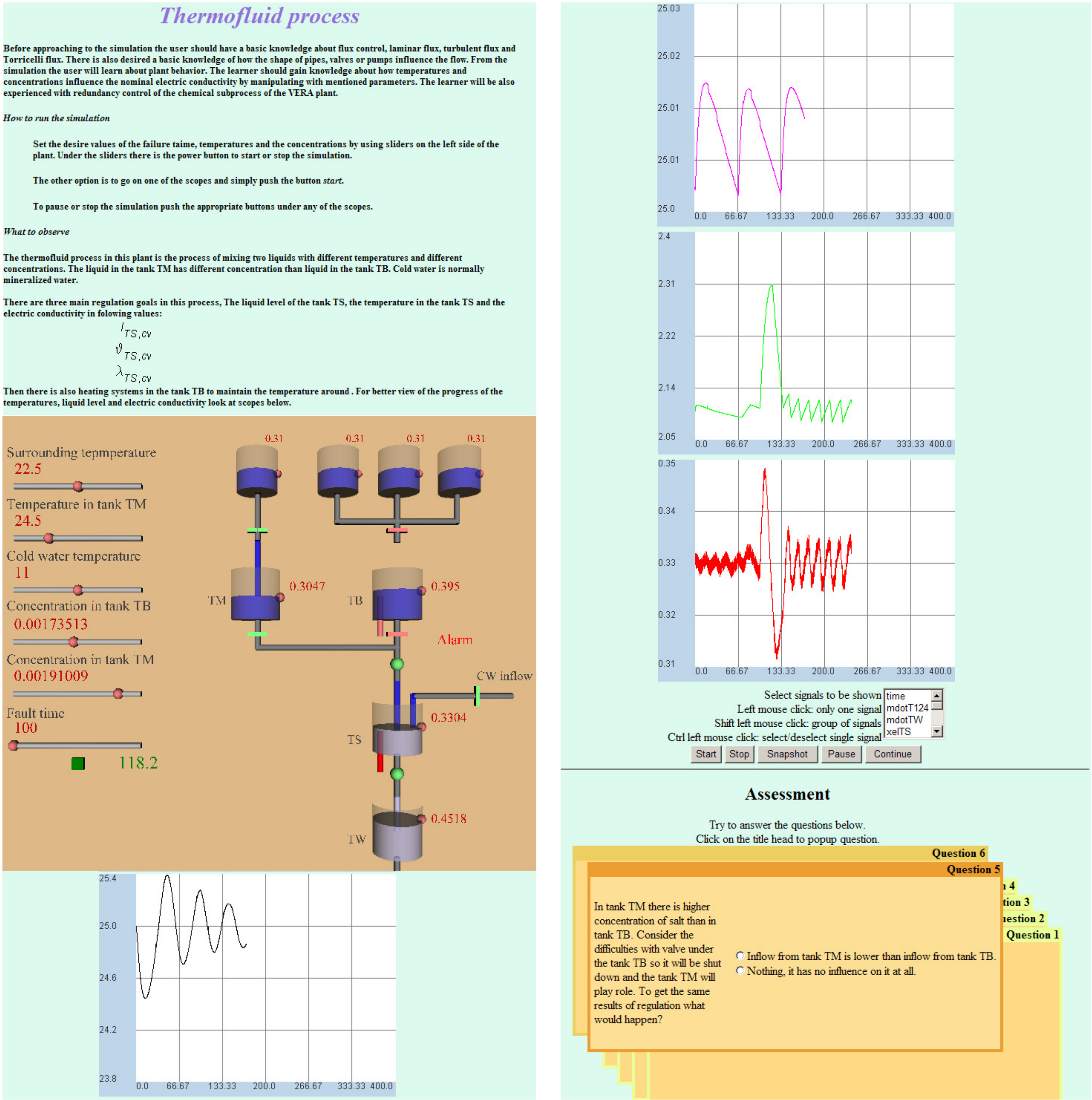
The VCLab authoring tools are used for specifying the entire simulation model, to generate the visual objects of the 3D model, to design the assessment and finally to compose the entire UoL. Figure 14 shows the screenshot of a learning session composed of several basic LOs. An active situation with a laboratory scene LO, several Scope LOs and an assessment LO with symbolic facilities are here used for exploring the dynamical behaviour of this process and to control it. The experiment flow is influenced by the assessment. Depending on the learner answers the scenarios can vary. An ontology used for building the UoL allows the easy navigation between the learning contents and assures their delivery in the right order according to the learning model from Figure 2.

### 4.7 Learner perspective of the grid implementation

From the learner perspective, the grid implementation simplifies the delivery process of the resources, in the sense that it removes the burden on the user of possessing and handling demanding software and hardware. The required hardware is always accessible; the higher complexity of the resource is compensated by the choice of a suited execution node, which is performed automatically by the grid infrastructure. The software for calculations and simulations is also always up to date because it is located on the server. Because of the server-sided execution the delivered content may be richer in details. Each LO of a UoL being executed may run on a different server. Therefore, the UoL can contain several LOs executing in parallel, what would not be possible in case of a conventional client-sided execution model. This property is especially important in case of Collaborative Learning described in detail in the following section.

It is especially important that the Learning Grid offers a personalized approach of learning in form of an activity. The learner profile carried together with the requests to the grid makes all experiences suited to the learners needs. The delivered content depends on the courses already taken and on the general advancement of the learner in a particular topic. Preferences, like language of content and its form: text, audio, video or their combination may be taken into account during delivery. As the progress made by the learner is hold in his profile, he may access the learning activities from any computer without caring for the synchronization of the context. The learner is not anymore bound to a particular workstation and his profile follows him.

Not only is the choice of the presented content affected, but also the overall quality of service. The right selection of the nodes for operation on an appropriate geographical location with respect to the learner results in faster response times. This is particularly important in case of animation, e.g. for real time experiments. Equipping the nodes with different kind of services allows a better user balancing and scalability. It is common that the enrolment to particular courses changes over time and even during semesters. The grid can adapt itself automatically to the demand for services.



Thermofluid process

Before approaching to the simulation the user should have a basic knowledge about flux control, laminar flux, turbulent flux and Torricelli flux. There is also desired a basic knowledge of how the shape of pipes, valves or pumps influence the flow. From the simulation the user will learn about plant behavior. The learner should gain knowledge about how temperatures and concentrations influence the nominal electric conductivity by manipulating with mentioned parameters. The learner will be also experienced with redundancy control of the chemical subprocess of the VERA plant.

*How to run the simulation*

Set the desire values of the failure taime, temperatures and the concentrations by using sliders on the left side of the plant. Under the sliders there is the power button to start or stop the simulation.

The other option is to go on one of the scopes and simply push the button *start*.

To pause or stop the simulation push the appropriate buttons under any of the scopes.

*What to observe*

The thermofluid process in this plant is the process of mixing two liquids with different temperatures and different concentrations. The liquid in the tank TM has different concentration than liquid in the tank TB. Cold water is normally mineralized water.

There are three main regulation goals in this process, The liquid level of the tank TS, the temperature in the tank TS and the electric conductivity in folowing values:

$$l_{TS,cv}$$

$$\vartheta_{TS,cv}$$

$$\lambda_{TS,cv}$$

Then there is also heating systems in the tank TB to maintain the temperature around . For better view of the progress of the temperatures, liquid level and electric conductivity look at scopes below.

Surrounding tepmperature 22.5

Temperature in tank TM 24.5

Cold water temperature 11

Concentration in tank TB 0.00173513

Concentration in tank TM 0.00191009

Fault time 100

Select signals to be shown
Left mouse click: only one signal
Shift left mouse click: group of signals
Ctrl left mouse click: select/deselect single signal

Assessment

Try to answer the questions below.
Click on the title head to popup question.

In tank TM there is higher concentration of salt than in tank TB. Consider the difficulties with valve under the tank TB so it will be shut down and the tank TM will play role. To get the same results of regulation what would happen?

○ Inflow from tank TM is lower than inflow from tank TB.
○ Nothing, it has no influence on it at all.

**Fig.14 Screenshot of an example session.**

## 5. Collaborative Learning in Grid Environment

An important aspect of learning in virtual environments is Collaboration, which is a micro phase in the experiment learning model of Figure 6. It is actually a scenario in which learning activities may benefit mostly from the Learning Grid. It is hard to implement rich collaborative environments capable of running several simultaneous experiments and supporting several groups of learners working together in parallel using the classical server centred approach. Even fastest server would very soon reach its limit in case of a large amount of simple simulations or in case of executing several complex ones. Distributed services on the grid remove these bottlenecks. This is addressed in the following.

### 5.1 Architectural perspective of grid supported Collaborative Learning

Collaborative e-learning applications start the execution querying the Grid Catalogue for a driver service that can perform collaborative experimenting, see Figure 15. The catalogue provides this application with the node hosting the service, which is suited for it and which can sustain the amount of potential users and embedded experiments offered by the environment. The corresponding Driver Service takes over the execution. It contacts the Grid Catalogue for instantiating the other Driver Services for the embedded Simulation LOs. A good granulation of the solution can be observed, mimicking the property of a conventional grid of splitting the tasks in small slices and assigning them to nodes spread on a grid.

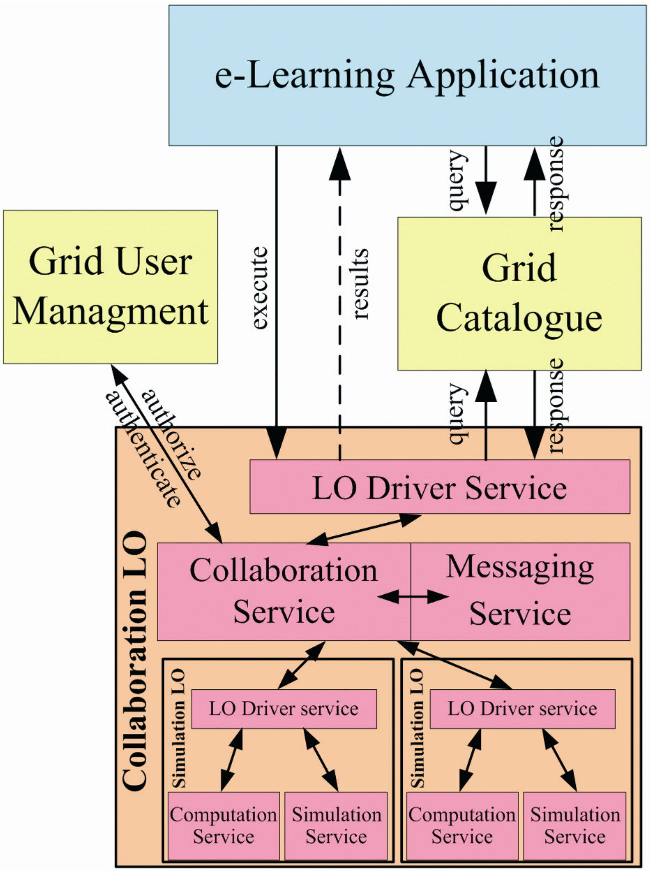


**Fig.15 Collaborative environment execution architecture**

For each instantiation of the Collaboration LO its supporting Simulation LOs may be delivered from different hosts depending on the current state of the grid and on the availability of the hosts. From the learners perspective there is no difference in learning experience what is assured by the quality of service constraints.

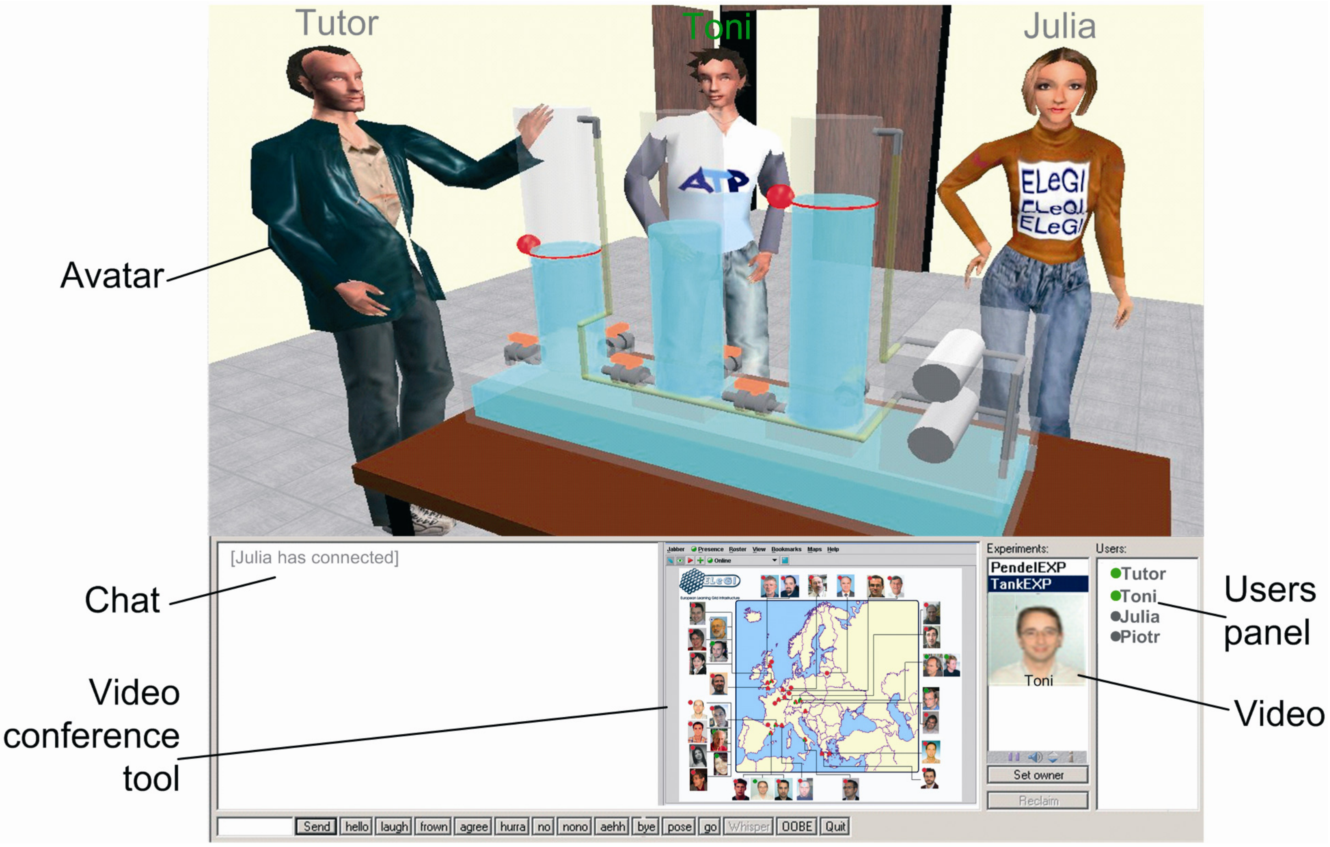
The Collaboration Service benefits from the authentication and authorization facilities of the grid infrastructure by the Grid User Management. Looking up the profile of the learner it may determine his role within the environment.

Collaborative experimenting using VCLab is based on a collaborative enhanced virtual laboratory scene as shown in Figure 16. The communication between the collaborating learners is based on two different types of information channels supported by the Collaboration Service. There is the 3D scene with the learner avatars, from where the participants can see and hear what the experimenting colleague is currently doing and to which gestures can be submitted as reactions. The other type of information channel is more classical. The Collaboration Service uses the Messaging Service. On default it is a textual chat, which may be replaced by binding another service to it providing audio or even video capabilities. This possibility is drawn from the architectural flexibility of the grid.

### 5.2 The Learner perspective of grid supported Collaboration Learning

The VCLab collaboration environment supports the full experiment learning model from Figure 6. It contains a 3D virtual scene including embedded experiments, see Figure 16. Participants of this virtual environment can be students of different universities participating simultaneously in the Learning Grid. They are represented by avatars and they communicate by means as shown above.

One important aspect of defining collaboration in e-learning is the definition of roles of participants and the assignment of these roles to them. The VCLab models four distinct roles within the collaboration environment, see Figure 17. These roles are Author, Tutor, Learner and Experimental Plant. The Author does not directly take part in the collaboration activities. Its tasks are, preparing the environment by means of defining experiments, tasks for learners, instructions, designing graphics and so on. The Tutor is a privileged


[Julia has connected]

Experiments:
PendelEXP
TankEXP

Users:
●Tutor
●Toni
●Julia
●Piotr

**Fig.16 Screenshot of a collaborative activity.**

participant of the collaboration activities. Its tasks are to provide the content to learners, to monitor their progress, to supervise experiments, to give hints, explanations and advice, to answer learners' questions, which may occur during the learning process. The Experimental Plant represents the modelled knowledge, which learners should gain during experimenting with it.

The learner role is described by a learner model. The learner is allowed to interact with the experiment in active (master learner) and in passive mode. An experimenting group may have a tutor and must have one student who is the master. The master is actively experimenting while the others are passive. The master's role can be passed to each of the participants to control the plant. Such a virtual meeting place could be the seed of creating a virtual learning community within a VO.

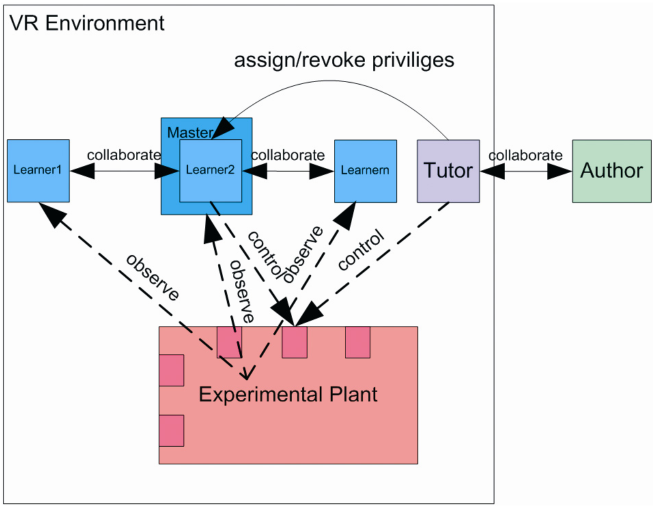


**Fig.17 Collaboration roles**

Another feature of this environment is that besides simultaneously running different experiments, a learner can dynamically plant its own experiment into the environment that he is working on. So it can be observed by fellow students and by the tutor. The Tutor can take over the control and make a demonstration, which changes will be reflected in the original learner experiment. After leaving the collaboration micro phase, the learner may continue its session regarding to the learning model. This example of dynamical binding is only possible due to the dynamic nature of the grid.

## 6. Conclusions

Learning Grids contribute to the achievements of the objectives given in the introductory chapter to this article through the definition of the learning services concept and their deployment through grid technologies. Learning services will be consumed in dynamic virtual communities based on communications and collaborations where learners, through direct experiences, create and share their knowledge in a contextualised and personalised way. This way of learning using grid resources can become now more open to learners in the engineering domain. From the 3D visual representation the learner can get the information about the plant more effectively than only from 2D scopes or only from a textual representation. The presented solution is complete and may be regarded as a proof of the concept for a Learning Grid.

The topic that is equally important but not discussed here is the authoring of the content of the grid supported Virtual Environments. VCLab provides a set of graphical authoring tools, which are self grid applications. They create 3D scenes, simulation models, the composition of them with supporting elements like Java applets, HTML text, and automating it with online assessments to produce addressed situations. The creation of collaborative environments is supported by composing previously created experiments with static elements of the environment and defining the roles of participants.

The application of grid technologies in education is of course a much wider topic than presented in this article and by the practical example of a virtual laboratory. Nonetheless the most important aspects of utilizing service-oriented grids in distance learning for engineering education are presented.

## References

[1] Foster, I. Kesselman, C. Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, vol 15, no 3 (2001).

[2] OGF: The Open Grid Forum.
Available: http://www .gridforum.org, 2009

[3] OGSA: The Open Grid Services Architecture, The Globus Alliance, University of Chicago. Available: http://www.globus.org/ogsa, 2009

[4] OASIS: Organization for the Advancement of Structured Information Standards. Available: http://www.oasis-open.org, 2009

[5] WSRF: The Web Service Resource Framework, The Globus Alliance. University of Chicago. Available: http://www.globus.org/wsrf, 2009

[6] VALLÉS, J. DIMITRAKOS, T. WESNER, S. SERHAN, B. RITROVATO, P.: The Grid for e-collaboration and Virtual Organisations, 2003, in Proc. of the eChallegnes e-2003 Conference, European Commission under the Information Society Technologies (IST) Programme.

[7] WSRF.NET: The Web Service Resource Framework and Microsoft .NET technologies, University of Virginia.
Available:
http://www.cs.virginia.edu/~gsw2c/wsrf.net.html, 2009

[8] IMS-LD: IMS Global Learning Consortium. Available: http://www.imsglobal.org, 2009

[9] SCHMID, C.: The Virtual Control Lab VCLab for Education on the Web, in Proc. of the 17th American Control Conference ACC'98, Philadelphia, 1998, pp. 1314-1318.

[10] ELEGI: European Learning Grid Infrastructure. Available: http://www.elegi.org, 2009

[11] CAPUANO, N. GAETA, A. LARIA, G. ORCIUOLI, F., RITROVATO, P.: How to use GRID technology for building next generation learning environments. In Proc. of the 2nd International LeGEWG Workshop e-Learning and Grid Technologies: a fundamental challenge for Europe, Paris, Ritrovato, P. Salerno, S. Gaeta, M Eds., Electronic Workshops in Computing, British Computer Society, 2006.

[12] HOTEQN: HotEqn – The IMGless Equation Viewer Applet. Available:
http://vclab.atp.rub.de/software/HotEqn/HotEqn.html, 2009

[13] JUHÁSOVÁ, E. SCHMID, C.: A learning unit for a thermofluid process. International Conference on Process Control PC'07, Štrbské Pleso, 2007

**Christian Schmid (Professor Dr.-Ing.)**

Ruhr-Universität Bochum
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl Automatisierungstechnik und Prozessinformatik
44801 Bochum
Germany
Tel.: ++49 234 32 24093
Fax: ++49 234 32 04093
E-mail cs@atp.rub.de

# Input shaping filters for the control of electrical drive with flexible load

**Martin Goubej, Radek Škarda, Miloš Schlegel**

**Abstract**
This paper deals with control of flexible mechanical systems. The goal is to modify the input signal in order to minimize the residual vibrations excited during a motion of a system with flexible parts. The filter is designed in the time domain via impulse function analysis. Possible application of the proposed solution is demonstrated on two examples of flexible system - control of a crane with hanging load and an electrical servo drive with attached flexible shaft. The effect of nonlinearities in the signal path caused by saturation of the servo loop controllers is studied. Various possibilities for the placement of the filter are discussed.

**Keywords:** Residual vibration control, input shaping, crane with load, flexible shaft, motion control

## Introduction

The control of flexible structure systems such as cranes or robotic manipulators introduces serious problem with residual vibrations. These motion-induced oscillations are caused by the flexible parts of the system and need to be attenuated in order to obtain precise behavior of the controlled system.

Generally, there are three possible approaches to suppress the unwanted vibrations. These include mechanical damping, active feedback control, open-loop filtering methods and their various combinations. Mechanical components such as silent-blocks or spring-damper modules can be introduced into machine design in order to increase the stiffness of the construction. However, it is difficult to predict a dynamical behavior of the machine in the phase of design; moreover, mechanical dampers are difficult to tune and mean additional expenses. Closed-loop active damping methods can achieve very good results because of the feedback, which suppresses nonlinearities of the system and uncertainty in the mathematical model ([5]), ([3]). The main disadvantage of this approach is the necessity of feedback sensors, complicated controller design and higher computational cost. On the other hand, the open-loop filtering methods use relatively simple algorithms to modify the input commands in the feed forward path in such a way that the resulting input signal led to the system does not excite the unwanted transient and residual oscillations. The advantage is simple design and absence of feedback sensors on the plant. The main drawback is the reduced robustness against uncertainty in the system model resulting from open-loop approach.

This paper deals with the last mentioned approach and uses so called Zero Vibration filter (ZV filter) ([6]) for command shaping of flexible structure systems ([2]), ([1]), ([7]). The theory is applied to the problem of the motion control of electrical servo drive with attached crane with load or flexible shaft.

## 1. Input shaping filter design

A general n-pulse input shaping filter can be described in the form of impulse function:

$$IS(t)=\sum_{i=1}^{n} A_i\delta(t-t_i),\ \ 0\le t_1 < t_{i+1}, A_i \neq 0 \qquad (1)$$

Where $A_i$ means amplitude of the i-th pulse and $\delta$ is dirac function with $t_i$ time shift

Response of the shaper in time domain can be determined by convolution with continuous input signal:

$$\begin{aligned} v(t) &= \int_{-\infty}^{\infty} h(\tau)u(t-\tau)d\tau \\ &= \int_{-\infty}^{\infty}\left(\sum_{i=1}^{n} A_i\delta(\tau-t_i)\right)u(t-\tau)d\tau \qquad (2) \\ &= \sum_{i=1}^{n} A_i w(t-t_i) \end{aligned}$$

It can be seen, that the filter has the form of sum of time delayed values of the input weighted by coefficients $A_i$.

The original input command is convoluted with the input shaper and the resulting signal is then led to the controlled system. This situation is illustrated in Fig. 1.

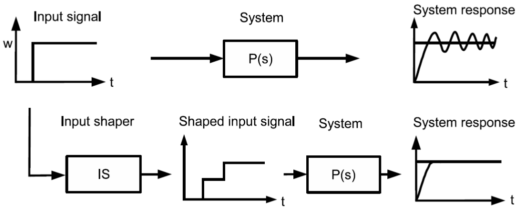


**Figure 1: Responses of the systems without and with ZV filter.**

The goal of the filter design is to choose the values of amplitudes $A_i$ and time delays $t_i$ such that after the last pulse has been led to the system, the amplitude of excited residual vibrations is equal to zero.

The design procedure will be illustrated for 2-pulse ZV filter and second-order system.

Consider linear system described by transfer function

$$P(s)=\frac{\omega^2}{s^2+2\xi\omega s+\omega^2} \tag{3}$$

with impulse function

$$h_p(t)=\frac{\omega}{\sqrt{1-\xi^2}}e^{-t\xi\omega}\sin(\omega_d t) \tag{4}$$

and two-pulse shaper with impulse function

$$IS(t)=A_1\delta(t-t_1)+A_2\delta(t-t_2) \tag{5}$$

The impulse response of serial connection of $IS(s)$ and $P(s)$ has the form

$$h(t)=A_1h_p(t-t_1)\cdot 1(t-t_1)+A_2h_p(t-t_2)\cdot 1(t-t_2) \tag{6}$$

where $1(t)$ is Heavids function. For time $t>t_2$, it holds

$$\begin{aligned} y(t) &= A_1h_p(t-t_1)+A_2h_p(t-t_2) \\ &= \frac{\omega}{\sqrt{1-\xi^2}}e^{-\xi\omega t}\sqrt{C^2+S^2}\sin(\omega_d t-\psi) \end{aligned} \tag{7}$$

where

$$\begin{aligned} C &= \sum_{i=1}^{2}A_ie^{\xi\omega t_i}\cos(\omega_d t_i) \\ S &= \sum_{i=1}^{2}A_ie^{\xi\omega t_i}\sin(\omega_d t_i) \\ \psi &= \arctan\frac{S}{C} \end{aligned} \tag{8}$$

It can be seen, that for minimizing the level of residual vibrations after the second pulse, the following expression has to be fulfilled

$$C^2+S^2=0 \tag{9}$$

By substituting (8) to (9), we obtain a nonlinear equation for $A_i$ and $t_i$, $i=1,2$. With proper choice of values

$$t_1=0,\ t_2=\frac{\pi}{\omega_d} \tag{10}$$

the equation can be reduced to the condition

$$\left(A_1-A_2e^{\xi\omega\frac{\pi}{\omega_d}}\right)^2=0 \tag{11}$$

Next, we get the second condition because of a requirement of the unit static gain of the filter

$$A_1+A_2=1 \tag{12}$$

By solving the algebraic equations (11) and (12) we get parameters of the filter:

$$\begin{aligned} A_1 &= \frac{1}{1+K},\ A_2=\frac{K}{1+K} \\ K &= e^{-\frac{\xi\pi}{\sqrt{1-\xi^2}}} \\ t_1 &= 0,\ t_2=\frac{\pi}{\omega_d},\ \omega_d=\omega\sqrt{1-\xi^2} \end{aligned} \tag{13}$$

It can be shown that those values are valid also for the systems with transfer function in the form

$$P(s)=\frac{\tau s+\omega^2}{s^2+2\xi\omega s+\omega^2} \tag{14}$$

For the proper function of the ZV filter, the exact value of natural frequency and damping coefficient of the oscillatory part of the system has to be known. The error in the system model results in non-zero residual oscillations. If the model of the system cannot be determined exactly, more robust version of the shaper can be designed by adding additional condition.

$$\frac{\partial}{\partial\omega}\left[C^2+S^2\right]=0 \tag{15}$$

The resulting three-step shaper is so called Zero Vibration Derivative (ZVD) filter and achieves less sensitivity with respect to parameter variations at the cost of slower setpoint response of the system.

## 2 Crane with load

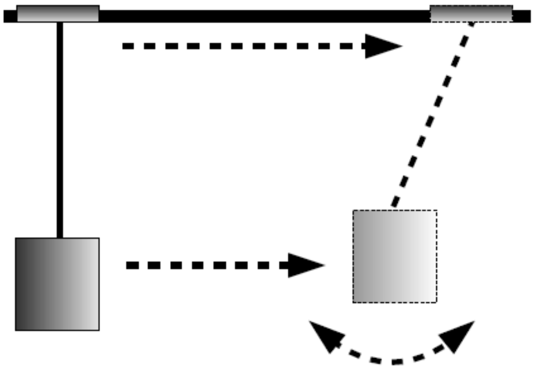


**Figure 2: Schematics of the crane and vibration of its load.**

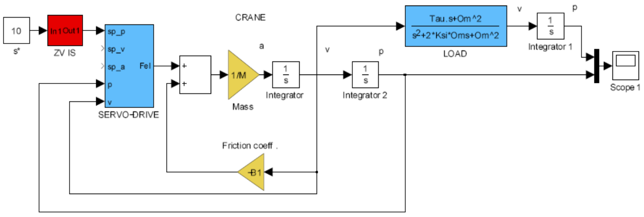


**Figure 3: Schematics of the crane with electrical drive in Simulink.**

Consider an electrical driven cargo crane with hanging load (Fig. 2). Such a system is depicted in block diagram in Fig. 3. It consists of electrical part of the drive with position and velocity feedback and an oscillatory second order system described by transfer function

$$P(s)=\frac{V_l(s)}{V_c(s)}=\frac{\tau s+\omega^2}{s^2+2\xi\omega s+\omega^2} \tag{16}$$

where $\omega$, $\xi$ are natural frequency and damping of the system, $V_c$ is the speed of the crane, $V_l$ is the angular speed of the hanging load

The system (16) describes the relationship between motion of the crane and its load. The forces acting on the crane caused by the movement of the load are omitted with respect to the ratio of its mass.

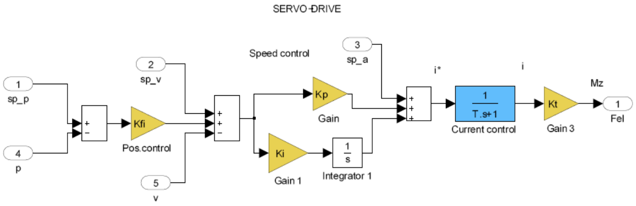


**Figure 4: Schematics of the electrical servo drive in Simulink.**

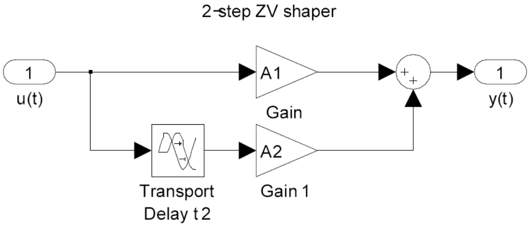


**Figure 5: Schematics of 2-step ZV filter in Simulink.**

The subsystem of the drive consists of three control loops in commonly used cascade structure (Fig. 4). The dynamics of the current loop is modeled as the first order system, the speed and position control is realized using standard PI and P controllers. The servo system contains also the feed forward inputs for planned trajectory following. A human operator sets the desired position of the crane $s^*$. Without using an input shaping filter, the motion of the crane induces the residual vibrations of the load (Fig. 6). The swinging load can hit some obstacles and an operating personnel has to wait before the oscillations damp out and they can continue with a manipulation. Using the ZV input shaper (Fig. 5), the operator command leading to the servo drive is filtered (Fig. 7) and the resulting movement does not excite the oscillatory dynamics of the load (Fig. 8).

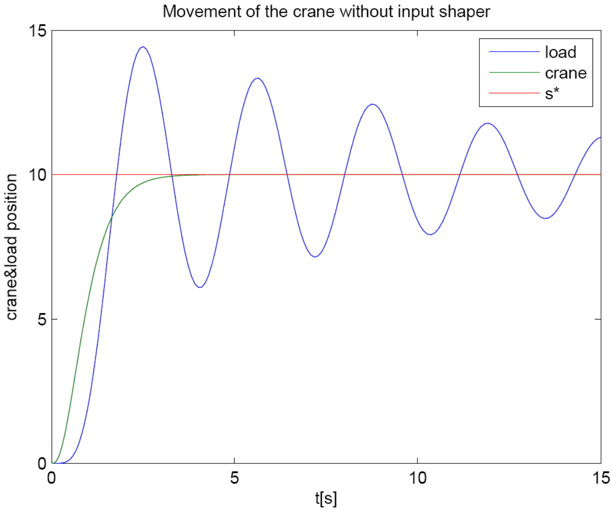


**Figure 6: Movement of the crane without ZV-filter.**

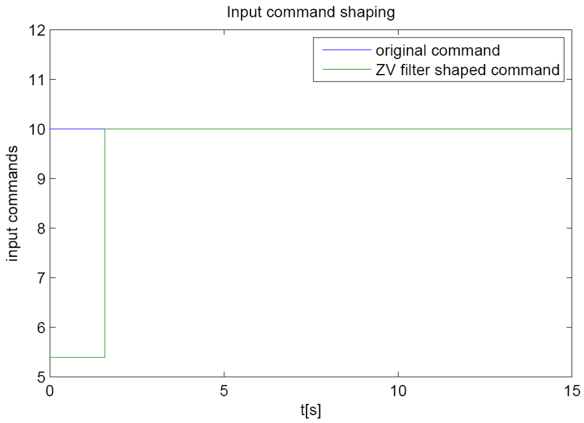


**Figure 7: Comparison of the original and shaped command signal.**

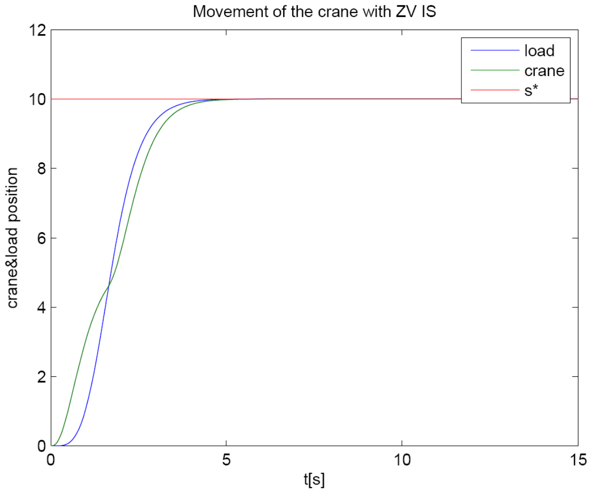


**Figure 8: Movement of the crane with ZV-filter.**

The great benefit of the proposed technique is, that shaper design can be made for the second order system of the swinging load and its filtering properties stay unchanged even after passing through the dynamics of the servo drive. However, this presumption is valid only in the case, that all the control loops of the drive work in the linear mode. If any of the controllers hits the saturation limits, the original filtered command signal is corrupted and the result is non-zero amplitude of the excited residual vibrations. This situation is depicted in Fig. 9, where saturation has been placed to the current control loop. It can be seen, that the vibrations has not been canceled out completely due to the nonlinearity in the drive control loop.

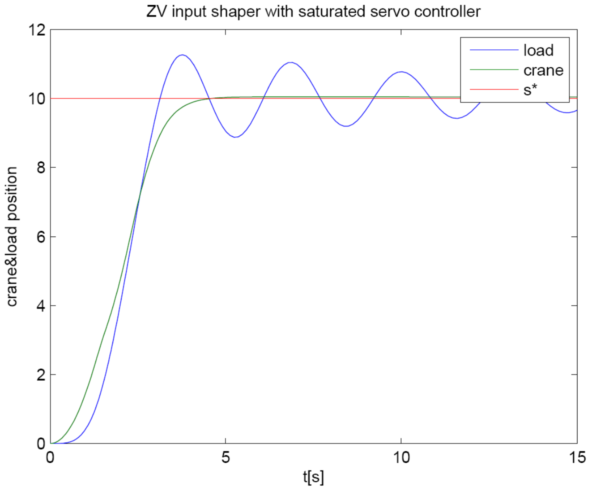


**Figure 9: Movement of the crane with ZV-filter and saturated servo controller.**

A possible solution of this problem is to add an superior system for trajectory planning. After the human operator sets a new desired position, this block computes the time optimal trajectory for the rest-to-rest movement with respect to limitations for velocity, acceleration and deceleration (optionally also with constraint for derivatives of acceleration and deceleration). The signals of desired position, velocity and acceleration led to the feed forward inputs of the servo controllers ensures the correct tracking of the planned trajectory and also prevent the saturation effect. This configuration is displayed in Fig. 10.

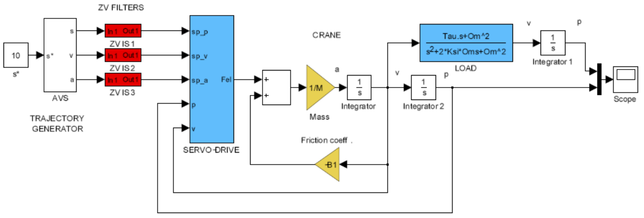


**Figure 10: Schematics of the crane with electrical drive and t-optimal trajectory generator (block AVS) in Simulink.**

The block AVS computes the desired trajectory, which is led to the servo drive controllers. The input shaper design remains the same, the filter has to be placed before all of the servo control inputs. Response of the system can be seen in Fig. 12. There are no residual vibrations even in the presence of nonlinearities in the servo control loop. All the controllers work in the linear mode, because of trajectory planning block. The resulting trajectory of the movement is no more time-optimal, nevertheless, it is suboptimal with respect to the demand for attenuation of swinging load vibrations. The important notice is that the filtered trajectory signal does not violate the default constraints for velocity, acceleration and deceleration due to the unity gain of the filter. Figure 11 shows the difference between t-optimal and real curves of position, velocity and acceleration. The presented example can be reduced easily to the case, where only the velocity of the crane and load should be controlled.

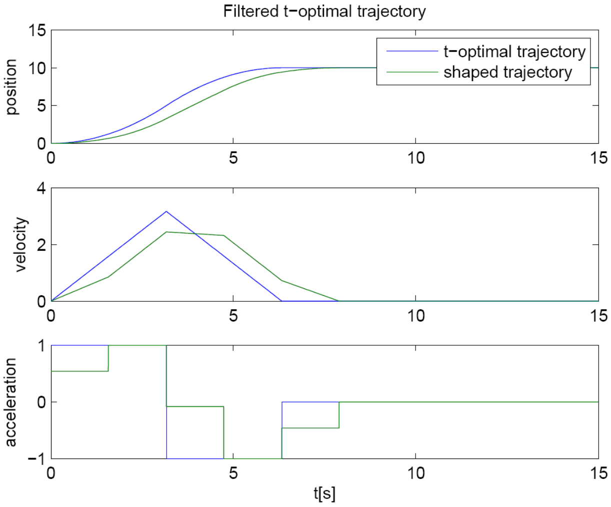


**Figure 11: T-optimal and ZV filtered trajectory for the crane.**

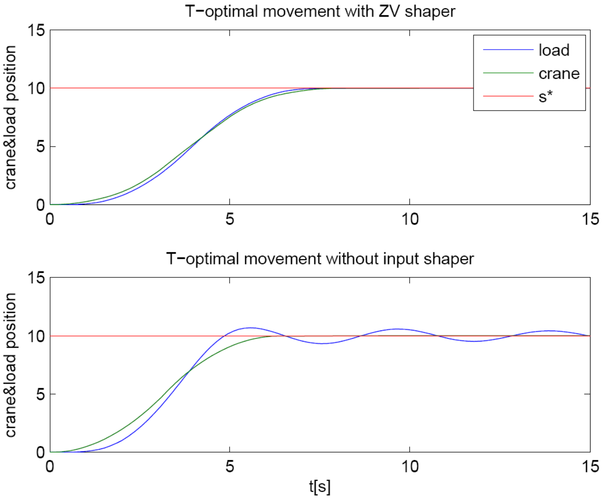


**Figure 12: Comparison of the time-optimal and time-optimal ZV filtered trajectory of crane.**

## 3 Flexible shaft

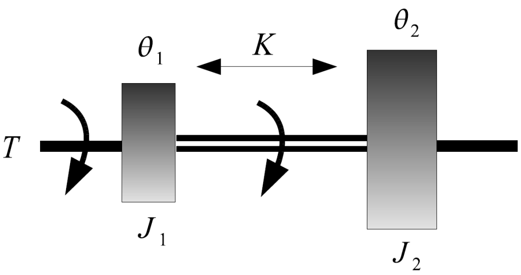


**Figure 13: Schematics of the flexible shaft.**

Consider a system consisting of electrical drive from the previous example and attached flexible shaft (Fig. 13). Dynamics of the system can be described by the set of two equations

$$\begin{aligned} J_1\ddot{\theta}_1 + B_1\dot{\theta}_1 &= T - K(\theta_1 - \theta_2) \\ J_2\ddot{\theta}_2 + B_2\dot{\theta}_1 &= K(\theta_1 - \theta_2) \end{aligned} \qquad (17)$$

where

$J_1, J_2$ are moments of inertia of the motor and the shaft

$B_1, B_2$ are coefficients of damping

$K$ is torsion spring constant

$T$ is torque

$\theta_1, \theta_2$ are angular displacements of motor and the shaft

The goal of the control system is to control the position of the shaft without exciting the residual vibrations on the flexible end. Here, the design of the filter is not so straightforward as in the previous case. The back propagation of the torque from the end of the shaft to the drive can not be omitted analogously to the crane. The entire system including the drive and flexible feedback has to be analyzed in order to find the frequencies, which have to be attenuated.

Transfer function between setpoint command $s^*$ and angle of the flexible end of the shaft can be derived from the block scheme (Fig. 14). The dynamics of the current loop can be omitted, because the electrical time constant of the drive is

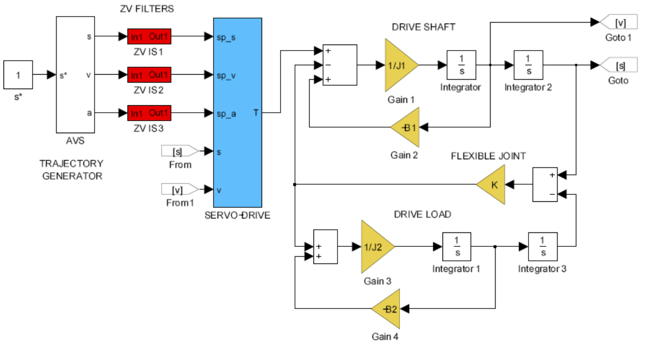


**Figure 14: Schematics of the flexible shaft with the electrical drive and ZV filter in Simulink.**

negligible with respect to mechanical time constant of the whole system.

$$\frac{\theta_2(s)}{s^*(s)} = \frac{num(s)}{den(s)} \qquad (18)$$

where

$$num(s) = KK_\theta K_t K_p \cdot s + KK_\theta K_t K_I$$

and

$$\begin{aligned} den(s) \; &= J_1 J_2 s^5 + (B_2 J_1 + J_2[K_t K_p + B_1])s^4 \\ &+ [KJ_1 + B_2(K_t K_p + B_1) \\ &+ J_2(K_t K_I + K_\theta K_t K_p + K)]s^3 \\ &+ [K(K_t K_p + B_1) + J_2 K\theta K_t K_I \\ &+ B_2(K_t K_I + K_\theta K_t K_p + K)]s^2 \\ &+ [K(K_t K_I + K_\theta K_t K_p + K) \\ &+ B_2 K_\theta K_t K_I - K^2]s + KK_\theta K_t K_I \end{aligned}$$

where

$K_t, K_P, K_I, K_\theta$ are parameters of controllers

$s^*$ is position setpoint

From the obtained transfer function, the location of the poles can be analyzed. For the typical configuration of the parameters, where servo control loops are set to achieve stable setpoint response without overshoot and presumption of $J_2 > J_1$, the resulting system has two pairs of oscillatory poles and one stable real pole. In most cases, only the slower pole pair lying closer to the imaginary axis needs to be canceled out by the filter. The natural frequency and damping can be easily computed from the location of the poles. Next, the ZV filter can be designed. If the second pair of oscillatory poles still causes unacceptable level of vibrations, a second ZV filter for faster poles can be added and serial-connected to the first one.

The problem with nonlinearities in the control loop due to saturation of servo controllers remains the same and also the solution is identical to the crane-load problem. For the proper function of the filter, trajectory planning block should be added to satisfy the demand for the linear function of the servo loop controllers.

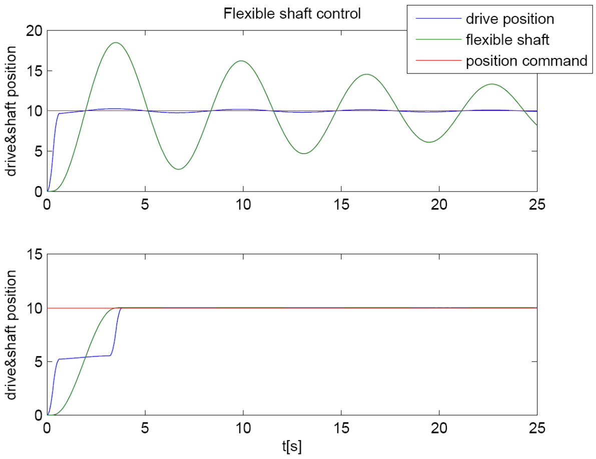


**Figure 15: Comparison of the flexible shaft movement without and with ZV filter.**

Figure 15 shows the results of flexible shaft control. Figure on the top illustrates the control without the input shaper, the lower one with the ZV filter. Using the filter, the vibrations has been completely canceled out.

## 4 Shaping filter in the closed loop

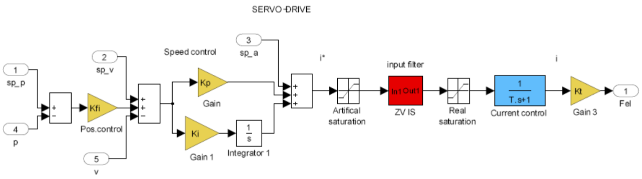


**Figure 16: Schematics of servo loop controllers with ZV filter.**

The input shaping filter can be placed also in the closed loop of the servo drive. The most suitable position for the filter is the speed controller loop (Fig. 16). The advantage of this approach is, that the saturation effect of the position and speed loop does not affect the filtering properties of the shaper. For every real electrical drive, the current loop contains saturation limits due to the limited supply voltage of the power inverter. Therefore, the value of the maximal current that the inverter is able to deliver to the motor is limited. This maximal value changes with actual rotational speed of the drive because of the back electromotive-force acting against the power supply. For the proper function of the filter, an artificial saturation block should be placed before the filter (Fig. 16). The saturation limits should be chosen in such a way that the output of the filter never hits the limits of the real saturation of the current loop. This ensures that the frequency band of the signal remains unaffected and the resulting motion of the drive does not excite the vibrations of the flexible load. Next, there exist some studies indicating the ability to suppress the measurement noise ([2]) .

The main drawback is that the insertion of the filter introduces an additional dynamics, which can affect the behavior of the closed loop. For the large natural frequencies, which have to be attenuated, the delay of the filter can be small with respect to the dynamics of the drive. Smaller values of the natural frequency mean slower filter, which has to be taken into account while tuning the loop controllers. The effect of the filter inside the closed loop is illustrated in the (Fig. 17). The figure on the top represents the movement of the crane with load without the input shaper. The drive controllers are tuned in order to achieve fast setpoint response without position overshoot. After introducing the filter to the closed loop, the overall dynamics of the drive has changed and the result is the oscillatory

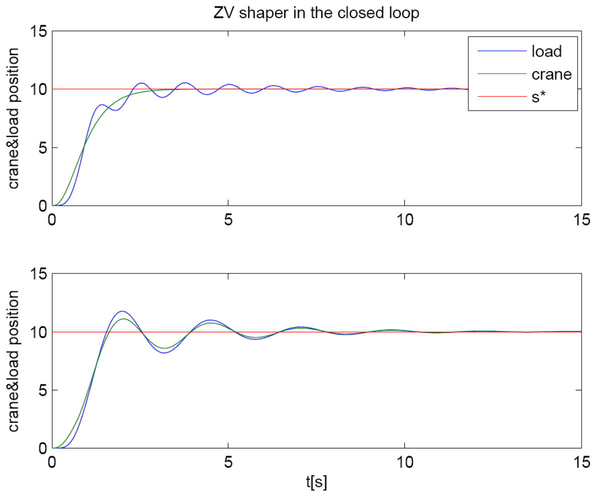


**Figure 17: System response a) without ZV filter b) with ZV filter in closed loop.**

movement of the crane as displayed in the lower figure. Even if the drive does not excite the residual vibrations of the hanging load, the controllers need to be re-tuned in order to achieve desired setpoint response. The integration of the filter inside the closed loop can cause even the unstable dynamics. It can be assumed that by adding an additional delay to the system, the overall dynamics of the closed loop become worse.

## 5 Conclusion

This paper presents the applications of the input shaping filters to the control of electrical servo drive with attached flexible load. The goal is to minimize any transient and residual vibrations induced by the movement of the drive. Firstly, the Zero Vibration filter is derived. The next part presents its utilization for the control of a crane with hanging load and a drive with attached flexible shaft. The results show significant improvement of the closed loop behavior and attenuation of the unwanted oscillations. The ZV filter can be easily implemented in a real time control system and some studies show its better performance compared to conventional notch filters ([2]). In the case of model uncertainty, more robust filter versions can be designed at the cost of increasing the delay of the filter. Next, the effect of saturation of the servo loop controllers is discussed and the solution is proposed by adding a trajectory planning block. The last part deals with the possibility of placing the filter inside the closed loop of the drive. The advantages and drawbacks of this approach are discussed.

## References

[1] K-S Chen and T-S Yang and K-S Ou and J-F Yin. Design of command shapers for residual vibration suppression in Duffing nonlinear systems. *Mechatronics*, 2008.

[2] J.R. Huey and K.L. Sorensen and W.E. Singhose. Useful aplications of closed-loop signal shaping controllers. *Control Engineering Practice*, 16:836--846, 2007.

[3] J. Mertl and M. Schlegel and M. Cech and P. Balda. Aktivni tlumeni vibraci lopatek v aerodynamickem tunelu . *www.rexcontrols.com*, 2005.

[4] M. Schlegel and O. Vecerek. Samonastavujici se regulator pro velmi slabe tlumené systemy . *www.rexcontrols.com*, 2003.

[5] N.C. Singer and W.P. Seering. Preshaping command imputs to reduce system vibration. *Journal of Dynamics Systems, Measurment and Control*, 112:76--82, 1990.

[6] K.L. Sorensen and W.E. Singhose. Command-induced vibration analysis using input shaping principles. *Automatica*, 44:2392--2397, 2008.

## Appendix

### Parameters of crane with load model (16)

$\omega_n$ = 2 rad

$\xi$ = 0.05

$\tau$ = 0.05 s

$M$ = 1 kg

### Parameters of flexible shaft model (17)

$J_1$ = 1 $kg \cdot m^2$

$J_2$ = 10 $kg \cdot m^2$

$K$ = 0.1

$B_1, B_2$ = 0.1

### Parameters of servo drive controllers (18)

$K_\varphi$ = 1

$K_p$ = 3

$K_t$ = 1

$K_I$ = 0.05

**Martin Goubej, Ing.**

University of West Bohemia
Faculty of Aplied Sciences
Department of Cybernetics
Univerzitní 8
30614, Pilsen
E-mail: mgoubej@kky.zcu.cz

**Radek Škarda, Ing.**

University of West Bohemia
Faculty of Aplied Sciences
Department of Cybernetics
Univerzitní 8
30614, Pilsen
E-mail: skardar@kky.zcu.cz

**Miloš Schlegel, Prof., Ing., CSc.**

University of West Bohemia
Faculty of Aplied Sciences
Department of Cybernetics
Univerzitní 8
30614, Pilsen
E-mail: schlegel@kky.zcu.cz

# A Neuro-Fuzzy controller for a trajectory following mobile robot

**Ivan Masár, Michael Gerke**

**Abstrakt**
The design of a motion controller for a mobile robot can be a very difficult and tedious task, especially for robots with a complex kinematic structure. Even though several types of motion controllers have been proposed in literature, they are not always applicable on car-like mobile robots equipped with conventional steering wheels. The reason is, that it is generally not possible to derive an inverse kinematic model for such robots. In this paper, a self-tuning intelligent controller for a quasi-omnidirectional mobile robot is presented. The controller is used to steer the robot following a desired trajectory. It is implemented as a neuro-fuzzy controller, which can adapt its parameters by a self-learning process in such a manner, that the mobile robot can follow a desired trajectory with required accuracy and speed. The process of controller parameters tuning is demonstrated on experiments with a quasi-omnidirectional mobile robot F.A.A.K.

**Kľúčové slová:**

mobile robot control, trajectory following, neuro-fuzzy controller

## Introduction

A precise motion control is a primary assumption for a successful application of every mobile robot. During the last years, various control design methods have been developed for this purpose. The situation is relative easy for mobile robots with a simple kinematic structure (e.g. with two differential driven wheels), or for robots without non-holonomic constraints, respectively. If their inverse kinematic model exits, it can be used to calculate the position and speed of every joint in the kinematic chain from the speed of the robot body. Therefore, a motion of the robot along a trajectory can be computed offline as a motion sequence of robot joints and links. In an ideal case, after applying of these partial motions, the robot would move as required. However, there are external forces acting on the robot during its movement (driving, centrifugal, friction, Coriolis and gravitational forces), that have to be compensated to follow the desired trajectory with high precision. For this purpose, the kinematic or dynamic model of the robot is used depending on requested precision of trajectory following. For simple applications, the kinematic model is usually preferred. The dynamic model of the robot is the base for a high precision motion control, but it is considerably more complicated than the kinematic model. Various model-based control systems for mobile robots can be found in [1], [2].

However, the problem with implementation of a model-based control arises if the inverse kinematic model of the robot does not exist. Such a situation occurs by every car-like mobile robot, i.e. if there are more wheels on a common axis or at least one steering wheel with the steering axis going through the contact point of the wheel with the floor. An example of a conventional steering wheel is depicted in Fig. 1.

A motion of the robot body consists of two translations (in X- and Y-axis) and one rotation (about Z-axis). It can be expressed in the robot *body* coordinate system $B$ as

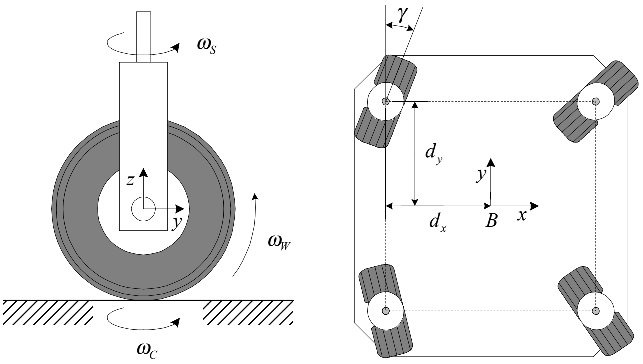


**Fig. 1 Conventional steering wheel**

$$\dot{\mathbf{p}}_B = \begin{bmatrix} v_{B_x} \\ v_{B_y} \\ \omega_{B_z} \end{bmatrix} = \mathbf{J}.\mathbf{q} = \mathbf{J}\begin{bmatrix} \omega_W \\ \omega_C \\ \omega_S \end{bmatrix}, \qquad (1)$$

where $\omega_W$ is a wheel rotational speed about X-axis (driving speed of the wheel), $\omega_C$ is a rotational speed of the wheel about Z-axis in the contact point with the floor and $\omega_S$ is a rotational speed about the steering axis. The matrix $\mathbf{J}$ is called a wheel Jacobian and in the case, that the steering axis of the steered wheel goes through the contact point with the floor, it is defined as follows

$$J = \begin{bmatrix} -R\sin(\gamma) & d_y & -d_y \\ -R\cos(\gamma) & -d_x & d_x \\ 0 & 1 & -1 \end{bmatrix}, \qquad (2)$$

where $R$ is a wheel radius and $d_x$, $d_y$ are distances of the wheel steering axis from the robot midpoint.

Obviously, the matrix $\mathbf{J}$ is singular and therefore it is not possible to derive the inverse kinematic model of the robot directly. A conventional steering wheel has less degrees of freedom than input variables and therefore it is redundant. Hence, the model-based control using the inverse kinematic model is not applicable on mobile robots equiped with conventional steering wheels.

For this reason, we have developed a neuro-fuzzy controller which can be used for trajectory following for car-like mobile robots with non-holonomic constraints. Before describing the controller functionality, a mobile robot used for verification of the proposed solution will be briefly introduced.

## Quasi-omnidirectional mobile robot F.A.A.K.

The quasi-omnidirectional mobile robot F.A.A.K. (Fig. 2) was developed on our department as an experimental platform for testing of advanced control and navigation strategies [4]. Even though it is a small-size robot, it is equipped with external sensorics, a visual system and enough processing power to implement advanced control strategies to be able to operate fully autonomous in various environments.

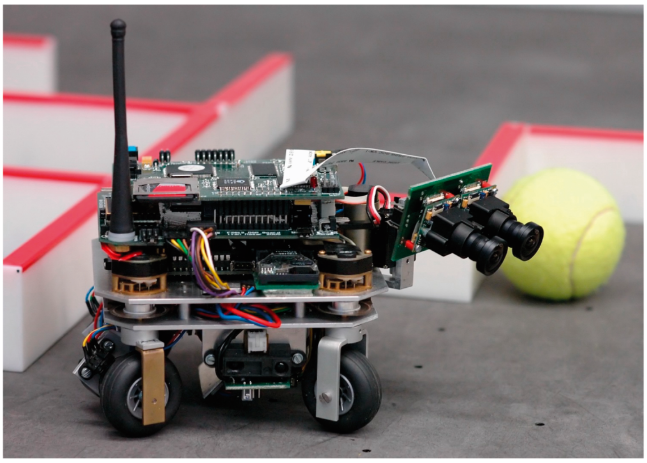


**Fig. 2 Mobile robot F.A.A.K.**

The robot is built on a non-conventional chassis with four steered wheels, whereas two of them are driven. Such kinematics allows to perform various types of motion, depending on given task and actual situation in robot environment. Thereby it is possible to change not only the radius, but also the instantaneous centre of robot rotation ICR (Fig. 3, Fig. 4) and thus to turn the curves with very small radius. Moreover, the robot can cruise, move laterally (Fig. 5) or rotate about its vertical axis (Fig. 6), respectively.

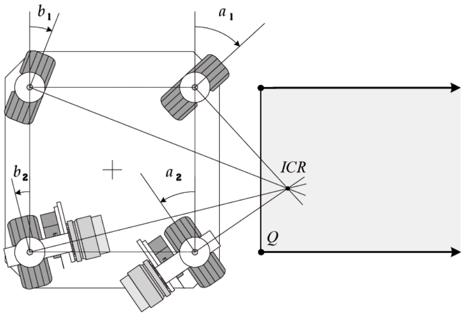


**Fig. 3 Turning with different signs of steering angles**

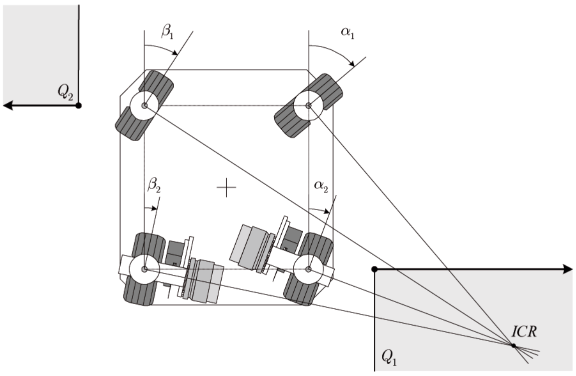


**Fig. 4 Turning with equal signs of steering angles**

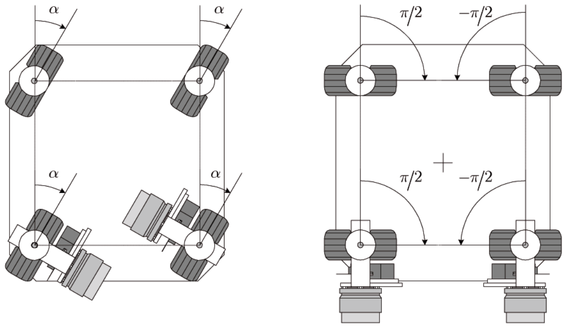


**Fig. 5 Cruising and side ride**

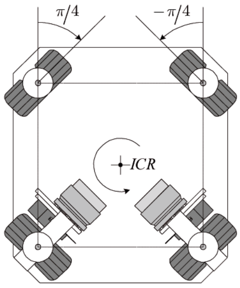


**Fig. 6 Rotation on a spot**

By using these motion types, the robot can overcome the constraints caused by the non-holonomic couplings and thus it is able to move in any direction with an arbitrary orientation. Therefore, the robot can move quasi-omnidirectional. Such a quasi-omnidirectional movement concept is very advantageous for robotic applications in artificial environments with many obstacles like households, offices, hospitals etc. Namely, thanks to the possibility to re-configure its kinematic structure, the robot can manoeuvre on a very limited space.

## Trajectory following problem

The problem of trajectory following by a mobile robot is shown in Fig. 7. Here a simplified kinematic model of the F.A.A.K. robot is used. This simplification can be applied under the assumption, that the steering wheels are controlled using Ackerman steering principle, i.e. that all perpendicular wheel axes intersect in an ICR at any time. If this condition is fulfilled, the steering wheels on a common axle can be replaced by a virtual wheel in the middle of this particular (front or rear) axle. The real robot wheels are then controlled according to Ackermann principle mentioned above. By introducing virtual steering wheels, the forward kinematic model of the robot can be reduced significantly. The control variables are then the steering angles $\gamma_1$, $\gamma_2$ of the virtual wheels and angular velocity of the rear steering wheel $\omega_2$.

obsah
articles

The position of the robot body is defined in a global *floor* coordinate system $F$ as

$$\mathbf{p}_F = \begin{bmatrix} p_x & p_y & \theta_z \end{bmatrix}^T \tag{3}$$

The trajectory following error $e_L$ is measured as the perpendicular distance of the robot midpoint to the trajectory, which is implemented using cubic splines. The control objective is to follow the trajectory with required forward speed and with minimal trajectory following error.

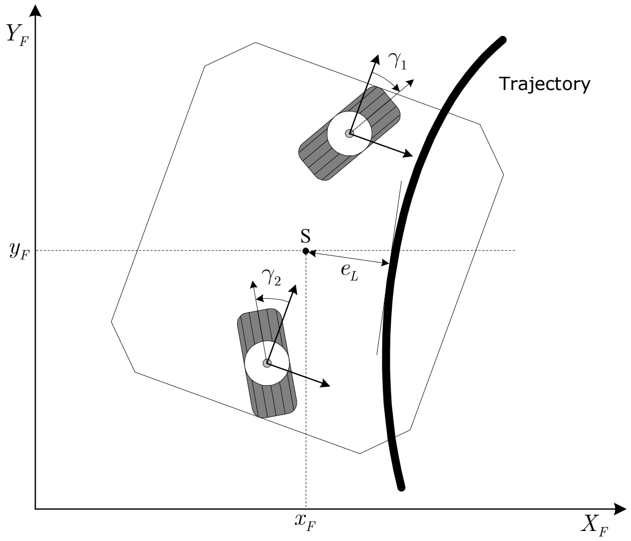


**Fig. 7 Line following problem**

## Neuro-Fuzzy Controller

As mentioned above, it is not possible to derive an inverse kinematic model for this kind of mobile robots and therefore to compute the steering angles and driving speeds of the wheels directly from the desired trajectory. For this reason, we developed a neuro-fuzzy controller to solve the trajectory following problem.

The main idea of the presented neuro-fuzzy controller is based on the fact, that it is possible to control the mobile robot following a predefined trajectory by a simple PD-controller under certain conditions. Such a PD-controller controls the steering angles of the steering wheels $\gamma_1$, $\gamma_2$ dependent on the actual distance from the trajectory. However, this controller works properly only for a specific forward speed of the robot and for a limited maximal distance from the trajectory.

To control the robot over its entire operating range, several PD-controllers could be designed. Every controller would be optimized for a specific working point defined by a forward speed of the robot. In addition, an supervisor logic would have to switch among PD-controllers according to the actual operating point. Nevertheless, such type of a switching causes well-known problems, like ejecting of control signal discontinuities and oscillations on the border between two working points, eventually instability of the control loop. Furthermore, the design of controller parameters for various operating points is not an easy task and requires many trials.

Therefore, we fused several PD-controllers with switching logic to a fuzzy logic controller (FLC), which changes the parameters of the PD-controller continuously according to actual conditions. The rule base of the controller contains the decision rules in the form (here as an example for the control of the front steering wheel):

$$r_i : \text{Wenn } e_L \text{ is } E_i \text{ and } \omega_2 \text{ is } \Omega_i \text{ then } \gamma_{1i} \text{ is } P_i e_L + D_i \dot{e}_L, \quad i = 1,\dots,N$$

where $e_L$, $\dot{e}_L$ and $\omega_2$ are input variables; $E_i$ and $\Omega_i$ are linguistic values of the input variables $e_L$ and $\omega_2$, respectively, used in the $i$-th rule. Output function implements a PD-controller with the parameters $P_i$ and $D_i$ using trajectory following error $e_L$ and its time change $\dot{e}_L$. Hence, the fuzzy controller is based on Takagi-Sugeno inference system.

The crisp output value of the control signal is computed by the singleton method. For the steering angle $\gamma_1$ of the front wheel, the output is computed from local outputs of the $N$ rules described above using the formula:

$$\gamma_1 = \frac{\sum_{i=1}^{N} \gamma_{1i}(e_L, \dot{e}_L)\mu_i(e_L, \omega_2)}{\sum_{i=1}^{N} \mu_i(e_L, \omega_2)}, \tag{4}$$

where $\mu_i$ is the weight of the $i$-th rule. Thus, the control signal is computed as a weighted average from several PD-controllers and hence without discontinuities. The fuzzy rules for the steering angle of the rear wheel $\gamma_2$ and its angular speed $\omega_2$ are implemented in an analogous manner.

The fuzzy controller with inference systems described above can be easily implemented as a structured neural network. The main advantage of such implementation is the possibility to optimize its parameters (i.e. the values of proportional and derivative gain of the PD-controllers and the parameters of the input/output fuzzy sets) by some adaptation procedure. Moreover, the parameters can be adapted continuously during robot movement by means of on-line training methods and available input-output data.

The structure of the closed control loop with the neuro-fuzzy controller for trajectory follwing is shown in Fig. 8. The proposed controller has two inputs - trajectory following error $e_L(k)$ and its one time-delayed sample $e_L(k-1)$ to compute the time change of the distance from the trajectory. The trajectory following error $e_L(k)$ is either computed from the actual robot position $\mathbf{p}_F(k)$ and predefined trajectory coordinates $\mathbf{d}_L(k)$ or directly measured by a line camera, which is mounted underneath the robot.

The outputs are steering angles $\gamma_1(k)$ and $\gamma_2(k)$ for both front and rear steering axes and the maximum forward speed of the rear wheel $\omega_2(k)$.

The tuning of controller parameters uses backpropagation learning and takes place in two steps.

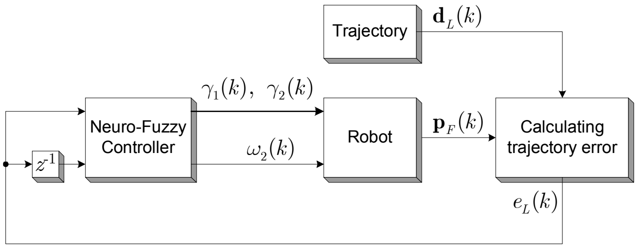


**Fig. 8 Control-loop with fuzzy controller**

### Step I - Training of dynamical model of the system

In the first phase, the dynamical model of the mobile robot following a predefined trajectory must be trained. This model is required for backpropagating of the trajectory error to the controller during an adaptation of its parameters. Because it is very difficult (if not impossible) to derive this model analytically in the form of differential equations and then backpropagate the control error through it, we used multilayer neural network instead. The inputs to this network are two steering angles $\gamma_1(k)$, $\gamma_2(k)$, angular velocity of the driving wheel $\omega_2(k)$ and the last distance of the robot from the trajectory $e_L(k-1)$ plus one delayed sample of every input variable in order to build the model dynamic, i.e. $\gamma_1(k-1)$, $\gamma_2(k-1)$, $\omega_2(k-1)$ and $e_L(k-2)$. Output of the network is the estimated distance from the trajectory $\tilde{e}_L(k)$. The structure of proposed neural network is given in Fig. 9.

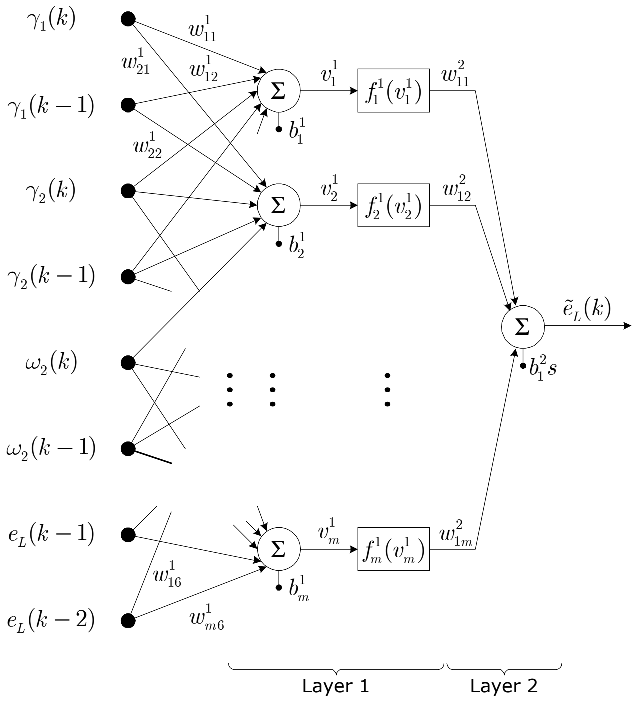


**Fig. 9 Neural network for modelling of the robot**

For learning of neural network based model of the robot, sufficient amount of training data (coordinates of the trajectory, robot position error, robot forward speed, etc.) must be recorded. Three methods of training data acquisition can be used:

- **Manual steering of the mobile robot along the desired path.** In this data acquisition mode, the line camera is used to measure the distance from the trajectory. The trajectory is represented by a contrast line draw on the floor (e.g. black line on the white floor). The mobile robot is teleoperated from MATLAB in this mode.
- **Manual steering of the simulation model of the robot.** In principle it is the same method of data acquisition as the previous one, but instead of the real robot, a robot simulation in a 3D environment running in MATLAB/Simulink is used. The main advantage of this method is the possibility to control the mobile robot also with a greater distance to the trajectory, because the distance is not limited by the measuring range of the line camera. Moreover, the motion of the robot can be viewed by the simulation from various viewpoints, including "through the robot camera lens" perspective, and the simulation can be stopped if the robot goes away from the trajectory or moves in a wrong direction.
- **Autonomous ride of the robot controlled by a PD-controller.** As mentioned above, in certain circumstances it is possible to follow a trajectory by mobile robot using a simple PD-controller. The conditions for trajectory following are limited by forward speed of the robot and maximal distance from the trajectory. Therefore, the training data sets can be recorded also during a real or simulated ride of the robot controlled by the PD-controller as long as it moves along the trajectory.

After collecting of a sufficient amount of training data, the neural network implementing robot model can be trained. For this purpose, hybrid learning [3] is very suitable because of its fast convergence and computational effort. This type of learning combines an adaptation of non-linear parameters of the system $\mathbf{\Theta}_N$ by the Levenberg-Marquardt algorithm according to equation

$$\Delta\mathbf{\Theta}_N = -\left[\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I}\right]^{-1}\mathbf{J}^T(\mathbf{e}_L - \tilde{\mathbf{e}}_L) \qquad (5)$$

and the estimation of its linear parameters $\mathbf{\Theta}_L$ using LSE-method

$$\mathbf{\Theta}_L = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{e}_L\,, \qquad (6)$$

where $\mathbf{J}$ is the Jacobian of the system, $\mathbf{A}$ is a matrix of activation functions of neurons from the Layer 1 and $\mathbf{e}_L$, $\tilde{\mathbf{e}}_L$ is a vector of output and estimated errors, respectively.

Because the learning of this neural network is a computational consumptive process, it is usually done off-line. Learning of the neural network is depicted in Fig. 10.

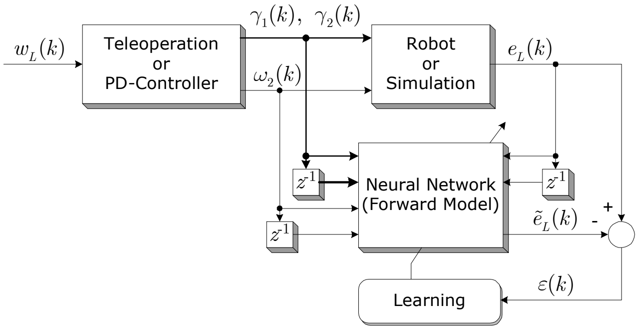


**Fig. 10 Training of the feedforward neural network**

### Step II - Training of the neuro-fuzzy controller

In the second step, the parameters of the neuro-fuzzy controller are adapted so that the mobile robot is able to follow the desired trajectory with specified tracking error and with arbitrary forward speed. For the purpose of learning, a neuro-fuzzy controller is implemented using a structured neural network. In this network, each layer represents a part of the Takagi-Sugeno fuzzy inference system.

The control loop with neuro-fuzzy controller during self-learning is depicted in Fig. 11.

The learning of the neuro-fuzzy controller can be realized again using simulations or by experiments with the real robot. However, the robot simulation is the preferred way, because it can accelerate the learning process. Moreover, the problems that can occur by experimentation with the real robot controlled by a not well-tuned neuro-fuzzy controller can be avoided. In that case, the robot always has to be stopped to prevent possible collisions. Afterwards the learning can be started again, but the training process is delayed. Good results can be achieved by combining simulations and real experiments. In this case, the parameters of the controller are adapted using simulations first. Thereafter, they are fine-tuned using experiments with the real robot.

The adaptation algorithm is based like before on the Levenberg-Marquardt concept. As measure for parameter adaptation, the sum of squared distances from the trajectory is used. Experiments have shown that it is not necessary to adjust the parameters of input fuzzy sets of the neuro-fuzzy controller, but only parameters of output functions, i.e. the gain values of the PD-controllers.

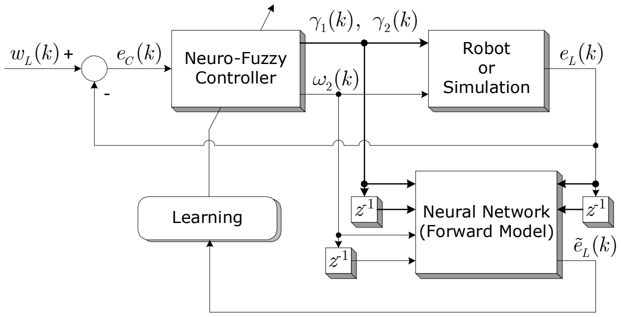


**Fig. 11 Training of the neuro-fuzzy controller**

## Experiment results

In the following graphs, the results of two experiments are presented. During experiment 1, the robot tries to find a straight line and to follow it. During experiment 2, the robot follows a curved trajectory. Start positions of the robot in both experiments are in Fig. 12 and Fig. 13.

In Fig. 14 and Fig. 15, trajectory distances from both experiments before and after learning are compared. At the beginning, a fuzzy controller with trial-and-error designed output function is used. With this initial controller set-up, the robot has a serious problem to follow the desired trajectory and during experiment 2, it diverges from the trajectory.

After 10 training steps, the trajectory following error was reduced rapidly and the robot was able to follow the trajectory without difficulites. After few more trainings epochs, the trajectory following error reached nearly the global optimum. It was not possible to reduce the line following precision further because of the kinematic and dynamic constraints of the robot.

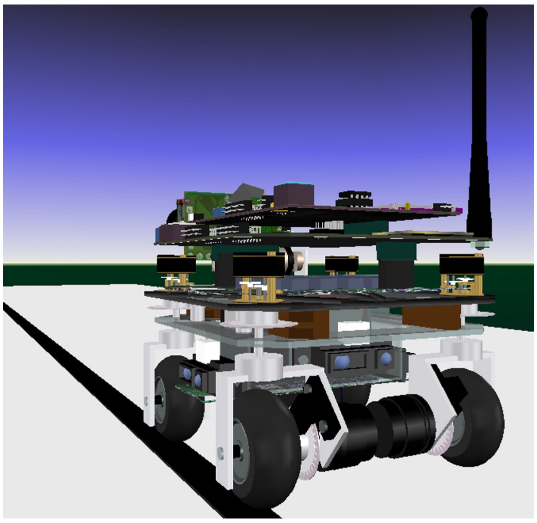


**Fig. 12 Experiment 1**

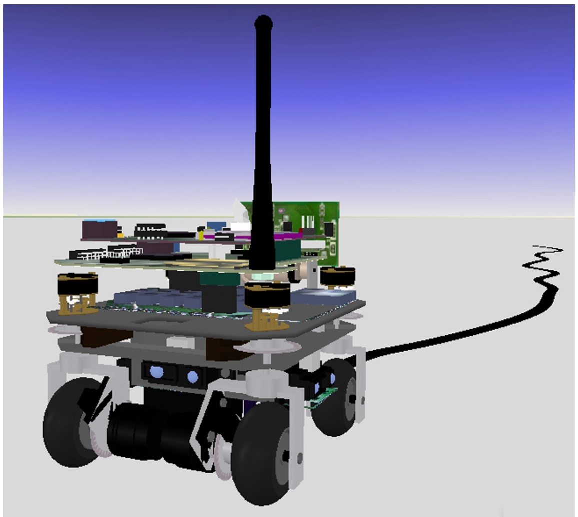


**Fig. 13 Experiment 2**

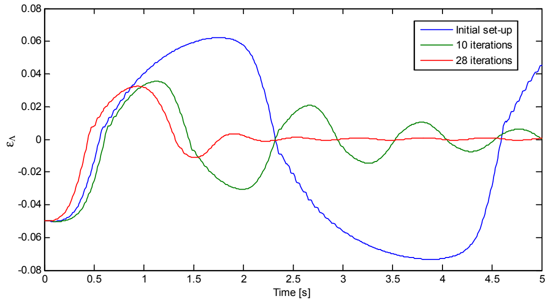


**Fig. 14 Experiment 1 – trajectory following error**

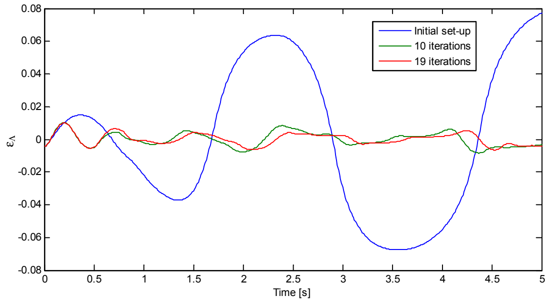


**Fig. 15 Experiment 2 – trajectory following error**

The improvement of the robot control during the learning of the neuro-fuzzy controller is obvious from Fig. 16 and Fig. 17. In these figures, the motion of the robot before and after adaptation of the controller parameters are depicted.

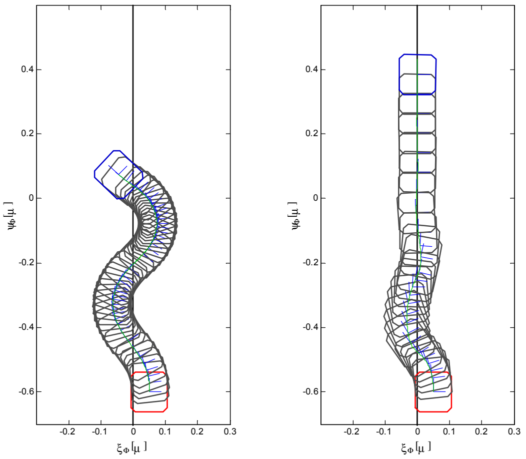


**Fig. 16 Experiment 1 – robot motion**

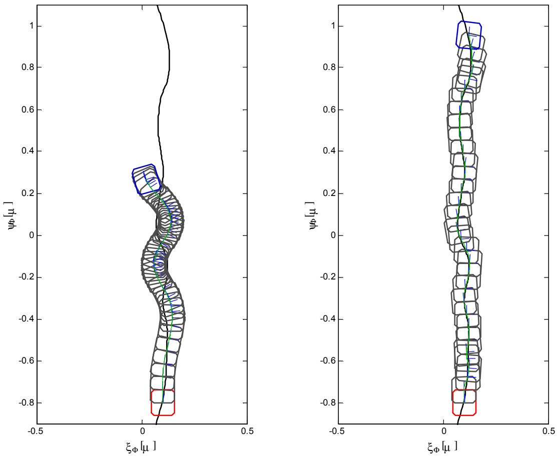


**Fig. 17 Experiment 2 – robot motion**

## Conclusion

In this paper, the problems of motion control of non-holonomic mobile robots have been analysed. Because of the absence of their inverse kinematic model, the classical model-based control methods are not applicable. As a solution we have developed an intelligent neuro-fuzzy controller for the trajectory following. The controller is able to adapt its parameters by self-learning, therefore the inverse kinematic or dynamic model of the robot are not required for its design. Moreover, the parameters can be adapted also on-line during robot movement or off-line in some regular intervals, respectively. After the learning phase, the mobile robot was capable to follow the desired trajectory with expected precision in any situation.

The proposed controller has been tested on the experimental quasi-omnidirectional mobile robot F.A.A.K. The simulation and experiment results have approved, that the designed neuro-fuzzy motion control concept is very suitable for the considered class of non-holonomic mobile robots because of its simple design and implementation.

## References

[1] CANUDAS de WIT, C., BASTIN, G., SICILIANO, B.: Theory of robot control, Springer-Verlag, Berlin, 1996

[2] MUIR, P.F.: Modeling and control of wheeled mobile robots, PhD Thesis, Department of Electrical and Computer Engineering and The Robotics Institute, Carnegie Mellon University Pittsburg, 1998

[3] JANG, J.-S.R., SUN, C.-T., MIZUTANI, E.: Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence, Prentice-Hall, Upper Saddle River, 1997

[4] MASÁR, I., GERKE, M: DSP-Based control of mobile robots, Proceedings of The European DSP Education and Research Symposium EDERS-2004, Birmingham, UK, 2004

**Dr.-Ing. Ivan Masár, Prof. Dr.-Ing Michael Gerke**

FernUniversität in Hagen
Fakultät für Mathematik und Informatik
Lehrgebiet Prozesssteuerung und Regelungstechnik
Universitätsstr. 27, PRG
D-58097 Hagen
Tel.: +49 2331 9871102
Fax: +49 2331 987354
E-mail: ivan.masar@fernuni-hagen.de

## Archív Archive

AT&P journal PLUS1: Adaptívne a nelineárne riadenie systémov (tlačená verzia, vydané 2001)
*Adaptive and nonlinear control systems (printed version, published 2001)*

AT&P journal PLUS2: Robotika, mechatronika, diskrétne výrobné systémy (tlačená verzia, vydané 2001)
*Robotics, mechatronics, discrete manufacturing systems (printed version, published 2001)*

AT&P journal PLUS3: Robustné systémy riadenia (tlačená verzia, vydané 2002)
*Robust control systems (printed version, published 2002)*

AT&P journal PLUS4: Samonastavujúce sa systémy v riadení procesov (tlačená verzia, vydané 2003)
*Selftuning systems in process control (printed version, published 2003)*

AT&P journal PLUS5: Robotické systémy (elektronická – CD verzia, vydané 2004)
*Robotics systems (electronic – CD version, published 2004)*

AT&P journal PLUS6: Mechatronika (elektronická – CD verzia, vydané 2005)
*Mechatronics (electronic – CD version, published 2005)*

AT&P journal PLUS7: Umelá inteligencia v praxi (elektronická – CD verzia, vydané 2005)
*Artifical intelligence in Practise (electronic – CD version, published 2005)*

AT&P journal PLUS 1 2006: Mechatronické systémy (elektronická – CD verzia, vydané 2006)
*Mechatronic systems (electronic – CD version, published 2006)*

AT&P journal PLUS 2 2006: Inteligentné meracie systémy (elektronická – CD verzia, vydané 2006)
*Intelligent measurement systems (electronic – CD version, published 2006)*

AT&P journal PLUS 1 2007: MMaMS'2007 (elektronická – CD verzia, vydané 2007)
*MMaMS'2007 (electronic – CD version, published 2007)*

AT&P journal PLUS 2 2007: Riadenie procesov (elektronická – CD verzia, vydané 2007)
*Process Control (electronic – CD version, published 2007)*

AT&P journal PLUS 1 2008: Mobilné robotické systémy (elektronická – CD verzia, vydané 2008)
*Mobile robotic systems (electronic – CD version, published 2008)*

AT&P journal PLUS 2 2008: Riadenie v energetike (elektronická – CD verzia, vydané 2008)
*Control of Power Systems (electronic – CD version, published 2008)*

AT&P journal PLUS 1 2009: Inteligentné pohybové systémy (elektronická – CD verzia, vydané 2009)
*Intelligent motion control systems (electronic – CD version, published 2009)*