



2/2011

| atp | journal |

plus


Process Control

*reviewed slovak professional
magazine for scientific
and engineering issues*

Riadenie procesov

*recenzované periodikum
vedeckých a inžinierskych
publikácií*

ISSN 1336-5010



Riadenie procesov

Process Control

Odborný garant

Prof. Ing. Miroslav Fikar, DrSc.,
riaditeľ Ústavu informatizácie,
automatizácie a matematiky
na FCHPT STU v Bratislave

Technical guarantee

Prof. Ing. Miroslav Fikar, DrSc.,
Head of Institute of Information Engineering,
Automation and Mathematics,
Faculty of Chemical and Food Technology,
Slovak University of Technology in Bratislava

Vydavateľ

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

Publisher

Spoluzakladateľ

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

Co-founder



www.ieee.cz

Redakčná rada

Draft committee

prof. Ing. Alexík Mikuláš, PhD., FRI ŽU, Žilina
doc. Ing. Dvoran Ján, CSc., FCHPT STU, Bratislava
prof. Ing. Fikar Miroslav, DrSc., FCHPT STU, Bratislava
doc. Ing. Hantuch Igor, PhD., KAR FEI STU, Bratislava
doc. Ing. Hrádický Ladislav, PhD., SJF TU, Košice
prof. Ing. Hulkó Gabriel, DrSc., SJF STU, Bratislava
prof. Ing. Jurišica Ladislav, PhD., FEI STU, Bratislava
doc. Ing. Kachaňák Anton, CSc., SJF STU, Bratislava
prof. Ing. Krokavec Dušan, CSc., KKUI FEI TU Košice
prof. Ing. Madarász Ladislav, PhD., FEI TU, Košice
prof. Ing. Malindžák Dušan, CSc., BERG TU, Košice
prof. Ing. Mészáros Alojz, CSc., FCHPT STU, Bratislava
prof. Ing. Mikleš Ján, DrSc., FCHPT STU, Bratislava
prof. Dr. Ing. Moravčík Oliver, MTF STU, Trnava
prof. Ing. Murgaš Ján, PhD., FEI STU, Bratislava
prof. Ing. Rástočný Karol, PhD., KRIS ŽU, Žilina
prof. Ing. Schreiber Peter, CSc., MTF STU, Trnava
prof. Ing. Skyva Ladislav, DrSc., FRI ŽU, Žilina
prof. Ing. Smieško Viktor, PhD., FEI STU, Bratislava
doc. Ing. Šturcel Ján, PhD., FEI STU, Bratislava
prof. Ing. Taufer Ivan, DrSc., Univerzita Pardubice
prof. Ing. Veselý Vojtech, DrSc., FEI STU, Bratislava
prof. Ing. Žalman Milan, PhD., FEI STU, Bratislava

Ing. Bartošovič Štefan,

generálny riaditeľ ProCS, s.r.o.

Ing. Boďo Vladimír, CSc.,

riaditeľ AXESS, spol. s r.o.

Ing. Csölle Attila,

riaditeľ Emerson Process Management, s.r.o.

Ing. Horváth Tomáš,

riaditeľ HMH, s.r.o.

Ing. Hrica Marián,

riaditeľ divízie A & D, Siemens, s.r.o.

Jiří Kroupa,

riaditeľ kancelárie pre SK, DEHN + SÖHNE

Ing. Murančan Ladislav,

PPA Controll a.s., Bratislava

Ing. Petergáč Štefan,

predseda predstavenstva Datalan, a.s.

Ing. Pilňan Branislav,

sales leader HPS, HONEYWELL s.r.o.

Marcel van der Hoek,

generálny riaditeľ ABB, s.r.o.

Redakcia

Editors office

ATP Journal

Košická 37

821 09 Bratislava 2

tel.: 02/5026 1752 – 5

fax: 02/5026 1757

e-mail: vydavatelstvo@hmh.sk

www.atpjournal.sk

Ing. Anton Gérer

šéfredaktor – editor in chief

gerer@hmh.sk

Ing. Martin Karbovanec

vedúci vydavateľstva – editorial office manager

karbovanec@hmh.sk

Bc. Zuzana Bakošová

marketingová manažérka – marketing manager

mediamarketing@hmh.sk

Ing. Branislav Bložon

odborný redaktor – editor

blozon@hmh.sk

Peter Kanda

DTP grafik – DTP graphic designer

dtp@hmh.sk

Dagmar Votavová

asistentka redakcie – editorial assistant

atp_podklady@hmh.sk

Mgr. Bronislava Chocholová

jazyková redaktorka – text corrector

Obsah

Prenosové matice a ich Jacobsonove formy pre nelineárne riadiace systémy na časových škálach: implementácia CAS	6
Belikov, J., Kotta, U., Leibak, A.	
Architektúra na správu dát pre teleoperované UAV systémy	13
Bahník, P., Pilka, J.	
Digitálny samoladiaci Smithov prediktov založený na umiestnení pólov	21
Bobál, V., Chalupa, P., Dostál, P., Brázdil, M.	
Vylepšené formy bounded real lemmy	29
Filasová, A., Krokavec, D.	
Návrh stavového spätnoväzbového riadenia využitím eigenstructure decoupling	34
Kocsis, P., Fónod, R.	
Diskrétné rozpájanie porúch v spojených zásobníkoch	40
Halás, M.	
Matematické modelovanie a implementácia navigácie vzducholode	45
Jelenčiak, F., Masár, I.	
Vzdialený experiment vo výuke riadenia	50
Kalúz, M., Čirka, L., Fikar, M.	
Prediktívne riadenie zmesy vzduch/palivo v reálnom motore	55
Kopačka, M., Šimončíč, P., Csambál, J., Honek, M., Wojnar, S., Polóni, T., Rohaf-Ilkiv, B.	
PIDTOOL 2.0 - softvér na identifikáciu z prechodových charakteristik a ladenie PID regulátorov	61
Oravec, J., Bakošová, M.	
Implementácia Explicitného MPC v reálnom čase na PLC	67
Rauová, I., Valo, R., Kvasnica, M., Fikar, M.	
Návrh PI regulátora pre laboratórny proces s neurčitostami	73
Závacká, J., Bakošová, M.	

Content

Transfer matrix and its Jacobson form for nonlinear control systems on time scales: CAS implementation	6
Belikov, J., Kotta, U., Leibak, A.	
Data Management Architecture for Tele-operated UAV System	13
Bahník, P., Pilka, J.	
Digital Self-tuning Smith Predictor Based on Pole Assignment Approach	21
Bobál, V., Chalupa, P., Dostál, P., Brázdil, M.	
Bounded Real Lemma Improved Forms	29
Filasová, A., Krokavec, D.	
Eigenstructure Decoupling in State Feedback Control Design	34
Kocsis, P., Fónod, R.	
Discrete-time Disturbance Decoupling of Coupled Tanks	40
Halás, M.	
Mathematical Modeling and Implementation of the Airship Navigation	45
Jelenčiak, F., Masár, I.	
Remote Experiment in Control Education	50
Kalúz, M., Čirka, L., Fikar, M.	
Air/Fuel Ratio Model Predictive Control of a Real-world Gasoline Engine	55
Kopačka, M., Šimončič, P., Csambál, J., Honek, M., Wojnar, S., Polóni, T., Rohaf-Ilkiv, B.	
PIDTOOL 2.0 – Software for Step-Response-Based Identification and PID Controller Tuning	61
Oravec, J., Bakošová, M.	
Real-Time Implementation of Explicit MPC Using PLC	67
Rauová, I., Valo, R., Kvasnica, M., Fikar, M.	
PI Controller Design for Laboratory Process with Uncertainties	73
Závacká, J., Bakošová, M.	

Transfer matrix and its Jacobson form for nonlinear control systems on time scales: CAS implementation

Juri Belikov, Ülle Kotta and Alar Leibak

Abstract

This paper suggests a detailed algorithm for computation of the Jacobson form of the polynomial matrix associated with the transfer matrix describing the multi-input multi-output nonlinear control system, defined on homogeneous time scale. The algorithm relies on the theory of skew polynomial rings. Its implementation is shown on the basis of Computer Algebra System (CAS) *Mathematica*.

Keywords: nonlinear control system, input-output models, time scales, symbolic computations

Introduction

In the Institute of Cybernetics at Tallinn University of Technology symbolic software package NLControl has been developed over the years within *Mathematica* environment, for the detailed information see [17], [24]. The package is based on different algebraic methods, in particular on the approach based on the differential one-forms, see [10], and on the theory of skew polynomial ring. It allows to solve various modelling, analysis and synthesis problems not only for continuous and discrete-time nonlinear control systems, but also for those defined on homogeneous time scales, see [7]. Note that the key idea of a time scale calculus is unification of the theories of differential and difference equations, see [5]. Both continuous and discrete-time (in terms of the difference operator) cases are merged in time scale formalism into a general framework which provides not only unification but also an extension. The main concept of the time scale calculus is the so-called delta-derivative that is a generalization of both time-derivative and the difference operator (but accommodates more possibilities, e.g. q -difference operator).

In the linear control theory the transfer matrix (TM) approach has been very popular. Recently the concept of the TM has been extended for the continuous-time nonlinear control systems [12] and later in [13] or discrete-time systems and for control system defined in terms of the pseudo-linear operator, see [14]. Note that the latter includes also the systems, defined on homogeneous time scales, since in that case the delta-derivative may be understood as the special case of the pseudo-linear operator. However, the pseudo-linear approach cannot handle the systems defined on non-homogeneous time scales, since the time scale formalism unifies both continuous- and discrete-time cases, it would be interesting to study whether TM-based transparent control methods can be extended to nonlinear systems defined on time scale. In TM-based control design, a special form of the matrix, the Jacobson-Teichmüller¹ form, plays a key role. The first step in transformation of the TM into the required form is to transform the polynomial matrix, associated with it, into the Jacobson form.

Note that in the case of nonlinear control systems, the polynomials belong into the non-commutative polynomial ring that is the principal ideal domain (p.i.d.). The basic algorithm to transform a polynomial matrix into this form was given in [9] for an arbitrary ring being the p.i.d. There exist a number of implementations of this algorithm such as [4], [8] and its fraction-free version [20]. However, except [15], not available for public use, all of them have been implemented either in Maple, e.g. [4], [8] or in Singular:Plural [20]. Moreover, it is not documented whether and how these packages are applicable for nonlinear control systems, in particular for those defined on homogeneous time scale.

The main contribution of the paper is the specification the algorithm given in [9] into the form necessary to handle the nonlinear control system defined on homogeneous time scale and description of the experience from its implementation in *Mathematica*, within the package NLControl. All the steps of the algorithm are clear, strictly defined and easily convertible into any programming code. It should be mentioned that some preliminary results for the discrete-time case were presented in [3].

1. Calculus on time scale

For a general introduction to the calculus on time scales, see [5]. Here we give only those notions and facts that we need in our paper.

A *time scale* \mathbb{T} is an arbitrary nonempty closed subset of the set \mathbb{R} of real numbers. The standard cases include $\mathbb{T} = \mathbb{R}$, $\mathbb{T} = \mathbb{Z}$, $\mathbb{T} = h\mathbb{Z}$ for $h > 0$, but also $\mathbb{T} = q^{\mathbb{Z}} := \{q^k: k \in \mathbb{Z}\} \cup \{0\}$, $q > 1$ is a time scale.

The following operators on \mathbb{T} are often used:

- the *forward jump operator* $\sigma: \mathbb{T} \rightarrow \mathbb{T}$, defined by $\sigma(t) := \inf\{\tau \in \mathbb{T}: \tau > t\}$ and $\sigma(\sup \mathbb{T}) = \sup \mathbb{T}$, if $\sup \mathbb{T} \in \mathbb{T}$,
- the *backward jump operator* $\rho: \mathbb{T} \rightarrow \mathbb{T}$, defined by $\rho(t) := \inf\{\tau \in \mathbb{T}: \tau < t\}$ and $\rho(\inf \mathbb{T}) = \inf \mathbb{T}$, if $\inf \mathbb{T} \in \mathbb{T}$,

¹ Note that in the linear control theory this form is called the Smith-McMillan form, see [16].

- the *graininess* function $\mu: \mathbb{T} \rightarrow [0, \infty)$, defined by $\mu(t) = \sigma(t) - t$.

If $\mu \equiv \text{const}$ then a time scale \mathbb{T} is called *homogeneous*. In this paper, we assume that the time scale \mathbb{T} is homogeneous.

Example 1:

- If $\mathbb{T} = \mathbb{R}$, then for any $t \in \mathbb{R}$, $\sigma(t) = t = \rho(t)$, and the graininess function $\mu(t) \equiv 0$.
- If $\mathbb{T} = h\mathbb{Z}$, for $h > 0$, then for every $t \in h\mathbb{Z}$, $\sigma(t) = t + h$, $\rho(t) = t - h$, and $\mu(t) = h$.
- If $\mathbb{T} = \overline{q\mathbb{Z}}$, for $q > 1$, then for every $t \in \mathbb{T}$, $\sigma(t) = qt$, $\rho(t) = t/q$, and $\mu(t) = (q - 1)t$.

So, the first two cases are homogeneous time scales whereas the third is not.

Definition 1: The *delta derivative* of a function $f: \mathbb{T} \rightarrow \mathbb{R}$ at t is the number $f^\Delta(t)$ such that for each $\varepsilon > 0$ there exists a neighbourhood $U(\varepsilon)$ of t , $U(\varepsilon) \subset \mathbb{T}$ such that for all $\tau \in U(\varepsilon)$, $|f(\sigma(t)) - f(\tau) - f^\Delta(t)(\sigma(t) - \tau)| \leq \varepsilon|\sigma(t) - \tau|$.

The typical special cases of the delta operator are summarized in the following remark.

Remark 1:

- If $\mathbb{T} = \mathbb{R}$, then $f: \mathbb{R} \rightarrow \mathbb{R}$ is delta differentiable at $f^\Delta(t) = \lim_{s \rightarrow t} \frac{f(t) - f(s)}{t - s} = f'(t)$, i.e. iff f is differentiable in the ordinary sense at t .
- If $\mathbb{T} = T\mathbb{Z}$, where $T > 0$, then $f: T\mathbb{Z} \rightarrow \mathbb{R}$ is always delta differentiable at every $t \in T\mathbb{Z}$ with $f^\Delta(t) = \frac{f(\sigma(t)) - f(t)}{\mu(t)} = \frac{f(t+T) - f(t)}{T}$ meaning the usual forward difference operator.

Proposition 1: Let $f: \mathbb{T} \rightarrow \mathbb{R}$, $g: \mathbb{T} \rightarrow \mathbb{R}$ be two delta differentiable functions defined on \mathbb{T} and let $t \in \mathbb{T}$. Then the delta derivative satisfies the following properties

- $f^\Delta = f + \mu f^\Delta$,
- $(\alpha f + \beta g)^\Delta = \alpha f^\Delta + \beta g^\Delta$, for any constants α and β ,
- $(fg)^\Delta = f^\sigma g^\Delta + f^\Delta g$,
- if $g g^\Delta \neq 0$, then $(f/g)^\Delta = (f^\Delta g - f g^\Delta)/(g g^\Delta)$.

For a function $f: \mathbb{T} \rightarrow \mathbb{R}$ we define second delta derivative $f^{[2]} := f^{\Delta\Delta}$ provided that f^Δ is delta differentiable on \mathbb{T} . Similarly we define higher order derivatives $f^{[n]}$.

Denote $\sigma^n := \underbrace{\sigma \circ \dots \circ \sigma}_{n\text{-times}}$ and $f^{\sigma^n} := f \circ \sigma^n$.

Proposition 2[18]: Let f and f^Δ be delta differentiable functions on homogeneous time scale \mathbb{T} . Then

- $f^{\Delta\sigma} = f^{\sigma\Delta}$,
- $f^{\sigma^n} = \sum_{k=0}^n C_n^k \mu^k f^{[k]}$.

At the end of this section we introduce some notation that will be useful in the following sections. Let f be a function admitting the delta-derivatives up to the c -th order. Let a and b be integers such that $0 \leq a \leq b \leq c$. We set $f^{[0]} = f$. Let $f^{[a..b]}$ denote the set $\{f^{[a]}, \dots, f^{[b]}\}$.

2. Preliminaries

Consider a multi-input multi-output nonlinear control system described by a set of higher order input-output delta-differential equations on the homogeneous time scale \mathbb{T} relating the inputs u_j , $j = 1, \dots, m$, the outputs y_i , $i = 1, \dots, p$ and the finite number of their delta derivatives:

$$y_i^{[n_i]} = \Phi_i \left(y_1^{[0..n_{i1}-1]}, \dots, y_p^{[0..n_{ip}-1]}, u_1^{[0..s_{i1}]}, \dots, u_m^{[0..s_{im}]}, \right) \quad (1)$$

where the functions Φ_i are real analytic functions of their arguments, and functions $y_i: \mathbb{T} \rightarrow \mathbb{R}$, $i = 1, \dots, p$ and $u_j: \mathbb{T} \rightarrow \mathbb{R}$, $j = 1, \dots, m$ are delta differentiable at least up to order n_i and $s_j := \max_{1 \leq i \leq p} (s_{ij})$, respectively.

2.1 Algebraic framework

Below we briefly recall the algebraic formalism for nonlinear control systems defined on homogeneous time scales, described in [2], [18], [19]. Let \mathcal{K} denote the field of meromorphic functions in a finite number of (independent) variables

$$\mathcal{C} = \left\{ y_1^{[0..n_{i1}-1]}, \dots, y_p^{[0..n_{ip}-1]}, u_j^{[k]}, j = 1, \dots, m, k \geq 0 \right\}.$$

Note that under the mild assumption on submersivity of system (1) (see below) the jump operator $\sigma: \mathcal{K} \rightarrow \mathcal{K}$ and the delta derivative $\Delta: \mathcal{K} \rightarrow \mathcal{K}$ may be extended to the field \mathcal{K} as follows, see [19]

$$\begin{aligned} \sigma(F) &= \left(y_1^{[0..n_1-1]}, \dots, y_p^{[0..n_p-1]}, u_1^{[0..s_1+1]}, \dots, u_m^{[0..s_m+1]} \right) \\ &:= F \left(y_1^{[0..n_1-1]\sigma}, \dots, y_p^{[0..n_p-1]\sigma}, u_1^{[0..s_1]\sigma}, \dots, u_m^{[0..s_m]\sigma} \right), \end{aligned}$$

where

$$\begin{aligned} &y_i^{[0..n_i-1]\sigma} \\ &= y_i^{[0..n_i-1]} + \mu \cdot \left[y_i^{[1..n_i-1]}, \right. \\ &\left. \Phi \left(y_1^{[0..n_{i1}-1]}, \dots, y_i^{[0..n_i-1]}, \dots, y_p^{[0..n_{ip}-1]}, u_1^{[0..s_{p1}]}, \dots, u_m^{[0..s_{pm}]} \right) \right], \end{aligned}$$

$$i = 1, \dots, p, u_j^{[0..s_j]\sigma} = u_j^{[0..s_j]} + \mu u_j^{[1..s_j+1]}, j = 1, \dots, m \text{ and}^2$$

$$\begin{aligned} \Delta(F) &= \left(y_i, \dots, y_p^{[n_i-1]}, u_i, \dots, u_j^{[k+1]} \right) \\ &:= \int_0^1 \left\{ \text{grad} F \left(y_i + h\mu y_i^\Delta, \dots, y_i^{[n_i-1]} \right. \right. \\ &\left. \left. + h\mu \Phi_i \left(y_1^{[0..n_{i1}-1]}, \dots, y_p^{[0..n_{ip}-1]}, u_1^{[0..s_{i1}]}, \dots, u_m^{[0..s_{im}]} \right), u_j \right. \right. \\ &\left. \left. + h\mu u_j^\Delta, u_j^{[k]} + h\mu u_j^{[k+1]} \right) \right. \\ &\left. \cdot \left[\begin{array}{c} \left(y_1^{[1..n_1-1]}, \dots, y_p^{[1..n_p-1]} \right)^T, \\ \Phi_i \left(y_1^{[0..n_{i1}-1]}, \dots, y_p^{[0..n_{ip}-1]}, u_1^{[0..s_{i1}]}, \dots, u_m^{[0..s_{im}]} \right) \\ \left(u_1^{[1..s_1+1]}, \dots, u_m^{[1..s_m+1]} \right)^T \end{array} \right] \right\} dh. \end{aligned}$$

Notice that we will use $\sigma(F)$ and F^σ to denote the action of σ on F . Similarly, both $\Delta(F)$ and F^Δ will be used interchangeably.

² Proposition 3.3 from [2] shows how $\Delta(F)$ may be calculated not using integral explicitly.

In case σ is not injective, there may exist non-zero functions φ such that $\sigma(\varphi) = 0$ meaning that the operator σ is not well-defined on the field \mathcal{K} . For σ to be an injective endomorphism on \mathcal{K} , the system (1) has to be submersive which can be guaranteed by the condition of the following theorem.

Theorem 1[19]: The nonlinear control system, defined on homogeneous time scale via the higher order i/o equations (1), is submersive if and only if the following condition

$$\text{rank}_{\mathcal{K}} \begin{pmatrix} 1 + \alpha_{11} & \cdots & \alpha_{1p} & \beta_{11} & \cdots & \beta_{1m} \\ \alpha_{p1} & \cdots & 1 + \alpha_{pp} & \beta_{p1} & \cdots & \beta_{pm} \end{pmatrix} = p \quad (2)$$

holds, where

$$\alpha_{ij} := \sum_{k=0}^{n_j-1} (-1)^{n_j-k-1} \mu^{n_j-k} \frac{\partial \Phi_i}{\partial y_j^{[k]}}$$

$i, j = 1, \dots, p$ and

$$\beta_{lk} := \sum_{j=0}^s (-1)^{s-j-1} \mu^{s-j+2} \frac{\partial \Phi_l}{\partial u_k^{[j]}}$$

$l = 1, \dots, p, k = 1, \dots, m$.

The operator Δ satisfies a generalization of Leibnitz rule

$$(FG)^\Delta = F^\sigma G^\Delta + F^\Delta G, \quad (3)$$

for $F, G \in \mathcal{K}$. The derivation satisfying rule (3) is called a " σ -derivation", see [9]. Therefore, \mathcal{K} is a differential field equipped with a σ -derivation Δ . In general, the field \mathcal{K} is not inversive, i.e. not every element of \mathcal{K} has a pre-image. Nevertheless, since Δ is injective, up to an isomorphism there exists an inversive σ -differential overfield \mathcal{K}^* , called the *inversive closure* of \mathcal{K} , such that Δ can be extended to \mathcal{K}^* and this extension is automorphism of \mathcal{K}^* , see [9]. In [2] the details of construction of \mathcal{K}^* for nonlinear control systems defined on time scales can be found. Below assume that \mathcal{K}^* is given and use the same symbol \mathcal{K} to denote the σ -differential field and its inversive closure.

Over the σ -differential field \mathcal{K} one can define the vector space

$$\mathcal{E} := \text{span}_{\mathcal{K}} \left\{ dy_i, dy_i^\Delta, \dots, dy_i^{[n_i-1]}, i = 1, \dots, p, du_i^{[k]}, j = 1, \dots, m, k \geq 0 \right\}.$$

The elements of \mathcal{E} are called one-forms. For $F \in \mathcal{K}$ we define the operator $d: \mathcal{K} \rightarrow \mathcal{E}$ as follows

$$dF := \sum_{i=1}^p \sum_{l=0}^{n_i-1} \frac{\partial F}{\partial y_i^{[l]}} dy_i^{[l]} + \sum_{j=1}^m \sum_{\ell=0}^k \frac{\partial F}{\partial u_j^{[\ell]}} du_j^{[\ell]}.$$

dF is said to be the (*total*) *differential* of the function F and is a one-form.

Let $\omega = \sum_j \alpha_j d\varphi_j$ be a one-form with $\alpha_j \in \mathcal{K}$ and $\varphi_j \in \mathcal{C}$. Then, the operators $\sigma: \mathcal{K} \rightarrow \mathcal{K}$ and $\Delta: \mathcal{K} \rightarrow \mathcal{K}$ induce the operators $\sigma: \mathcal{E} \rightarrow \mathcal{E}$ and $\Delta: \mathcal{E} \rightarrow \mathcal{E}$ by

$$\sigma(\omega) := \sum_i \sigma(\alpha_i) d[\sigma(\varphi_i)], \quad (4)$$

$$\Delta(\omega) := \sum_i \{ \Delta(\alpha_i) d\varphi_i + \sigma(\alpha_i) d[\Delta(\varphi_i)] \}. \quad (5)$$

Since $\sigma(\alpha_i) = \alpha_i + \mu \Delta(\alpha_i)$, (5) may be alternatively written as

$$\Delta(\omega) := \sum_i \{ \Delta(\alpha_i) d\varphi_i + (\alpha_i + \mu \Delta(\alpha_i)) d[\Delta(\varphi_i)] \}.$$

It has been proved that $\Delta(dF) = d[F^\Delta]$, $\sigma(dF) = d[F^\sigma]$ and $\Delta\sigma = \sigma\Delta$, see [2].

2.2 Polynomial framework

Next, we recall the polynomial formalism which allows representing the nonlinear i/o equations (1) via two polynomial matrices. Consider the differential field \mathcal{K} with the σ -derivation Δ with σ being an automorphism of \mathcal{K} . A *left* differential polynomial is an element which can be uniquely written in the form $a(\partial) = \sum_{i=0}^n a_i \partial^{n-i}$, $a_i \in \mathcal{K}$, where ∂ is a formal variable and $a(\partial) \neq 0$ if and only if at least one of the coefficients a_i , $i = 0, \dots, n$ is nonzero. If $a_0 \neq 0$, then the positive integer n is called the degree of the left polynomial $a(\partial)$, denoted by $\text{deg} a(\partial)$. Besides that we set $\text{deg} 0 = -\infty$. The addition of the left differential polynomials is defined in the standard way. However, for $a \in \mathcal{K}$ the multiplication is defined by

$$\partial \cdot a := a^\sigma \partial + a^\Delta. \quad (6)$$

The ring of differential polynomials will be denoted by $\mathcal{K}[\partial; \sigma, \Delta]$. Since σ is an automorphism, the ring of the left differential polynomials is a skew polynomial ring, that is proved to satisfy the left Ore condition, see [11]. By left Ore condition for all nonzero $a, b \in \mathcal{K}[\partial; \sigma, \Delta]$ there exist nonzero $a_1, b_1 \in \mathcal{K}[\partial; \sigma, \Delta]$ such that $a_1 b = b_1 a$, that is, a and b have a common left multiple (clm). The ring $\mathcal{K}[\partial; \sigma, \Delta]$ can, therefore, be embedded into its quotient field (field of fractions) by defining its left quotients as $\frac{a}{b} = b^{-1} \cdot a$, see [23]. Denote the resulting quotient field by $\mathcal{K}(\partial; \sigma, \Delta)$. Moreover, we write $\mathcal{K}(\partial; \sigma, \Delta)^{p \times m}$ for the set of $p \times m$ rational matrices with entries in $\mathcal{K}(\partial; \sigma, \Delta)$, and $\mathcal{K}[\partial; \sigma, \Delta]^{p \times m}$ for the set of $p \times m$ polynomial matrices with entries in $\mathcal{K}[\partial; \sigma, \Delta]$.

Denote $\sigma^n(a)$ by a^{σ^n} for $a \in \mathcal{K}$.

Lemma 1: Let $a \in \mathcal{K}$. Then $\partial^n \cdot a \in \mathcal{K}[\partial; \sigma, \Delta]$, for $n \geq 0$, and $\partial^n \cdot a = \sum_{i=0}^n C_n^i (a^{[n-i]})^{\sigma^i} \partial^i$.

In order to describe the i/o equation (1) via two polynomial matrices, we define

$$\partial^k dy_v := dy_v^{(k)}, \quad \partial^l du_v := du_v^{(l)} \quad (7)$$

for $k, v = 1, \dots, p, v = 1, \dots, m$ and $l \geq 0$ in the vector space \mathcal{E} . Since an arbitrary one-form $\omega \in \mathcal{E}$ has the form $\omega = \sum_{v=1}^p \sum_{i=0}^{n-1} a_{vi} dy_v^{(i)} + \sum_{v=1}^m \sum_{j=0}^k b_{vj} du_v^{(j)}$, where $a_{vi}, b_{vj} \in \mathcal{K}$, so ω can be expressed in terms of the left differential polynomials as $\omega = \sum_{v=1}^p (\sum_{i=0}^{n-1} a_{vi} \partial^i) dy_v + \sum_{v=1}^m (\sum_{j=0}^k b_{vj} \partial^j) du_v$. A left differential polynomial can be considered as an operator acting on vectors $y = [y_1, \dots, y_p]^T$ and $u = [u_1, \dots, u_m]^T$ from \mathcal{E} : $(\sum_{i=0}^k a_i \partial^i)(\alpha d\zeta) := \sum_{i=0}^k a_i (\partial^i \cdot \alpha) d\zeta$, with $a_i, \alpha \in \mathcal{K}$ and $d\zeta \in \{dy, du\}$, where by Lemma 1, $\partial^i \cdot \alpha = \sum_{k=0}^i C_i^k (a^{[n-i]})^{\sigma^k} \partial^k$. It is easy to note that $\partial(\omega) = \Delta(\omega)$, for $\omega \in \mathcal{E}$.

Now, by differentiating equation (1) and using (7) we get

$$P(\partial)dy = Q(\partial)du, \quad (8)$$

where $P(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]^{p \times p}$ and $Q(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]^{p \times m}$.

We assume that the Dieudonné determinant of the matrix $P(\partial)$ in (8) is nonzero, see [1] for details. The latter means that the following definition of the transfer matrix is well-defined.

Definition 2: An element of the form $H(\partial) := P^{-1}(\partial)Q(\partial) \in \mathcal{K}(\partial; \sigma, \Delta)^{p \times m}$, such that $dy = H(\partial)du$, is said to be a transfer matrix of nonlinear system³ (1).

³ Notice that there exists an algorithm which allows to obtain the transfer matrix from a nonlinear system described by state-space equations, for additional information see [12].

Note that though every control system can be described by the transfer matrix, the converse is not always true. The reason is that the one-form corresponding to the transfer function may not be integrable, see [13] for details.

2.3 Polynomial matrices

Here we recall some basic properties of matrices with skew-polynomial entries. Suppose the matrix $P(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]^{p \times m}$.

Definition 3: An elementary column (row) operation on a polynomial matrix $P(\partial)$ is one of the following four operations

- i. interchanging two columns (rows);
- ii. multiplying any column (row) by invertible element $k \in \mathcal{K}[\partial; \sigma, \Delta]$ from the right (left);
- iii. adding a right (left) multiple of one column (row) to another;
- iv. replacement of the first elements of any two columns (rows) by their greatest common left (right) divisor ($\text{gcl}(r)$) and zero, respectively.

All these operations correspond to multiplication of the matrix $P(\partial)$ by an elementary matrix $E_R^s(\partial)$ or $E_L^s(\partial)$ from the right or left, respectively [Cohn, 1974], where $s \in \{(i) - (iv)\}$. Operations (i)-(iii) may be represented by the product of the matrices of the form $E_{ij}(\partial) = I_v + 1_{ij}k$ with I_v identity matrix and 1_{ij} the matrix made of a single 1 at the intersection of row i and column j , $1 \leq i, j \leq v$, and zeros elsewhere, with $k \in \mathcal{K}[\partial; \sigma, \Delta]$ and with $v = m$ for actions with columns and $v = p$ for actions with rows, see [21]. The elementary matrices corresponding to the operations from Definition 3 can be obtained

- i. by swapping columns (rows) i and j of the identity matrix;
- ii. by multiplying all elements of the corresponding column (row) of identity matrix by $k \in \mathcal{K}[\partial; \sigma, \Delta]$;
- iii. from identity matrix with element $k \in \mathcal{K}[\partial; \sigma, \Delta]$ in entry (i, j) .
- iv. The procedure for constructing this matrix is described in the algorithm presented in Section 4.

Definition 4: A matrix $U(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]^{q \times q}$ is called unimodular if it has an inverse $U^{-1}(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]^{q \times q}$.

Every right or left unimodular matrix $U_R(\partial)$ or $U_L(\partial)$ may be obtained as a product of the corresponding elementary matrices from Definition 3.

In order to find the gclid , one may use the *left Euclidean division algorithm*, see [6]. For given two polynomials $p_1(\partial)$ and $p_2(\partial)$ with $\deg(p_1(\partial)) > \deg(p_2(\partial))$ there exist unique polynomials $\gamma_1(\partial)$ and $p_3(\partial)$ such that

$$p_1(\partial) = p_2(\partial)\gamma_1(\partial) + p_3(\partial), \quad \deg(p_3(\partial)) < \deg(p_2(\partial)).$$

Using the left Euclidean division algorithm, after $k - 1$ steps, one obtains $p_i(\partial) = p_{i+1}(\partial)\gamma_i(\partial) + p_{i+2}(\partial)$ for $i = 2, \dots, k - 2$ and $p_{k-1}(\partial) = p_k(\partial)\gamma_{k-1}(\partial)$. Hence the gclid of $p_1(\partial)$ and $p_2(\partial)$ is $p_k(\partial)$. Moreover, eliminating polynomials $p_{k-1}(\partial), \dots, p_3(\partial)$ we get the *Bézout identity*, i.e. there exist polynomials $u(\partial), v(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]$ such that $p_1(\partial)u(\partial) + p_2(\partial)v(\partial) = p_k(\partial)$. Note that the *right Euclidean division algorithm* can be defined in a similar manner.

2. Jacobson Form

For $P(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]^{p \times m}$ one can find elementary row and column operations corresponding to multiplication by unimodular matrices $U_L^{p \times p}(\partial)$ and $U_R^{m \times m}(\partial)$, respectively, such that

$$U_L(\partial)P(\partial)U_R(\partial) = \Lambda(\partial), \quad (9)$$

where $\Lambda(\partial) = \text{diag}\{\lambda_1(\partial), \dots, \lambda_r(\partial), 0, \dots, 0\}$, and $\lambda_i(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]$ are unique monic polynomials obeying a property that $\lambda_{i+1}(\partial)$ is divisible by $\lambda_i(\partial)$, $\lambda_i(\partial) \parallel \lambda_{i+1}(\partial)$, i.e. there exist $\alpha_i(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]$ such that $\lambda_{i+1}(\partial) = \lambda_i(\partial) \cdot \alpha_i(\partial)$ for all $i = 1, \dots, r - 1$. The matrix $\Lambda(\partial)$ is called the *Jacobson form* of $P(\partial)$, and $\lambda_i(\partial)$ are called the invariant polynomials of $P(\partial)$, see [22].

Suppose $H(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]^{p \times m}$ is a transfer matrix whose entries are assumed to be in the irreducible form, i.e. without common left factors in the corresponding numerators and denominators, and write it in a standard form

$$H(\partial) = [q(\partial)]^{-1}P(\partial), \quad (10)$$

where the matrix $P(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]^{p \times m}$ is a polynomial matrix and $q(\partial)$ is the monic least common left multiple (lclm) of the denominators of all entries of $H(\partial)$. Then, $P(\partial) = q(\partial)H(\partial)$ is a polynomial matrix that can be transformed into the Jacobson form as above.

2.1 The main Algorithm

The algorithm, presented below, allows transforming the matrix $P(\partial)$ into the Jacobson form. Consider the matrix

$$P(\partial) = \begin{pmatrix} p_{11}(\partial) & \cdots & p_{1m}(\partial) \\ \vdots & \ddots & \vdots \\ p_{p1}(\partial) & \cdots & p_{pm}(\partial) \end{pmatrix}$$

in the ring $\mathcal{K}[\partial; \sigma, \Delta]^{p \times m}$.

Step 1. $k := 1$.

Step 2. Find $p_{ij}(\partial) \neq 0$ for $i = k, \dots, p$ and $j = k, \dots, m$ with the lowest degree and, using operation (i) from Definition 3, put it on the position (k, k) .

Step 3. Using elementary column (item (a)) and row (item (b)) operation (iv) from Definition 3,

- a. replace the elements $p_{kk}(\partial)$ and $p_{kj}(\partial)$ for $j = k + 1, \dots, m$ with their gclid and zero, respectively. This operation can be implemented by solving the following equations

$$p_{kk}(\partial)a_{kk}(\partial) + p_{kj}(\partial)c_{jk}(\partial) = e_{kk}(\partial), \quad (11)$$

$$p_{kk}(\partial)b_{kj}(\partial) + p_{kj}(\partial)d_{jj}(\partial) = 0 \quad (12)$$

with respect to $a_{kk}(\partial), b_{kj}(\partial), c_{jk}(\partial)$ and $d_{jj}(\partial)$, and multiplying $P(\partial)$ from the right by the elementary matrix $E_{Rkj}^4(\partial)$, which can be constructed as follows. Create $m \times m$ identity matrix and put the elements $a_{kk}(\partial), b_{kj}(\partial), c_{jk}(\partial)$ and $d_{jj}(\partial)$ on the positions $(k, k), (k, j), (j, k)$ and (j, j) , respectively. Making $(m - k)$ replacements specified above, we transform the matrix $P(\partial)$ into the new matrix with $p_{kk}(\partial) = e_{kk}(\partial)$, $p_{k, k+1}(\partial) = \dots = p_{km}(\partial) = 0$ and some new elements $p_{il}(\partial)$ for $i = k + 1, \dots, p$ and $l = k, \dots, m$ obtained after multiplication $P(\partial)$ by the respective matrix $U_{Rk}(\partial) = E_{Rk, k+1}^4(\partial) \cdot \dots \cdot E_{Rkm}^4(\partial)$.

- b. replace the elements $p_{kk}(\partial)$ and $p_{ik}(\partial)$ for $i = k + 1, \dots, p$ with their gclid and zero, respec-

tively. The previous operation can be implemented by solving the following equations

$$a_{kk}(\partial)p_{kk}(\partial) + b_{ki}(\partial)p_{ik}(\partial) = e_{kk}(\partial), \quad (13)$$

$$c_{ik}(\partial)p_{kk}(\partial) + d_{ii}(\partial)p_{ik}(\partial) = 0 \quad (14)$$

with respect to $a_{kk}(\partial), b_{ki}(\partial), c_{ik}(\partial)$ and $d_{ii}(\partial)$, and multiplying $P(\partial)$ from the left by the elementary matrix $E_{L_{ik}}^4(\partial)$, which can be constructed as follows. Create $p \times p$ identity matrix and put the elements $a_{kk}(\partial), b_{ki}(\partial), c_{ik}(\partial)$ and $d_{ii}(\partial)$ on the positions $(k, k), (k, i), (i, k)$ and (i, i) , respectively. Making $(p - k)$ replacements specified above, we transform the matrix $P(\partial)$ into the new matrix with $p_{kk}(\partial) = e_{kk}(\partial), p_{k+1,k}(\partial) = \dots = p_{pk}(\partial) = 0$ and some new elements $p_{lj}(\partial)$ for $j = k + 1, \dots, m$ and $l = k, \dots, p$ obtained after multiplication $P(\partial)$ by the respective matrix $U_{Lk}(\partial) = E_{L_{k+1,k}}^4(\partial) \cdot \dots \cdot E_{L_{pk}}^4(\partial)$.

However, in the course of doing this, nonzero entries may reappear in the k -th row of the matrix $P(\partial)$, and one has then to repeat Step 3. Note that at each iteration the number of divisors of the element $p_{kk}(\partial)$ reduces, and therefore, in a finite number of steps the k -th row and column become zero. The latter means that after a finite number of consecutive steps one will obtain the matrix with $p_{kk}(\partial) = e_{kk}(\partial)$ and other entries in the k -th row and column equal to zero.

Step 4. If $k \neq \min(p, m) - 1$, then $k := k + 1$ and go to **Step 2**, otherwise go to **Step 5**.

Step 5. If $p = m$, then go to **Step 6**, otherwise depending whether $m > p$ or $m < p$, one has to execute additional $(m - k)$ or $(p - k)$ operations over the last column(s) or row(s) described in **Steps 3(a)** or **3(b)**, respectively.

Step 6. Consider the elements of the main diagonal $p_{ii}(\partial), \dots, p_{kk}(\partial)$. Here, the following two cases are possible:

- If the divisibility property holds for all pairs $p_{ii}(\partial) \parallel p_{jj}(\partial)$ for $1 \leq i < j \leq k$, then go to **Step 7**.
- If the divisibility property does not hold for some pair of elements $p_{ii}(\partial)$ and $p_{jj}(\partial)$ with $1 \leq i < j \leq k$, i.e. $p_{ii}(\partial) \nparallel p_{jj}(\partial)$, then, using row operation (iii) from Definition 3, the matrix $P(\partial)$ has to be transformed into a new matrix with element $p_{jj}(\partial)$ on the position (i, j) by adding the j -th row to the i -th row. After that, execute again all **Steps 2-5** with modified matrix $P(\partial)$ and $k = i$. The main idea of this transformation and the subsequent executing of the steps 2-5 consists in replacing the element $p_{ii}(\partial)$ by $\text{gcd}(p_{ii}(\partial), p_{jj}(\partial))$ or $\text{gcd}(p_{ii}(\partial), p_{jj}(\partial))$, respectively, obeying a divisibility property $p_{ii}(\partial) \parallel p_{jj}(\partial)$.

Step 7. End of the Algorithm.

Remark 2: Equations (11) and (13) are Bézout identities and can be solved using the left and right Euclidean division algorithm, respectively. Besides, equations (12) and (14) are the right and left Ore conditions, respectively. For example, for (14) it means that there exist $c_{ik}(\partial), d_{ii}(\partial) \in \mathcal{K}[\partial; \sigma, \Delta]$ such that $c_{ik}(\partial)p_{kk}(\partial) = -d_{ii}(\partial)p_{ik}(\partial)$ holds.

We have implemented the algorithm for computing Jacobson form in *Mathematica* package NLControl. However, it should be mentioned that even for the very simple examples calculations become extremely complex. Note that in our calculations, we have to simplify the obtained expressions using the relations, defined by the system equations (1) as well as those, obtained from (1) by taking the delta deriva-

tives. If not done, the computations may lead to erroneous result.

Example 1: Consider the system described by the input-output equations

$$\begin{aligned} y_1^{[2]} &= u_1(1 + y_1^\Delta) + u_1^\Delta(y_1 + \mu y_1^\Delta) - u_2 \\ y_2^\Delta &= u_1 y_2 - u_2 \end{aligned} \quad (15)$$

First, we compute, according to Definition 2 and using the property (i) from Proposition 1, the transfer matrix of the system (15)

$$H(\partial) = \begin{pmatrix} \frac{y_1^\sigma \partial + y_1^\Delta + 1}{\partial^2 - u_1^\sigma \partial - u_1^\Delta} & \frac{1}{-\partial^2 + u_1^\sigma \partial + u_1^\Delta} \\ \frac{y_2}{\partial - u_1} & \frac{1}{-\partial + u_1} \end{pmatrix}.$$

Since the lclm of all the denominators in $H(\partial)$ equals to $\partial^2 - u_1^\sigma \partial - u_1^\Delta$, multiplying numerators of the elements $h_{21}(\partial), h_{22}(\partial)$ by ∂ from the left, decomposition (10) for this example takes the form

$$H(\partial) = (\partial^2 - u_1^\sigma \partial - u_1^\Delta)^{-1} \cdot \begin{pmatrix} y_1^\sigma \partial + y_1^\Delta + 1 & -1 \\ y_2^\sigma \partial + y_2^\Delta & -\partial \end{pmatrix}.$$

Obviously, the element $p_{12}(\partial) = -1$ is that of the lowest possible degree of $P(\partial)$ and, after permuting the rows and columns, i.e. multiplying $P(\partial)$ by the corresponding elementary matrix⁴

$$E_{R12}^1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

from the right, we obtain

$$P(\partial) = \begin{pmatrix} -1 & y_1^\sigma \partial + y_1^\Delta + 1 \\ -\partial & y_2^\sigma \partial + y_2^\Delta \end{pmatrix}. \quad (16)$$

Next, one can easily check that $e_{11} := \text{gcd}(p_{11}, p_{12}) = 1$. After solving equations (11) and (12), corresponding to this example, i.e. the equations

$$\begin{aligned} (-1) \cdot a_{11}(\partial) + (y_1^\sigma \partial + y_1^\Delta + 1) \cdot c_{21}(\partial) &= 1, \\ (-1) \cdot b_{12}(\partial) + (y_1^\sigma \partial + y_1^\Delta + 1) \cdot d_{22}(\partial) &= 0, \end{aligned}$$

we obtain $a_{11}(\partial) = -1, b_{12}(\partial) = y_1^\sigma \partial + y_1^\Delta + 1, c_{21}(\partial) = 0, d_{22}(\partial) = 1$. According to Step 3(a), we construct the matrix

$$E_{R12}^4(\partial) = \begin{pmatrix} -1 & y_1^\sigma \partial + y_1^\Delta + 1 \\ 0 & 1 \end{pmatrix}$$

and multiply (16) by it from the right to get

$$\begin{pmatrix} 1 & 0 \\ \partial - y_1^{\sigma^2} & \partial^2 - (2y_1^{\Delta\sigma} - y_2^\sigma + 1)\partial - y_1^{[2]} + y_2^\Delta \end{pmatrix}. \quad (17)$$

Again, one can check that $e_{11} := \text{gcd}(p_{11}, p_{21}) = 1$. Therefore, solving equations (13) and (14), i.e. the equations

$$\begin{aligned} a_{11}(\partial) \cdot 1 + b_{12}(\partial) \cdot \partial &= 1, \\ c_{21}(\partial) \cdot 1 + d_{22}(\partial) \cdot \partial &= 0, \end{aligned}$$

we obtain $a_{11}(\partial) = 1, b_{12}(\partial) = 0, c_{21}(\partial) = -\partial, d_{22}(\partial) = 1$. According to Step 3(b), we construct the matrix

$$E_{L21}^4(\partial) = \begin{pmatrix} 1 & 0 \\ -\partial & 1 \end{pmatrix}$$

and multiply (17) by it from the left to obtain

$$\Lambda(\partial) = \begin{pmatrix} 1 & 0 \\ 0 & \partial^2 - (2y_1^{\Delta\sigma} - y_2^\sigma + 1)\partial - y_1^{[2]} + y_2^\Delta \end{pmatrix}.$$

⁴ In order not to mislead the reader, note that not all the operations listed in Definition 3 have been used in this example, but only those that correspond to the cases $s = 1$ and $s = 4$.

Due to the fact that the number of rows of $P(\partial)$ equals to the number of its columns, one can directly go to Step 6. Obviously, the division property $\lambda_1(\partial) \parallel \lambda_2(\partial)$ holds. Finally, decomposition (9) of $P(\partial)$ is

$$\Lambda(\partial) = E_{L21}^4(\partial) P(\partial) E_{R12}^1(\partial) E_{R12}^4(\partial).$$

Conclusions

In this paper we have suggested a detailed algorithm for computation of the Jacobson form of the polynomial matrix associated with the transfer matrix describing the multi-input multi-output nonlinear control system, defined on homogeneous time scale, using the theory of skew polynomials. In addition, we adapted the algorithm given in [9] for the case of the nonlinear control systems defined on homogeneous time scale. Notice that, using previous experience with *Mathematica* program, we implemented our results in NLControl package. However, the algorithm is presented in a form that can be easily implemented by any programming language.

Acknowledgements

This work was partially supported by the Governmental funding project no. SF0140018s08 and Estonian Science Foundation Grants no. 8365 and 8787.

References

[1] ARTIN, E.: *Geometric algebra*. Interscience publishers, New York, London, 1957.

[2] BARTOSIEWICZ, Z., KOTTA, Ü., PAWŁUSZEWICZ, E., WYRWAS, M.: Algebraic formalism of differential one-forms for nonlinear control systems on time scales. *Estonian Academy of Science*, 56(3): 264–282, 2007.

[3] BELIKOV, J., KOTTA, Ü., LEIBAK, A.: Transformation of the transfer matrix of the nonlinear system into the Jacobson form. *International Congress on Computer Applications and Computational Science*, pp. 495–498, Singapore, 2010.

[4] BLINKOV, Y.A., CID, C.F., GERDT, V.P., PLESKEN, W., ROBERTZ, D.: The maple package „Janet”: II. linear partial differential equations. *The 6th International Workshop on Computer Algebra in Scientific Computing*, pp. 41–54, Passau, Germany, 2003.

[5] BOHNER, M., PETERSON, A.: *Dynamic equations on time scales*. Birkhäuser, Boston, USA, 2001.

[6] BRONSTEIN, M., PETKOVŠEK, M.: An introduction to pseudolinear algebra. *Theoretical Computer Science*, 157(1):3–33, 1996.

[7] CASAGRANDE, D., KOTTA, Ü., TÕNSO, M., WYRWAS, M.: Mathematica application for nonlinear control systems on time scales. *International Congress on Computer Applications and Computational Science*, pp. 621–624, Singapore, 2010.

[8] CHYZAK, F., QUADRAT, A., ROBERTZ, D.: Oremodules: A symbolic package for the study of multidimensional linear systems. *Applications of Time Delay Systems*, Lecture Notes in Control and Information Sciences, pages 233–264. Springer Berlin / Heidelberg, 2007.

[9] COHN, P.M.: *Free rings and their relations*. Academic Press, London, UK, 1985.

[10] CONTE, G., MOOG, C., PERDON, A.M.: *Algebraic Methods for Nonlinear Control Systems. Theory and Appli-*

cations. Communications and Control Engineering. Springer-Verlag, London, 2nd edition, 2007.

[11] FARB, B., DENNIS, R.K.: *Noncommutative algebra*. Springer-Verlag, New York, USA, 1993.

[12] HALÁS, M.: An algebraic framework generalizing the concept of transfer functions to nonlinear systems. *Automatica*, 44(5):1181–1190, 2008.

[13] HALÁS, M., KOTTA, Ü.: Transfer functions of discrete-time nonlinear control systems. *Estonian Academy Science*, 56(4):322–335, 2007.

[14] HALÁS, M., KOTTA, Ü.: Pseudo-linear algebra: a powerful tool in unification of the study of nonlinear control systems. *The 7th IFAC Symposium on Nonlinear Control Systems*, pp. 684–689, Pretoria, South Africa, 2007.

[15] INSUA, M.A., LADRA, M.: Smith normal form can be computed using Gröbner bases. *International Congress of Mathematicians*, Madrid, Spain, August 2006.

[16] ITO, N., SCHMALE, W., WIMMER, H.K.: Computation of minimal state space realizations in Jacobson normal form. *Contemporary Mathematics*, pp. 221–232. *American Mathematical Society*, Boston, MA, USA, 2003.

[17] KOTTA, Ü., TÕNSO, M.: Linear algebraic tools for discrete-time nonlinear control systems with Mathematica. *Nonlinear and Adaptive Control*, Lecture Notes in Control and Information Sciences, pp. 195–205. Springer Berlin / Heidelberg, 2003.

[18] KOTTA, Ü., BARTOSIEWICZ, Z., PAWŁUSZEWICZ, E., WYRWAS, M.: Irreducibility, reduction and transfer equivalence of nonlinear input-output equations on homogeneous time scales. *Systems and Control Letters*, 58(9):646–651, 2009.

[19] KOTTA, Ü., REHAK, B., WYRWAS, M.: On submersivity assumption for nonlinear control systems on homogeneous time scales. *Estonian Academy of Sciences*, 60(1):25–37, 2011.

[20] LEVANDOVSKYY, V. SCHINDELAR, K.: Computing diagonal form and Jacobson normal form of a matrix using Gröbner bases. <http://arxiv.org/abs/1003.3785>, March 2010.

[21] LÉVINE, J.: *Analysis and Control of Nonlinear Systems*. Springer-Verlag, Berlin, Germany, 2009.

[22] NAKAYAMA, T.: A note on the elementary divisor theory in noncommutative domains. *Bull. Amer. Math. Soc.*, 44(10):719–723, 1938.

[23] ORE, O.: Theory of non-commutative polynomials. *Annals of Mathematics*, 34:480–508, 1933.

[24] TÕNSO, M., RENNİK, H., KOTTA, Ü.: WebMathematica-based tools for discrete-time nonlinear control systems. *Estonian Academy of Science*, 58(4):224–240, 2009.

Juri Belikov, MSc.

Tallinn University of Technology
Control Systems Department
Institute of Cybernetics
Akadeemia tee 21
12618 Tallinn
Estonia
E-mail: jbelikov@cc.ioc.ee

Ülle Kotta, DSc.

Tallinn University of Technology
Control Systems Department
Institute of Cybernetics
Akadeemia tee 21
12618 Tallinn
Estonia
E-mail: kotta@cc.ioc.ee

Alar Leibak, PhD.

Tallinn University of Technology
Department of Mathematics
Ehitajate tee 5
19086 Tallinn
Estonia
E-mail: alar@staff.ttu.ee

Data Management Architecture for Tele-operated UAV System

Pavol Bahník, Jakub Pilka

Abstrakt

Nowadays, more frequently than ever, the unmanned aerial vehicles (UAVs) are used effectively as mobile sensor platforms. The UAV system equipped with an airborne camera and special sensors is a valuable source of various, important information helping to build an actual overview of an environment. It can take place like an observer in disaster situations as well as a special mobile monitoring device which is able to collect required data from a predefined area. This paper introduces our approach to design effective data management architecture to be able to manage, reliably distribute and represent different types of measured data with taking many aspects and limitations of the tele-operated UAV system to consideration.

Klíčové slová: UAV, communication, data flow, data representation, 3D scan, laser scanner

INTRODUCTION

In the last few years various UAV systems became very popular as an effective platform to observe particular areas and collect the data using specific sensors. The UAVs enable us to obtain a bird's eye view of the environment, having access to areas, where often only incomplete and inconsistent information is available. To get actual and precise information or data from a desired place is important in many situations of a modern crisis management or plenty of inspection or data acquisition tasks. Proper utilization of collected data usually varies according to the purpose of the required mission of the UAV system. The main task of the mission normally forces to use specific, narrow focused type of sensors that need to be carried by the UAV. For this reason we decide to develop a modular, service based data management architecture, which can provide all desired functionality to complete the mission successfully and enable the operator to control the UAV system reliably and comfortably.

Management and distribution of data provide an interesting field of research in different domains, ranging from hardware architecture over communication and network architecture, resource awareness to categorization, deployment, flow control and representation of data.

OVERVIEW OF OUR UAV SYSTEM FUNCTIONALITY AND LIMITATIONS

At our department we are focused on the development of an autonomous flying airship. We use a robotic airship (BLIMP) filled with helium which is 9 meters long with maximal 2.5 meters diameter and its payload is about 5 to 6 kilograms.

The required task of the mission has a big influence on the desired functionality of an UAV system. The functionality of an UAV system and its data management architecture from various aspects could be split into several levels.

The first and base level of the main functionality of an UAV system and its data management architecture is of course to

provide the possibility to be controlled by the operator during the flight. This requirement is closely coupled with communication capabilities and the on-board autonomy of the system. Depending on the increasing on-board autonomy there appear three possible control modes.

- Remote Control
- Teleoperation
- Automatic mode



Fig. 1 Robotic Airship during a field test in Hemer, Germany, 2009

In the remote control mode the data management architecture provides to transmit primer and necessary data or commands from the operator to UAV's actuator system. In this case the level of autonomy is very low and the operator has to control the UAV manually using a remote control. This flying mode delivers many constraints depending on a potential application of the UAV. In our case one of the main limitations is that controlling the airship manually via remote control is a quiet difficult task for the pilot. It assumes to have some skills to keep the airship flying smoothly by rough weather conditions. Moreover the airship has to be

visible for the pilot permanently during the flight. On the other hand this type of control is useful for some maneuvers which could be problematic for the autonomous control algorithms as well as a very important backup control system by any unexpected failure of the on-board systems.

The teleoperation mode increases the on-board autonomy into a semi-autonomous control, which means that the operator is able to control the UAV via joystick with support of basic automatic control algorithms to assure the desired course and altitude during the flight. The operator doesn't need to promptly react on each disturbance appeared during the flight and doesn't need to keep a visual contact with the UAV as well. It effects also the data management architecture, because it is necessary to deliver and intuitively represent an information of the actual position and many additional information from UAV's sensor system, which can give the operator feedback and a better overview of the environment. In this mode it is very important to provide and keep reliable communication with low latency between the UAV and the operator, because the system is usually not able to make any decision itself. In fact many UAV systems are working properly on semi-autonomous mode of control, because it enables them to complete many types of missions successfully.

The full automatic mode is the highest level of autonomy. The autonomy of an UAV relieves the operator of controlling the UAV and enables him to concentrate just on the main task of the mission. The operator just pickups the desired points to fly, the system calculates a flight path and flies it over. In this mode the control system is enhanced with a supervisory control and decision algorithms with support of various integrated modules like a path planner and collision avoiding algorithms. Although all information, which give a feedback for the operator like an actual position, are still important, this level of autonomy allows the system to make some pre-programmed decision by itself. With the support of a more advanced on-board intelligence, a short communication delays or disconnections between the UAV and base station communication does not necessary lead to any critical situation or damage of the UAV. Such a sophisticated control system enables the UAV to be applied in some kind of missions, where the capability of low latency communication is limited or restricted. On the other hand, in this mode various HW and SW modules are coupled together. These modules normally depend on each other and for the correct functionality of the whole system the effective exchange of actual data or the event messages between these collaborative modules is one of the major challenges for the data management architecture.

The second level of the functionality of an UAV system and its data management architecture is to enable the whole system to fulfill the required task of the mission successfully. As we mentioned in the introduction part of this paper, generally the tasks could be focused on monitoring some areas like an observer or any data acquisition tasks. For the data management architecture it usually means that various additional data from third-party sensor systems need to be transferred online to some operation centre or collected to be post-processed and analyzed after the mission. Our department was a member of two research projects: International mine detection and removal (iMR) and International forest fire combat (iWBB).

In the first project the role of our airship was to be used as an inspection vehicle to ensure the total destruction of mines by a high-energy laser system. Therefore, it was equipped with a remote camera system [1]. This camera system is able to stream the video signal online to the base station and even to remote control the camera's viewpoint by the operator.



Fig. 2 The view on a destroyed dummy landmine which is captured by the on-board camera system during a test flight in Hemer, Germany, 2009

In the second project our airship was used as a mobile sensor platform for the third-party heat, smoke, gas detection and pollution monitoring sensoric systems. Another important aspect of our research was the establishment of redundant communication links between the airship and the base station, and to create a data interface to the project operation centre [2].

The airship equipped with such a sensoric system is a valuable source of information about the actual situation for the firemen.



Fig. 3 The detection and monitoring of fire using the third-party sensoric system during the test flight in Hemer, Germany, 2010

As you can see in Figure 3 the airship was measuring data using these sensors to detect and monitor the fire. These data were online transferred to the project operation centre using the communication link of the airship's communication system.

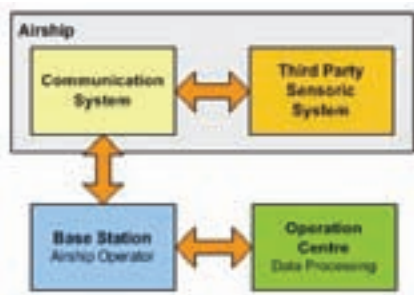


Fig. 4 Schematic of the data flow between the airborne sensoric system and the operation and data processing centre located on the ground.

SYSTEM ARCHITECTURE AND ENGINEERING TRADEOFFS

During the design of our system architecture some limitations have to be taken to consideration. Our airship is powered by electric energy. The source of this power is just the battery system of the airship. Minimizing the power consumption of all embedded systems is a strict requirement, in order to save the energy that could be used to power the airship actuators and to maximize the time of flight as much as possible. The second very important limitation is the restricted payload of the airship.

The primary components of the hardware platform consist of a few modules. The first module represents an appropriate pre-designed control board with a DSP processor to handle a low level control of the airship in real-time. It has very low power consumption and it is optimized to minimal size and weight. Moreover it enables a rapid prototyping using the Matlab development environment. It includes the navigation system, which couples various sensors like GPS and an inertial measurement unit (IMU), which uses a combination of accelerometers and gyroscopes. This board is the main source of telemetric data for the operator as well as the superior system to process his commands. It is the most important node in the data management architecture.

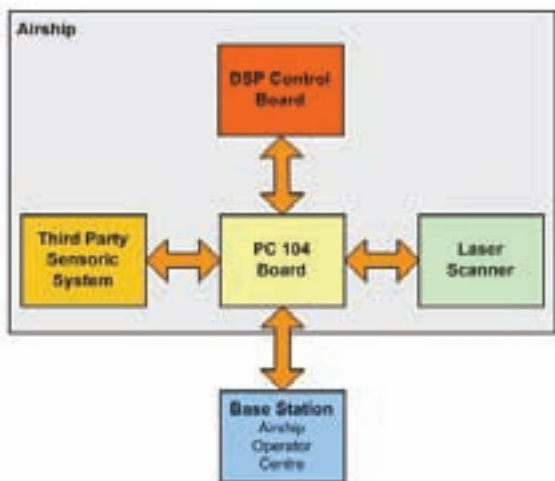


Fig. 5 The primary components of the hardware platform.

The next module is the embedded PC 104 board, which is dedicated to manage the data flow control and reliable distribution of data between all the collaborative modules and the base station on the ground. It is based on the Intel x86 processor architecture and uses a 500 MHz AMD Geode processor with 1 GB of RAM. The system uses a 32 GB compact flash card (CF) for storage and has many HW interfaces like dual Ethernet, RS232 or USB ports, to be

able to connect additional third-party sensors or devices. This platform is a balanced trade-off between the needed resources and power consumption. Nowadays mostly expanding boards are based on Intel Atom platform with more resource capability, but usually coupled with higher power consumption. For the purpose of a data flow manager, communication router and data collector we rather took the less powerful variant.

The base station, as an airship operator centre, is located on the ground. This module consists of a powerful mobile workstation PC equipped with pre-developed SW modules to keep the reliable communication between the airship and the base station and a special control panel module for the airship.

Our last tested additional module is the laser scanner UTM-30LX produced by Hokuyo Automatic CO., LTD. It is a relatively small type of a laser scanner. It weights just about 210 grams and its power consumption is less than 8 W, so it is an ideal device for a middle-sized UAV like our airship. It is intended to be used for the purpose of making 3D scans of the environment the airship is flying over and for obstacles detection, as an information source for the collision avoiding algorithms.

CATEGORIZATION AND DEPLOYMENT OF DATA

By design of data management and communication architecture for tele-operated UAV system it is very important to specify all the data sources. The main source of data is of course the navigation system of the airship which delivers information of the actual position, altitude, orientation and velocities of the airship. As a second source of information and event messages the control panel module is located on the base station. It fires control commands and event messages from the operator to the airship's on-board control system.



Fig. 6 Schematic of the data relations between the system components.

The next source of valuable information is the battery and actuator system status observer that gives an information of the actual battery capacity, voltage, current and propellers rpm. Other sources of data are also the laser scanner system and third-party sensoric systems, which usually use our

communication platform just to transfer or collect measured data.

We can categorize the data to different types based on their utilization. Some applications like navigation and control system need to execute their calculations in real-time. For these applications a continuous exchange of their data in real-time is necessary. This real-time requirement is realized by joining and executing these application modules on the same real-time platform.

Real-time delivery: Some applications require that a message must be delivered within a specified time, otherwise the message becomes useless or its information content is decreasing after the time bound. [4]

Another type of data represents event based messages. These messages can be usually invoked by the operator like a control or setup commands, path-planner recalculated flight-points or any obstacle detection event fired by collision avoiding system. These messages usually don't appear in every real-time sample, but a low latency delivery is necessary to keep their relevancy. Moreover their delivery status has to be checked. This approach is used also for the main exchange of information between airship on-board systems and the base station on the ground.

The next type of data is usually collected and stored on a storage device of the PC 104 board. These data are further used for post processing and analysis. The source of these data is the laser scanner system and any third-party sensoric system. It often handles large amounts of data. In some cases it is required to transfer these data online to the base station or any mission operation centre on the ground. In this case the communication link needs to handle larger amounts of data, but the low latency by data delivery is not critical.

FLOW CONTROL AND REPRESENTATION OF DATA

In the previous few sections the primary requirements and constraints of data management architecture, system architecture and data categorization have been discussed. But the most important part of data management architecture is to provide a reliable communication between all of these collaborative modules. This capitol explains the main concept of the communication architecture.

Communication Architecture

In the beginning several communication approaches have been tested to find out the best results. The main criteria of a wireless network structure are reliability of a communication channel, communication range, baud rate and latency by transferring data packets. Moreover all the devices have to be certified by European Regulations.

The ad-hoc network structure was chosen as the first communication approach. In this case the network is decentralized and does not rely on a pre-existing network infrastructure. Communication between the nodes in the network is realized just like a point-to-point data link. This type of wireless network structure fits the requirements of communication architecture for the UAV system, because there are just two nodes which need to communicate with each other. It is the communication node located on the base station on the ground and the on-board communication node of the UAV system. As the first type of wireless network a popular Wi-Fi IEEE 802.11 b/g standard transmitting in the 2.4 GHz frequency band with allowed transmit power of 100 mW has been chosen. Several tests have been realized to prove our main

criteria. The best result was the baud rate around 10 Mbit/s and very low latency by transferring data packets, but even in the line of sight the communication range was just about a 100 to 190 meters by using 9 dB Omni-directional antennas. As a second difficulty by using this wireless network type appears a problem with the reliability of a communication channel. After disconnection caused by coming out of the communication range and returning back into it, the Wi-Fi device was not able to establish the communication link before reaching a very near bound of about a 50 meters from base station. It means that a save operation area for a flying UAV to keep reliable communication channel has to be restricted even to a half of the available communication range. As a second type of wireless communication a radio-modem transmitting in ISM 868 frequency band with 250 mW transmit power (by European regulations is allowed up to 500 mW) was applied. During some tests the same criteria have been evaluated. In this case the baud rate was just 28.8 kbit/s with latency about 120 to 250 ms, but the communication range in the line of sight was about 1500 meters by using 5 dB Omni-directional antennas. The reliability of the communication channel was excellent.

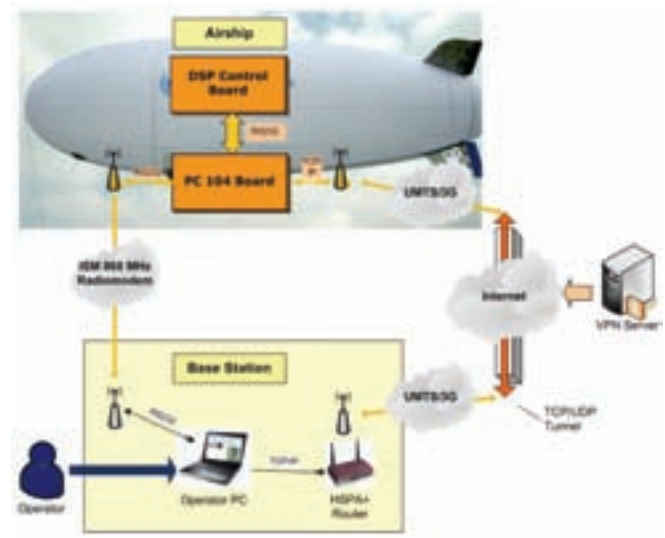


Fig. 7 Schematic of the Communication Architecture.

For the second approach we decided to use an already existing network infrastructure of the Universal Mobile Telecommunications System (UMTS). This type of 3G network infrastructure is nowadays provided by any mobile telecommunication operator. The client is able to connect to internet using resources of this infrastructure. On the other hand this internet connection is provided by the telecommunication operator as a paid service and there is no direct possibility to make a point-to-point data link. To be able to communicate from one UMTS network device on the base station to another device connected to the PC 104 board on the airship a virtual private network (VPN) has to be established. For this purpose a VPN server with a public IP address is used. The same criteria have been evaluated as well as in the case of ad-hoc networks. The baud rate, reliability of communication channel and data delivery latency depends of the actual signal coverage and of network type. The typical values are presented in the table Tab. 1. Especially the data delivery latency varies depending on the actual network link capacity and of course on the delays caused by overhead of the necessary VPN network, which routes the data via VPN server.

This type of network connection is in our communication architecture primary used as a redundant communication channel. So the whole communication architecture, illustrated on the figure 7, consists of the communication channel based on point-to-point communication via radio-modem,

and as a second redundant channel the communication via UMTS is used.

	Downlink	Uplink	Latency
HSDPA	7.2 Mbit/s	3.6 Mbit/s	100-300 ms
UMTS	384 kbit/s	128 kbit/s	200-1000 ms
EDGE	236 kbit/s	59.2 kbit/s	300-2500 ms
GPRS	60 kbit/s	40 kbit/s	400-3000 ms

Tab. 1 Typical downlink, uplink and latency values of most used GSM standards.

Data Flow and Reliability

The data management architecture has to manage and to keep reliable data flow with the base station using the pre-designed communication architecture. Various data have to be routed via these two communication channels by specific criteria. The very important telemetric data, control commands from the operator with a higher priority needs to be transferred with low latency. These information and messages are usually smaller data-packets which are transferred via a point-to-point radio-modem communication channel. For the data, which low latency is not such a limiting factor like data measured by third-party sensors or large amounts of collected data, the second UMTS communication channel will be used. The UMTS communication channel will be used also as a backup system in situations by radio-modem communication interruption. The pre-developed data router manager handles this functionality. The data flow between the components is realized as a service oriented architecture (SOA). It is based on a pre-developed lightweight protocol applied mainly over TCP protocol, so it uses primary a TCP socket interface.

When some component needs a functionality not provided by itself, it asks the system for the required service. If other component of the system has this capability, their location will be provided and finally the client component can consume the service using the common interface in the provider component. The interface of a SOA component must be simple and clear enough to be easily implemented in different platforms both hardware and software. [3]

On the base station is also a pre-developed data hub server as a service provider for multiple clients located on the ground: like control panel component on the base station or any third-party clients, which use a service to become the data from its sensoric system located on the airship as it is illustrated on the figure 4. The data hub server checks also a delivery status of each message and control the network delay measuring the message delivery with time constraints.

Data Representation

The next challenge for the data management architecture is the effective and intuitive representation of distributed data for the operator. For this purpose a special control panel module for the airship as a graphical user interface (GUI) has been developed. It is the important module in the system architecture used for interaction between the operator and the airship.

The control panel provides the input and output capabilities. The input functionality allows the operator to control the airship. It enables him to select the airship control mode, setup some system options. The output functionality provides

the indication of important data like information of the actual position, altitude and orientation of the airship.

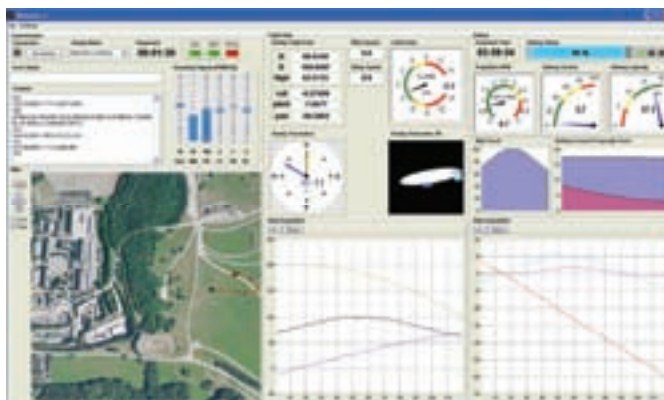


Fig. 8 GUI of the airship's control panel module.

For a better overview for the operator the airship orientation is visualized with the help of a three-dimensional virtual reality model. The actuating signals are indicated as well as the information of the airship systems and batteries state. There is also a possibility to record the measuring data using the pre-developed data acquisition module.

The actual position is displayed in the two-dimensional (2D) map, which is geo-referenced in Universal Transverse Mercator (UTM) geographic coordinate system. It is a very practical grid-based 2D Cartesian coordinate system. The position on the Earth is referenced in the UTM system by the UTM zone, and the easting (E) and northing (N) coordinate pair. [7]

The navigation system of the airship calculates the actual position in local north, east, down (NED) coordinates as it is illustrated in Figure 9.

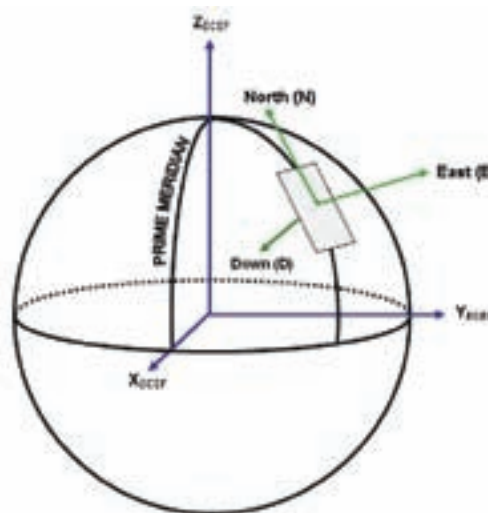


Fig. 9 NED Coordination system.

This local coordinate system needs to be referenced to specify its location on the surface of the Earth. For this purpose the navigation system sets the reference location in Earth-Centered-Earth-Fixed (ECEF) conventional terrestrial coordinate system and this information is sent to the control panel. The reference point is displayed on the map of the control panel. The transformation from ECEF coordinate system to UTM is necessary. As a first step the position needs to be converted from ECEF to geodetic coordinates

WGS84 (latitude ϕ , longitude λ , height h) using the Kaplan algorithm. Afterwards the conversion from geodetic coordinates (latitude ϕ , longitude λ) to UTM coordinates pair (E,N) and the UTM zone is calculated.

3D SCANNING OF THE EARTH SURFACE BY A LASER SCANNER

A typical data acquisition task for our airship could be a scanning of the Earth's surface by a laser scanner. The laser scanner is an additional sensor installed on-board. A basic principle is based on measuring the distance of the scanned points from chosen reference point that is usually placed over the surface. Knowing its exact position and having a possibility to measure its distance precisely to all points of the scanned surface and direction of the measurement, it is possible to calculate exactly their positions.

Choice Of The Measurement Method

A 3D scanning of the surface based on reflection of waves can be carried out in several ways: By sonar using ultrasonic waves, or by radar working with electromagnetic waves. The distance from a reference point may also be determined using stereoscopy, whereby it is evaluated by comparing two pictures of an object made from two reference points. This leads to different angles of view. Precision of all these systems is given by the used wave lengths and by their propagation. Systems like sonar have a relatively slow response speed and a relatively low resolution not enabling to see details of the scanned surface. A reasonable improvement was brought by laser technology guaranteeing high linearity of the ray movement, its fast speed (above 300 000 m/s) and measurement of details in range of millimeters.

By scanning the Earth's surface, the resulting precision depends highly on precision of the reference point localization and orientation of the beam. The final resolution will directly depend on the density of emitting and orientation of the beams.

Measurement Principle

Assume a laser device emitting laser beams oriented in a plane. That means that this system is able to scan just information about a curve lying in this plane. When wishing to scan the entire space, such scan must be repeated in infinitely many other planes. In order to get stereoscopic information the measurement must be repeated at least from one additional place. Such a measurement can e.g. be based on a translation movement, when the laser moves its reference position and the beam falls to the surface from other reference positions.

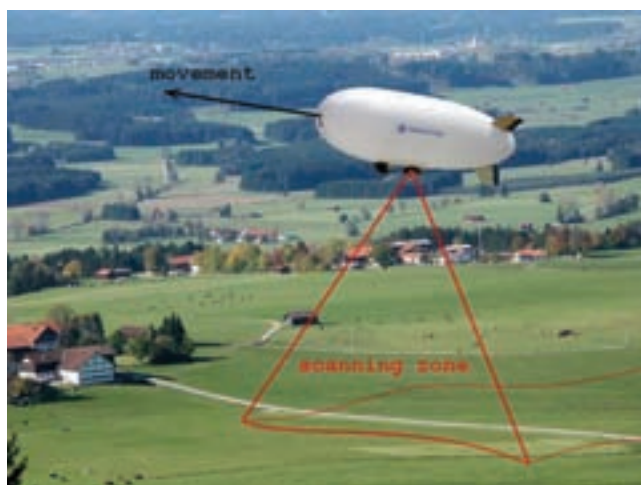


Fig. 9 Earth's surface scanning illustration by a laser scanner.

During the measurement, the reference position and orientation of the beam must be measured in order to enable the

calculation of the points of reflection. This information is usually achieved by a measurement and recording system completing the laser measurement by information about the laser reference position in 3D coordinating system $[x,y,z]$, as well as angular orientation of the beam in all three axes X, Y, Z denoted usually as Roll, Pitch, and Yaw.

Data Identification

For this task two frames are given: navigation frame (FRAME) and body frame (BODY). Data acquisition is considered in body frame. For following representation of measured data, they have to be transformed into a global navigation frame.

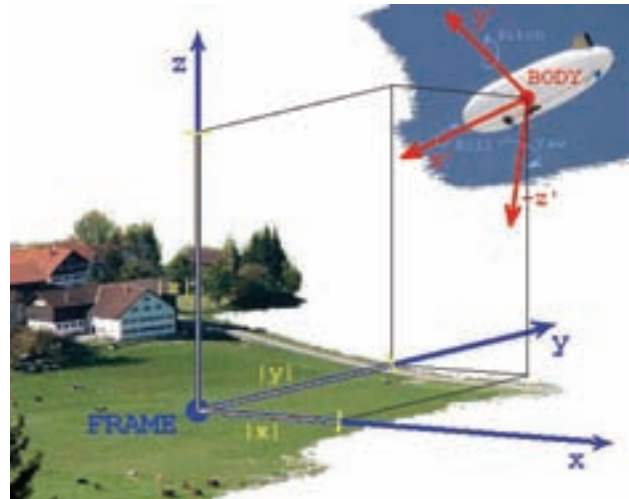


Fig. 10 Illustration of the coordinate systems BODY and FRAME.

The body frame may freely move and rotate in the navigation frame, i.e. it may change the coordinates of its origin in navigation frame, as well as orientation of its axes. The navigation frame is usually considered to be static.

While scanning laser beams are emitted with the origin placed to the origin of body frame $[0,0,0]$.

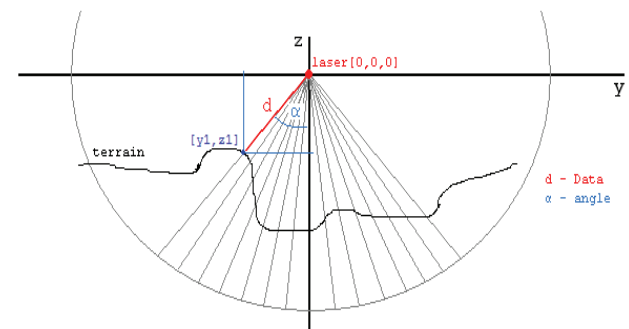


Fig. 11 Scanning principle.

These always lie in a plane YZ. After being reflected by the Earth's surface, its points are calculated according to:

$$y1 = d * \sin\alpha \tag{1}$$

$$z1 = d * \cos\alpha \tag{2}$$

d – Measurement of the scanned point from laser.

α – Angle between zero reference position and the emitted beam.

Under zero reference position we will understand beam orientation parallel to axis Z, i.e. vertically to the Earth in direction of $-Z$. While looking into the X direction, deviation

from the zero position do the right means deviation under a negative angle α . Deviation to the left corresponds to positive angles. This sign convention enables an easy calculation of Y coordinates. For such a measurement, the first X-coordinate will be zero, i.e. the measurement will be represented as [0,y,z].

Known coordinates in body frame are necessary to transform into navigation frame coordinates. The coordinate transformation is calculated by means of the direction cosine matrix (DCM). The DCM matrix performs the coordinate transformation of a vector in navigation frame axes (ox_0, oy_0, oz_0) into a vector in body frame axes (ox_3, oy_3, oz_3).

$$\begin{bmatrix} ox_3 \\ oy_3 \\ oz_3 \end{bmatrix}_{BODY} = DCM \begin{bmatrix} ox_0 \\ oy_0 \\ oz_0 \end{bmatrix}_{FRAME} \quad (3)$$

In the case of 3D space there exist in principle 12 different options how to describe a general body orientation by a sequence of three rotations. It could be demonstrated that the rotation according to X (Roll) axis, then with respect to Y (Pitch) axis and finally with respect to Z (Yaw) axis is different from that one with the same rotation in Roll, Pitch and Yaw carried out in different order. DCM transformation matrix is defined by combination of three axis transformation matrices M(X), M(Y), M(Z).

In our case the rotation angles in body frame are known and we need to carry out the inverse transformation into the navigation frame coordinates. This can be done according to:

$$\begin{bmatrix} ox_0 \\ oy_0 \\ oz_0 \end{bmatrix}_{FRAME} = DCM^{-1} \begin{bmatrix} ox_3 \\ oy_3 \\ oz_3 \end{bmatrix}_{BODY} \quad (4)$$

By translation movement of the body frame origin, the transformation is calculated as follows:

$$\begin{bmatrix} ox_0 \\ oy_0 \\ oz_0 \end{bmatrix}_{FRAME} = DCM^{-1} \begin{bmatrix} ox_3 \\ oy_3 \\ oz_3 \end{bmatrix}_{BODY} + \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{POSITION} \quad (5)$$

In the navigation frame the points are recorded in a way that corresponds to their position within the scanned surface.

System Implementation

The type of used laser scanner was briefly described in the section 3. Up to now a few first tests have been done mostly in laboratory conditions. The laser scanner enables a measurement in the range from 30 to 60 meters (m). The final results depend on the surface reflection capability, whereby precision given by the producer is for measurement up to 10 m in range of 0.03 m, and for measurement up to 30 m in the range up to 0.05 m. The first statement was partially confirmed by a measurement repeated several times in laboratory conditions, whereby the maximal difference of two measured samples was 28 mm. Of course, this does not yet guarantee the absolute measurement precision, but characterizes repeatability of the measurement.

During the tests the laser scanner was installed 1.35 m over the surface on a moving platform which enables to control the translation in axis X. The position x values in body frame X axes was incrementally preset in range of -0.9 m up to 0.9 m. Laser beam was emitted in the plane YZ.

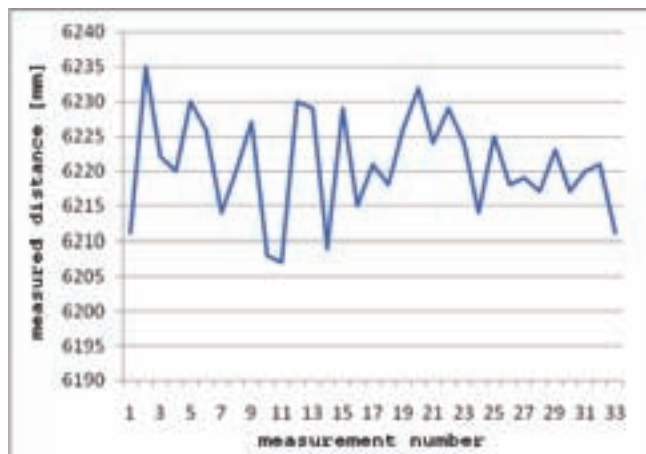


Fig. 12 Results of the measurement precision for 33 realized experiments.

Pitch was changed in the range from -32° up to 32° . It corresponds to changes of Y value within the range from -0.81 m up to 0.81 m for the given distance of the body frame origin respectively laser scanner over the scanned surface. These tests were carried out by scanning two boxes and one box with banked surface and decreasing height from 0.245 m, 0.18 m to 0.075 m. The results, displayed in Figure 13, illustrate a relatively correct reproduction of the scanned surface distorted partially by the measurement noise. For noise elimination some various filtration procedures may be applied.

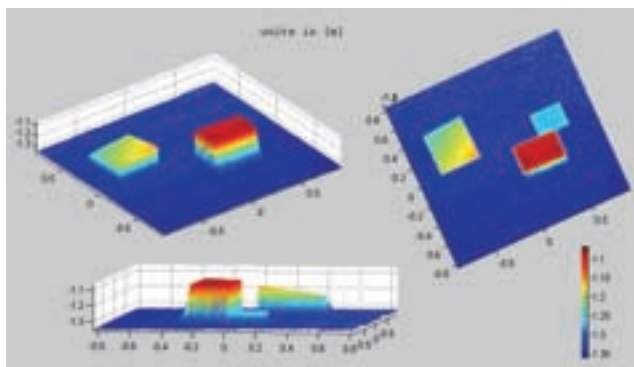


Fig. 13 Results of scanning.

The determination of the laser position, respectively the body frame origin and its rotation angles (roll, pitch, yaw), has a dominant influence on the final measurement precision.

CONCLUSION

This paper has introduced our approach to design a modular, service based data management architecture for our UAV system, in our case the robotic airship, to provide the data management and routing as well as a reliable distribution of information between all the collaborative system modules. For this purpose the system and communication architecture has been described. As a very important component of the complex system the control panel module for intuitive data representation and control has been developed. Such architecture enables to establish a robust communication structure, comfortable control capability for the operator and even to provide a service based interface for a mission operation centre with an opportunity of online distribution of important data from the airborne sensoric system as a valuable source of information. The main constraints and experiences by the design and practical experiments have been mentioned as well. The most valuable experiences were gained by fulfilling the major tasks in our two research projects, where design and development of the archi-

ecture was formed by the real constraints and problems, which appear during the tests.

As a future work some enhancement of communication data flow and a more robust reliability are planned. The communication structure could be used as a source of information to any leashing algorithms.

Leashing is then performed by having the robot react to low message rates by moving towards the base in order to improve the communication. [5]

The data throughput by a communication channel with low band rate like our radio-modem data link could be enhanced by any real-time data compression methods.

Huffman style compression scheme exploits temporal locality and delta compression to provide better bandwidth utilization, thus reducing latency for real time applications. [6]

The next interesting mission for our airship is a 3D scanning of the Earth surface by the additional sensor system: the laser scanner. The last section introduces the measurement method and our first results in this problem.

REFERENCES

[1] GERKE, M. (2009). Mine detection and removal (iMR). In: Annual Report, Fakultät für Mathematik und Informatik, FernUniversität in Hagen. http://prt.fernuni-hagen.de/forschung/ANNUALS/ET09_PRT.pdf

[2] GERKE, M. (2009). Forest fire prevention and extinction (iWBB). In: Annual Report, Fakultät für Mathematik und Informatik, FernUniversität in Hagen. http://prt.fernuni-hagen.de/forschung/ANNUALS/ET09_PRT.pdf

[3] PASTOR, E. LOPEZ, J. ROYO, P. (2006). A Hardware/Software Architecture for UAV Payload and Mission Control. In: Proceedings of the 25th Digital Avionics Systems Conference, October 15-19, 2006, Portland, Oregon.

[4] KANSAL, P. KANSAL, D. ARUN, B. (2010). Compression of Various Routing Protocol in Wireless Sensor Network. In: International Journal of Computer Applications (0975-8887) Vol. 5 – No.11.

[5] HAUERT, S. LEVEN, S. ZUFFEREY, J.-Ch. FLOREANO, D. (2010) Communication-based Leashing of Real Flying Robots. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation. Anchorage Convention District, May 3-8, 2010, Anchorage, Alaska, USA

[6] SZALAPSKI, T. MADRIA, S. (2010). Real-Time Data Compression in Wireless Sensor Network. In: MDM'10 Proceedings of the Eleventh International Conference on Mobile Data Management. Kansas City, Missouri.

[7] Wikipedia [online]. Universal Transverse Mercator coordinate system. <http://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system>. Retrieved 2011-01-15

[8] GERKE, M. MASÁR, I. (2007). Modeling, Sensorics and Control of a Robotic Airship. In: IAESTED International Conference on Robotics and Applications. Wuerzburg, Germany.

[9] GERKE, M. MASÁR, I. HUBA, M. (2008). Dynamical Modeling and Virtual Reality Simulation of a small Robotic Airship. In: 5th International Conference on Electrical and Power Engineering. Iasi, Romania.

Abstract

Nowadays, more frequently than ever, the unmanned aerial vehicles (UAVs) are used effectively as mobile sensor platforms. The UAV system equipped with an airborne camera and special sensors is a valuable source of various important, information helping to build an actual overview of an environment. It can take place like an observer in disaster situations as well as a special mobile monitoring device which is able to collect required data from a predefined area. This paper introduces our approach to design effective data management architecture to be able to manage, reliably distribute and represent different types of measured data with taking many aspects and limitations of the tele-operated UAV system to consideration.

Ing. Pavol Bahník*, Jakub Pilka**

FernUniversität in Hagen *
Fakultät für Mathematik und Informatik
Lehrgebiet Prozesssteuerung und Regelungstechnik
Universitätsstr. 27, PRG
D-58097 Hagen
Tel.: +49 2331 9871105
Fax: +49 2331 987354
E-mail pavol.bahnik@fernuni-hagen.de

Slovak University of Technology **
Faculty of Electrical Engineering and Information Technology
Institute of Control and Information Technology
Ilkovičova 3
SK-812 19 Bratislava
E-mail xpilkaj@stuba.sk

Digital Self-tuning Smith Predictor Based on Pole Assignment Approach

Vladimír Bobál, Petr Chalupa, Petr Dostál, Michal Brázdil

Abstract

Time-delays (dead times) are found in many processes in industry. Time-delays are mainly caused by the time required to transport mass, energy or information, but they can also be caused by processing time or accumulation. The contribution is focused on a design of algorithms for self-tuning digital control for processes with time-delay. The algorithms are based on the some modifications of the Smith Predictor (SP). One modification of the SP based on the digital PID controller was applied and it was compared with new designed modification based on polynomial (pole assignment) approach. The program system MATLAB/SIMULINK was used for simulation verification of these algorithms.

Key words: Time-delay, Smith predictor, Self-tuning control, ARX model, Recursive identification, Pole assignment

Introduction

Time-delays appear in many processes in industry and other fields, including economical and biological systems (see [1]). They are caused by some of the following phenomena:

- the time needed to transport mass, energy or information,
- the accumulation of time lags in a great numbers of low order systems connected in series,
- the required processing time for sensors, such as analyzers; controllers that need some time to implement a complicated control algorithms or process.

Consider a continuous time dynamical linear SISO (single input $u(t)$ – single output $y(t)$) system with time-delay T_d .

The transfer function of a pure transportation lag is $e^{-T_d s}$, where s is complex variable. Overall transfer function with time-delay is in the form

$$G_d(s) = G(s)e^{-T_d s} \quad (1)$$

where $G(s)$ is the transfer function without time-delay.

Processes with significant time-delay are difficult to control using standard feedback controllers mainly because of the following:

- the effect of the disturbances is not felt until a considerable time has elapsed;
- the effect of the control action requires some time to elapse;
- the control action that is applied based on the actual error tries to correct a situation that originated some time before.

The problem of controlling time-delay processes can be solved by some control methods using

- PID controllers;

- time-delay compensators;
- model predictive control techniques.

It is clear that many processes in industry are controlled by the PID controllers. When the process contains a time-delay, the tuning of the PID controller is difficult. The most popular tuning rules for processes with small time-delay were proposed by Ziegler and Nichols 1942. Several methods for new tuning rules were proposed for stable and unstable processes with time-delay. A presentation and review of some these methods are introduced in [2]. When a high performance of the control process is desired or the relative time-delay is very large, a predictive control strategy must be used.

Predictor based control strategy have been used in many control applications. The performance of the closed-loop system can be improved by using a predictor structure in two main causes: a) when the process has a significant time-delay; when the future reference is known. The predictive control strategy includes a model of the process in the structure of the controller. The first time-delay compensation algorithm was proposed by Smith [3]. This control algorithm known as the Smith Predictor (SP) contained a dynamic model of the time-delay process and it can be considered as the first model predictive algorithm.

Historically first modifications of time-delay algorithms were proposed for continuous-time (analogue) controllers. On the score of implementation problems, only the discrete versions are used in practice in this time.

The majority of processes met in the industrial practice have stochastic characteristics and eventually they embody non-linear behaviour. Traditional controllers with fixed parameters are often unsuitable for such processes because their parameters change. One possible alternative for improving the quality of control for such processes is the use of adaptive control systems. Different approaches were proposed and utilized. One successful approach is represented by self-tuning controller (STC). The main idea of an STC is

based on the combination of a recursive identification procedure and a selected controller synthesis. Some STC modifications of the digital Smith Predictors are designed and verified by simulation in this paper.

The paper is organized in the following way. The problem of a control of the time-delay systems is described in the Introduction. The general principle of the Smith Predictor is described in Section 1. The discretization of analogue version and principle of digital Smith Predictor is introduced in Section 2. Two modifications of digital controllers that are used for self-tuning versions SPs are proposed in Section 3. Section 4. contains brief description of the recursive identification procedure. Simulation configuration is presented in Section 5. Results of simulation experiments are summed in Section 6.

1. Principle of Smith Predictor

The principle of the SP is shown in Fig. 1. It can be divided into two parts – the primary $G_c(s)$ controller and predictor part. This algorithm was primarily designed for continuous time PID controller. The predictor is composed of a model of the process without time delay and $G_m(s)$ (so called as the fast model) and a model of the time delay $e^{-T_d s}$. Then the complete process model is

$$G_p(s) = G_m(s)e^{-T_d s} \quad (2)$$

The fast model $G_m(s)$ is used to compute an open-loop prediction. The difference between the output of the process $y(t)$ and the model including time delay $\hat{y}(t)$ is the predicted error $\hat{e}_p(t)$ as shown is in Fig. 2, where $u(t)$, $w(t)$, $e(t)$ and $e_s(t)$ are the control signal, reference signal, the error and the noise. If there are no modeling errors or disturbances, the error between the current process output and the model output will be null and the predictor output signal $\hat{y}_p(t)$ will be the time-delay-free output of the process. Under these conditions, the controller $G_c(s)$ can be tuned, at least in the nominal case, if the process had no time delay.

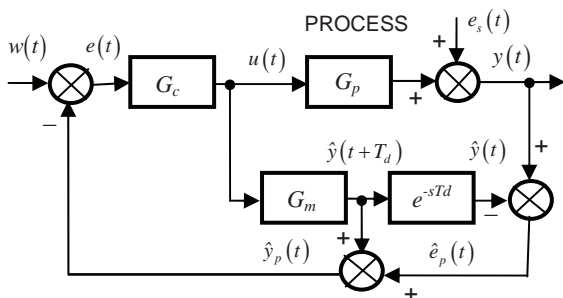


Fig. 1 Block diagram of an analogue Smith Predictor

The Smith Predictor structure for the nominal case (without modelling errors) has three fundamental properties: time-delay compensation, prediction and dynamic compensation.

2. Digital Smith Predictor

Although time-delay compensators appeared in the mid 1950s, their implementation with analogue technique was very difficult and these were not used in industry. Since 1980s digital time-delay compensators can be implemented. In spite of the fact that all these algorithms are implemented

on digital platforms, most works analyze only the continuous case. The digital time-delay compensators are presented e.g. in [4] – [6]. The discrete versions of the SP and its modifications are suitable for time-delay compensation in industrial practice. Most of authors designed the digital SP using discrete PID controllers with fixed parameters. However, the SP is more sensitive to process parameter variations and therefore requires an auto-tuning or adaptive approach in many practical applications [7, 8]. In [9], the structure of the discrete disturbance observer time-delay compensator is analyzed.

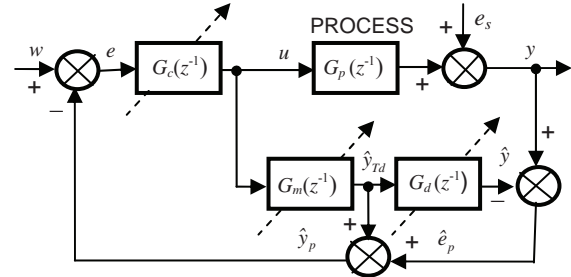


Fig. 2 Block diagram of a digital Smith Predictor with tuning

2.1 Structure of Digital SP

The block diagram of a digital SP (see [4], [7], [8]) is shown in Fig. 2. The function of the digital version is similar to the classical analogue version. The block $G_m(z^{-1})$ in Fig. 2 represents process dynamics without the time-delay. The primary (main) controller $G_c(z^{-1})$ can be designed by the different approaches (for example digital PID control or methods based on algebraic approach). The outward feedback-loop through the block $G_d(z^{-1})$ in Fig. 2 is used to compensate for load disturbances and modelling errors. The dash arrows indicate the self-tuned parts of the Smith Predictor.

Most industrial processes can be approximated by a reduced order model with some pure time-delay. Consider the following second order linear model with a time-delay

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} z^{-d} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-d} \quad (3)$$

for demonstration of some approaches to the design of the adaptive Smith Predictor. The term z^{-d} represents the pure discrete time-delay. The time-delay is equal to dT_0 where T_0 is the sampling period.

If the time-delay is not an exact multiple of the sampling period T_0 , then dT_0 represents the largest integer multiple of the sampling period with remaining fractional deal absorbed into $B(z^{-1})$ using the modified z-transformation [10].

2.2 Identification of Time-delay

In this paper, the time-delay is assumed approximately known or possible to be obtained separately from an off-line identification using the least squares method [11]

$$\hat{\theta} = (F^T F)^{-1} F^T y \quad (4)$$

where the matrix F has dimension $(N-n-d, 2n)$, the vector y $(N-n-d)$ and the vector of parameter model estimates $\hat{\theta}$ $(2n)$.

N is the number of samples of measured input and output data, n is the model order.

Equation (4) serves for a one-off calculation of the vector of parameter estimates $\hat{\theta}$ using N samples of measured data. The individual vectors and matrix in equation (4) have the form

$$y^T = [y(n+d+1) \quad y(n+d+2) \quad \dots \quad y(N)] \quad (5)$$

$$\hat{\theta}^T = [\hat{a}_1 \quad \hat{a}_2 \quad \dots \quad \hat{a}_n \quad \hat{b}_1 \quad \hat{b}_2 \quad \dots \quad \hat{b}_n] \quad (6)$$

$$F = \begin{bmatrix} -y(n+d) & -y(n+d-1) & \dots & -y(d+1) \\ -y(n+d+1) & -y(n+d) & \dots & -y(d+2) \\ \vdots & \vdots & \dots & \vdots \\ -y(N-1) & -y(N-2) & \dots & -y(N-n) \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} u(n) & u(n-1) & \dots & u(1) \\ u(n+1) & u(n) & \dots & u(2) \\ \vdots & \vdots & \dots & \vdots \\ u(N-d-1) & u(N-d-2) & \dots & u(N-d-n) \end{bmatrix}$$

For the second order model (3) are these expressions e.g. for $N=10$

$$y^T = [y(5) \quad y(6) \quad \dots \quad y(10)]$$

$$\hat{\theta}^T = [\hat{a}_1 \quad \hat{a}_2 \quad \hat{b}_1 \quad \hat{b}_2]$$

$$F = \begin{bmatrix} -y(4) & -y(3) & u(2) & u(1) \\ -y(5) & -y(4) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ -y(9) & -y(8) & u(7) & u(6) \end{bmatrix}$$

Consider that model (3) is the deterministic part of the stochastic process described by the ARX (regression) model

$$y(k) = \frac{B(z^{-1})z^{-d}}{A(z^{-1})}u(k) + \frac{1}{A(z^{-1})}e_s(k) \quad (8)$$

where $e_s(k)$ is the random nonmeasurable component.

The ARX model (8) can be expressed as a stochastic difference equation

$$y(k) = -a_1y(k-1) - a_2y(k-2) + b_1u(k-d-1) + b_2u(k-d-2) + e_s(k) \quad (9)$$

The vector of parameter model estimates is computed by solving equation (4)

$$\hat{\theta}^T(k) = [\hat{a}_1 \quad \hat{a}_2 \quad \hat{b}_1 \quad \hat{b}_2] \quad (10)$$

and is used for computation of the prediction output

$$\hat{y}(k) = -\hat{a}_1y(k-1) - \hat{a}_2y(k-2) + \hat{b}_1u(k-d-1) + \hat{b}_2u(k-d-2) \quad (11)$$

The quality of ARX model can be judged by the prediction error, i.e. the deviation

$$\hat{e}(k) = y(k) - \hat{y}(k) \quad (12)$$

The prediction error plays a key role in identification of regressions model parameters derived from measured data. It is important for selecting the structure (order) of the regression model and a suitable sampling period. The quality of the model is also judged by the purpose for which it is used.

In this paper, the prediction error was used for suitable choice of the time-delay dT_0 . The LSM algorithm (4) – (7) is computed for several time-delays dT_0 and the suitable time-delay is chosen on the based of quality identification by using prediction error (12).

3. Algorithms of Digital Smith Predictors

3.1 Digital PID Smith Predictor (PIDSP)

Hang et al. [8], [12] used to design of the main controller $G_c(z^{-1})$ the Dahlin PID algorithm [13]. This algorithm is based on the desired close-loop transfer function in the form

$$G_e(z^{-1}) = \frac{1 - e^{-\alpha}}{1 - z^{-1}} \quad (13)$$

where $\alpha = T_0/T_m$ and T_m is desired time constant of the first order closed-loop response. It is not practical to set T_m to be small since it will demand a large control signal $u(k)$ which may easily exceed the saturation limit of the actuator. Then the individual parts of the controller are described by the transfer functions

$$G_c(z^{-1}) = \frac{(1 - e^{-\alpha}) \hat{A}(z^{-1})}{(1 - z^{-1}) \hat{B}(1)}; \quad G_m(z^{-1}) = \frac{z^{-1} \hat{B}(1)}{\hat{A}(z^{-1})}$$

$$G_d(z^{-1}) = \frac{z^{-d} \hat{B}(z^{-1})}{z^{-1} \hat{B}(1)} \quad (14)$$

where $\hat{B}(1) = \hat{B}(z^{-1})|_{z^{-1}=1} = \hat{b}_1 + \hat{b}_2$.

Since $G_m(z^{-1})$ is the second order transfer function, the main controller $G_c(z^{-1})$ becomes a digital PID controller having the following form:

$$G_c(z^{-1}) = \frac{U(z)}{E(z)} = \frac{q_0 + q_1z^{-1} + q_2z^{-2}}{1 - z^{-1}} \quad (15)$$

where $q_0 = \gamma$, $q_1 = \hat{a}_1\gamma$, $q_2 = \hat{a}_2\gamma$ using by the substitution $\gamma = (1 - e^{-\alpha}) / \hat{B}(1)$. The PID controller output is given by

$$u(k) = q_0e(k) + q_1e(k-1) + q_2e(k-2) + u(k-1) \quad (16)$$

3.2 Digital Pole Assignment Smith Predictor (PASP)

The second controller applied in this paper was designed using a polynomial approach. Polynomial control theory is based on the apparatus and methods of a linear algebra [14], [15]. The polynomials are the basic tool for a description of the transfer functions. They are expressed as the finite sequence of figures – the coefficients of a polynomial. Thus, the signals are expressed as infinite sequence of figures. The controller synthesis consists in the solving of linear polynomial (Diophantine) equations [16].

The design of the controller algorithm is based on the general block scheme of a closed-loop with two degrees of freedom (2DOF) according to Fig. 3 [17].

The controlled process is given by the transfer function in the form

$$G_p(z^{-1}) = \frac{Y_p(z)}{U(z)} = \frac{B(z^{-1})}{A(z^{-1})} \quad (17)$$

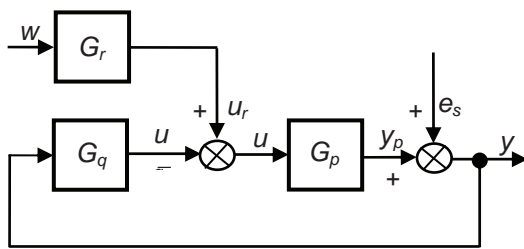


Fig. 3 Block diagram of a closed loop 2DOF control system

where A and B are the second order polynomials. The controller contains the feedback part G_q and the feedforward part G_r . Then the digital controllers can be expressed in the form of a discrete transfer functions

$$G_r(z^{-1}) = \frac{R(z^{-1})}{P(z^{-1})} = \frac{r_0}{1 + p_1 z^{-1}} \quad (18)$$

$$G_q(z^{-1}) = \frac{Q(z^{-1})}{P(z^{-1})} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{(1 + p_1 z^{-1})(1 - z^{-1})} \quad (19)$$

According to the scheme presented in Fig. 3 (for $e_s = 0$), the output y can be expressed as

$$Y(z^{-1}) = \frac{G_p(z)G_r(z)}{1 + G_p(z)G_q(z)}W(z^{-1}) \quad (20)$$

Upon substituting from Equation (17) - (19) into Equation (20) it yields

$$Y(z^{-1}) = \frac{B(z^{-1})R(z^{-1})}{A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1})}W(z^{-1}) \quad (21)$$

where

$$A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1}) = D(z^{-1}) \quad (22)$$

is the characteristic polynomial.

The procedure leading to determination of polynomials Q , R and P in (18) and (19) can be briefly described as follows (see [18]). A feedback part of the controller is given by a solution of the polynomial Diophantine equation (22). An asymptotic tracking is provided by a feedforward part of the controller given by a solution of the polynomial Diophantine equation

$$S(z^{-1})D_w(z^{-1}) + B(z^{-1})R(z^{-1}) = D(z^{-1}) \quad (23)$$

For a step-changing reference signal value, polynomial $D_w(z^{-1}) = 1 - z^{-1}$ and S is an auxiliary polynomial which does not enter into controller design.

A feedback controller to control a second-order system without time-delay will be derived from Equation (22). A system of linear equations can be obtained using the uncertain coefficients method

$$\begin{bmatrix} \hat{b}_1 & 0 & 0 & 1 \\ \hat{b}_2 & \hat{b}_1 & 0 & \hat{a}_1 - 1 \\ 0 & \hat{b}_2 & \hat{b}_1 & \hat{a}_2 - \hat{a}_1 \\ 0 & 0 & \hat{b}_2 & -\hat{a}_2 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ p_1 \end{bmatrix} = \begin{bmatrix} d_1 + 1 - \hat{a}_1 \\ d_2 + \hat{a}_1 - \hat{a}_2 \\ d_3 + \hat{a}_2 \\ d_4 \end{bmatrix} \quad (24)$$

where the characteristic polynomial is chosen as

$$D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3} + d_4 z^{-4} \quad (25)$$

For a step-changing reference signal value it is possible to solve Equation (23) by substituting $z = 1$

$$R = r_0 = \frac{D(1)}{B(1)} = \frac{1 + d_1 + d_2 + d_3 + d_4}{b_1 + b_2} \quad (26)$$

The 2DOF controller output is given by

$$u(k) = r_0 w(k) - q_0 y(k) - q_1 y(k-1) - q_2 y(k-2) + (1 - p_1)u(k-1) + p_1 u(k-2) \quad (27)$$

4. Recursive Identification Procedure

The regression (ARX) model of the following form

$$y(k) = \Theta^T(k)\Phi(k) + e_s(k) \quad (28)$$

is used in the identification part of the designed controller algorithms, where

$$\Theta^T(k) = [a_1 \quad a_2 \quad b_1 \quad b_2] \quad (29)$$

is the vector of model parameters and

$$\Phi^T(k-1) = [-y(k-1) \quad -y(k-2) \quad u(k-d-1) \quad u(k-d-2)] \quad (30)$$

is the regression vector. The non-measurable random component $e_s(k)$ is assumed to have zero mean value $E[e_s(k)] = 0$ and constant covariance $R = E[e_s^2(k)]$.

All digital adaptive SP controllers use the algorithm of identification based on the Recursive Least Squares Method (RLSM) extended to include the technique of directional (adaptive) forgetting. Numerical stability is improved by means of the LD decomposition [18], [19]. This method is based on the idea of changing the influence of input-output data pairs to the current estimates. The weights are assigned according to amount of information carried by the data.

5. Simulation Verification Adaptive Digital SP Controller Algorithms

Simulation is a useful tool for the synthesis of control systems, allowing one not only to create mathematical models of a process but also to design virtual controllers in a computer. The mathematical models provided are sufficiently close to a real object that simulation can be used to verify the dynamic characteristics of control loops when the structure or parameters of the controller change. The models of the processes may also be excited by various random noise generators which can simulate the stochastic characteristics of the processes noise signals with similar properties to disturbance signals measured in the machinery.

The above mentioned SP controllers are not suitable for the control of unstable processes. Therefore, three types of processes were chosen for simulation verification of digital adaptive SP controller algorithms.

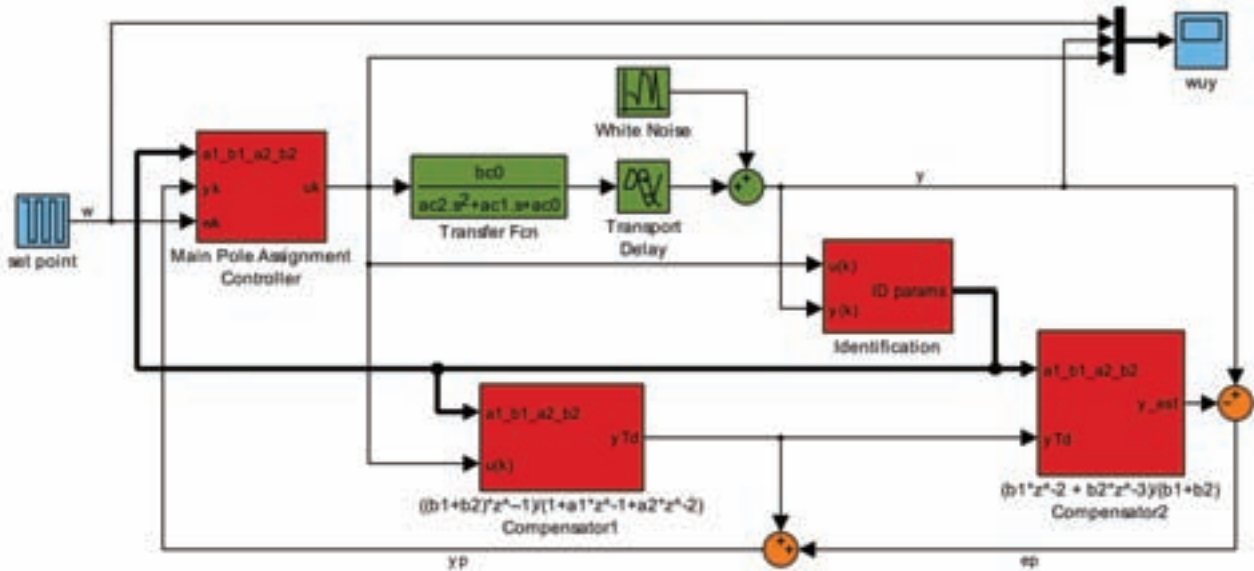


Fig. 4 Simulink control scheme

Consider the following continuous-time transfer functions:

- 1) Stable non-oscillatory $G_1(s) = \frac{2}{(s+1)(4s+1)} e^{-4s}$
- 2) Stable oscillatory $G_2(s) = \frac{2}{4s^2 + 2s + 1} e^{-4s}$
- 3) With non-minimum phase $G_3(s) = \frac{-5s+1}{(s+1)(4s+1)} e^{-4s}$

Let us now discretize them a sampling period $T_0 = 2s$. The discrete forms of these transfer functions are (see Equation (3))

$$G_1(z^{-1}) = \frac{0.4728z^{-1} + 0.2076z^{-2}}{1 - 0.7419z^{-1} + 0.0821z^{-2}} z^{-2}$$

$$G_2(z^{-1}) = \frac{0.6806z^{-1} + 0.4834z^{-2}}{1 - 0.7859z^{-1} + 0.3679z^{-2}} z^{-2}$$

$$G_3(z^{-1}) = \frac{-0.5489z^{-1} + 0.8897z^{-2}}{1 - 0.7419z^{-1} + 0.0821z^{-2}} z^{-2}$$

A simulation verification of proposed design was performed in MATLAB/SIMULINK environment. A typical control scheme used is depicted in Fig. 4.

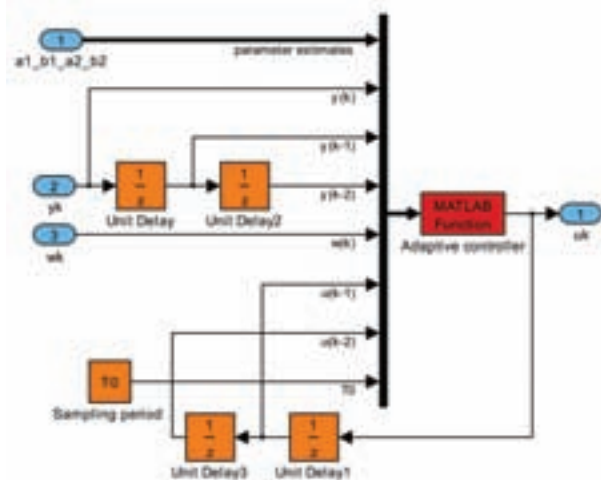


Fig. 5 Internal structure of the controller

This scheme is used for systems with time-delay of two sample steps. Individual blocks of the Simulink scheme

correspond to blocks of the general control scheme presented in Fig. 2. Blocks Compensator 1 and Compensator 2 are parts of the Smith Predictor and they correspond to $G_m(z^{-1})$ and $G_d(z^{-1})$ blocks of Fig. 2 respectively. The control algorithm is encapsulated in Main Pole Assignment Controller which corresponds to $G_c(z^{-1})$ Fig. 2 block. The Identification block performs the on-line identification of controlled system and outputs the estimates of 2nd order ARX model (a1, b1, a2, b2) parameters.

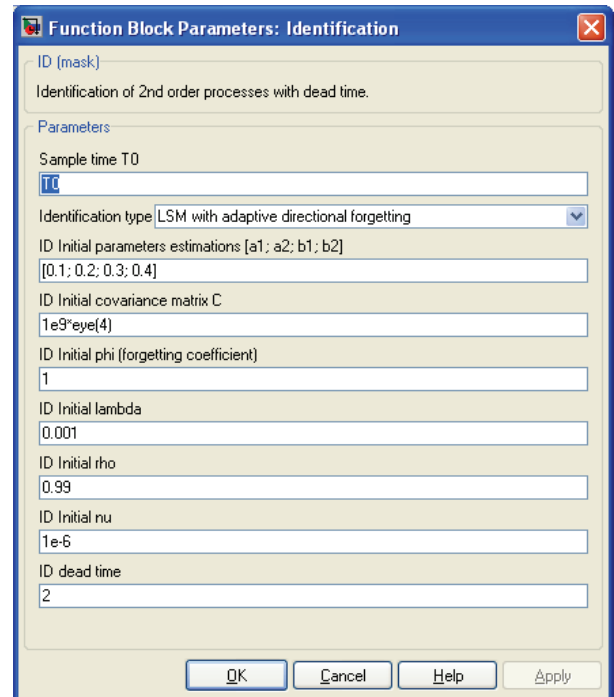


Fig. 6. Dialog for setting identification parameters

The on-line identification part of the scheme, which is represented by block Identification block in Fig. 3, uses several parameters that are entered via standard Simulink dialog. This dialog is presented in Fig. 6.

The most important parameters from the point of view of the problem this papers is coping with are sample time, initial parameters estimations and dead time. The dead time is not entered in time units but in sample times. The other parame-

ters affect the method used to compute ARX model and their detailed description can be found in [18], [19].

6. Simulation Results

The configuration for simulation verification of the designed algorithms was chosen as follows:

- All three control loops were verified in the non-adaptive versions without a random noise.
- A suitable time constant T_m was chosen for the control using the PIDSP controller and the pole assignment of the closed-loop was calculated. These poles were used for the design of the PASP controller.
- Both control loops were verified in the adaptive versions with a random noise. Firstly, without a priori information (the initial values of the model parameter estimates were chosen randomly). Secondly, using a priori information (the initial estimates were chosen on the basis of the previous experiments).
- The outputs of the process models were influenced by White Noise Generator with mean value $E = 0$ and covariance $R = 10^{-4}$.

6.1 Simulation verification of Adaptive Digital PIDSP

Figs. 7 - 10 illustrate the simulation control performance using PIDSP controller (14) - (16). From Figs. 7 and 8 (the control of the stable non-oscillatory model $G_1(z^{-1})$) is obvious that the control process is dependent on knowledge of a priori information. The process output y has a large overshoot, when the initial model parameter estimates are chosen randomly.

Using a priori information (the initial estimates were chosen on the basis of the previous experiments) leads to very good control quality (without overshoot of y and with short settling time).

Simulation results for the models $G_2(z^{-1})$ (the stable oscillatory model) and $G_3(z^{-1})$ (the non-minimum phase model) are shown in Figs. 9 and 10. The control quality (with a priori information) is very good. In case of the control of the non-minimum phase model $G_3(z^{-1})$ the course of the process output y has typical overshoots (undershoots) by change of the reference signal w .

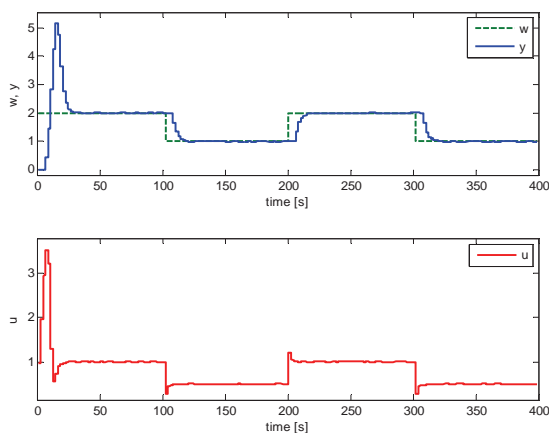


Fig. 7 Simulation results: control of the model $G_1(z^{-1})$, controller PIDSP (without a priori information)

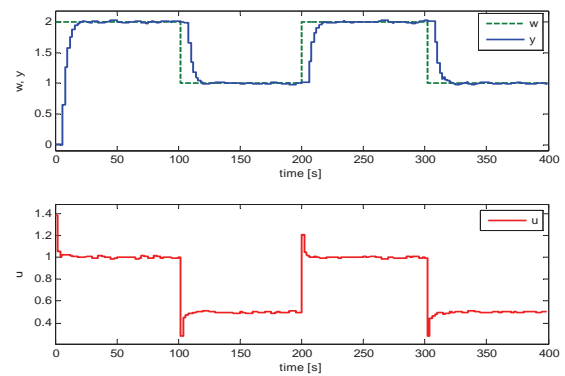


Fig. 8 Simulation results: control of the model $G_1(z^{-1})$, controller PIDSP (with a priori information)

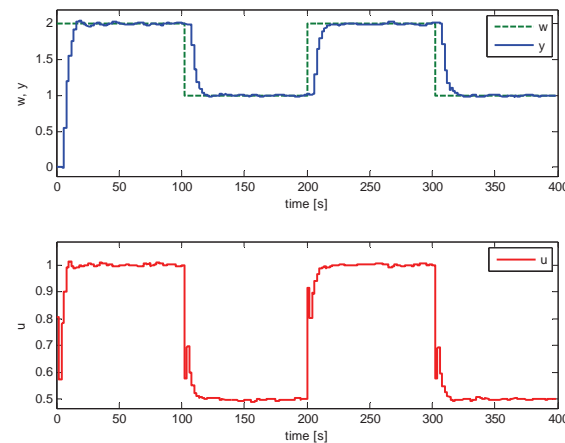


Fig. 9 Simulation results: control of the model $G_2(z^{-1})$, controller PIDSP (with a priori information)

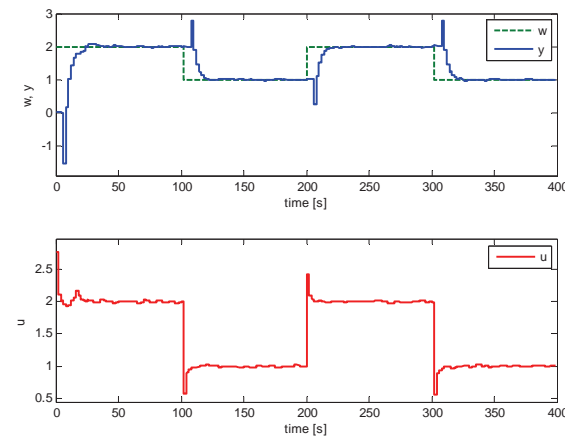


Fig. 10 Simulation results: control of the model $G_3(z^{-1})$, controller PIDSP (with a priori information)

6.2 Simulation verification of Adaptive Digital PASP

Figs. 11 - 13 illustrate the simulation control performance using PASP controller (14), (27). From Figs. 11 and 12 (the control of the stable model $G_1(z^{-1})$), it is obvious that the control process is not dependent on knowledge of a priori information (the control courses in both cases are practically identical). In the case of choosing of the suitable closed-loop poles, the adaptive PASP controller is more robust than the adaptive PIDSP controller.

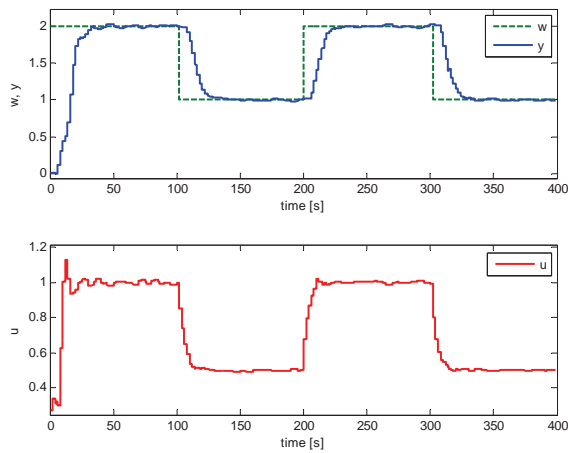


Fig. 11 Simulation results: control of model $G_1(z^{-1})$, controller PASP (without a priori information)

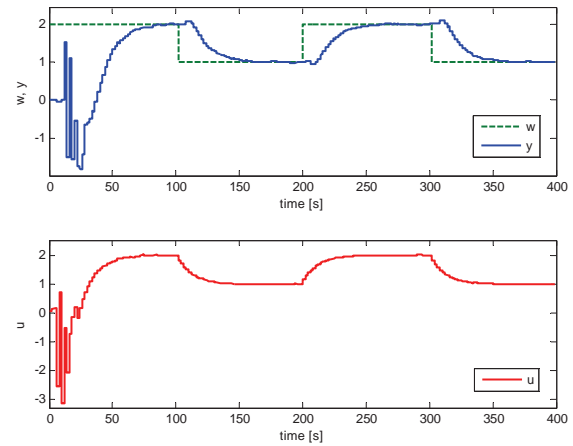


Fig. 14 Simulation results: control of the model $G_3(z^{-1})$, controller PASP (with a priori information)

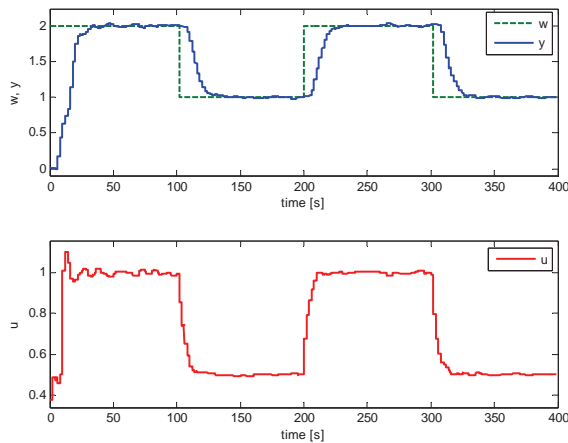


Fig. 12 Simulation results: control of the model $G_1(z^{-1})$, controller PASP (with a priori information)

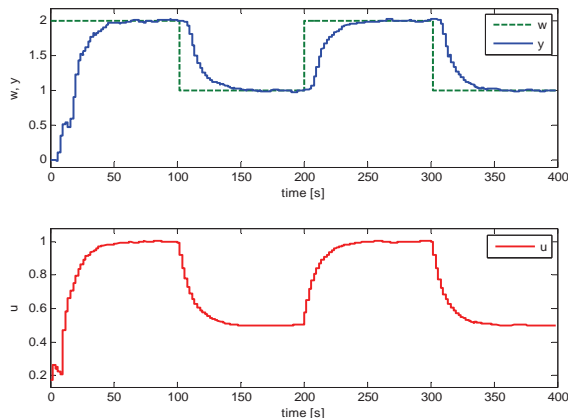


Fig. 13 Simulation results: control of the model $G_2(z^{-1})$, controller PASP (with a priori information)

Fig. 13 illustrates the simulation control performance of the stable oscillatory model $G_2(z^{-1})$. The control process is relatively slow without overshoot of y and u (it is the cautious adaptive controller).

Fig. 14 illustrates the simulation control performance of the non-minimum phase model $G_3(z^{-1})$. The control process is good after initial part.

Conclusion

Adaptive Smith Predictor algorithms for control of processes with time-delay based on polynomial design (pole assignment) was proposed. The polynomial controllers were derived purposely by analytical way (without utilization of numerical methods) to obtain algorithms with easy implementability in industrial practice. The pole assignment control algorithm was compared by simulation with adaptive digital Smith PID Predictor. Three models of control processes were used for simulation verification (the stable non-oscillatory, the stable oscillatory and the non-minimum phase). Results of simulation verification demonstrated advantages and disadvantages of individual approaches (see Figs. 7 – 14) for control of above mentioned processes with time-delay. The control quality depended on suitable start-up conditions of the recursive identification – important part plays a priori information. For the pole assignment method is very important suitable choice of the closed – loop poles. The designed adaptive SP algorithms will be verified in real time conditions for a control of the laboratory heat exchanger.

Acknowledgments

The authors wish to thank to the Ministry of Education of the Czech Republic (MSM7088352101) for financial support. This article was created with support of Operational Programme Research and Development for Innovations co-funded by the European Regional Development Fund (ERDF), national budget of the Czech Republic within the framework of the Centre of Polymer Systems project (reg. number: CZ.1.05/2.1.00/03.0111) and grant 1M0567.

References

- [1] NORMEY-RICO, J. E. and E. F. CAMACHO (2007). *Control of Dead-time Processes*. Springer-Verlag, London.
- [2] ÅSTRÖM, K. J. and T. HÄGGLUND (1995). *PID Controllers: Theory, Design and Tuning*. Instrument Society of America.
- [3] SMITH, O. J. (1957). Closed control of loops, *Chem. Eng. Progress*, **53**, 217-219.
- [4] VOGEL, E. F. and T. F. EDGAR (1980). A new dead time compensator for digital control. In: *Proceedings ISA Annual Conference*, Houston, USA, 29-46.

- [5] PALMOR, Z. J. and Y. HALEVI (1990). Robustness properties of sampled-data systems with dead time compensators. *Automatica*, **26**, 637-640.
- [6] NORMEY-RICO, J. E. and E. F. CAMACHO (1998). Dead-time compensators: A unified approach. In: *Proceedings of IFAC Workshop on Linear Time Delay Systems (LDTS'98)*, Grenoble, France, 141-146.
- [7] HANG C. C., LIM, K. W. and T. T. TAY (1986). On practical applications of adaptive control, In: *Proceedings Adaptive and Learning Systems-Theory and Applications*, London/New York, 105-108.
- [8] HANG, C. C., TONG, H. L. and K. H. WENG (1993). *Adaptive Control*. Instrument Society of America.
- [9] TORRICO, B. C. and J. E. NORMEY-RICO (2005). 2DOF discrete dead-time compensators for stable and integrative processes with dead-time. *Journal of Process Control*, **15**, 341-352.
- [10] ÅSTRÖM, K. J. and B. WITTENMARK (1984). *Computer Controlled Systems-Theory and Design*. Prentice-Hall, Englewood Cliffs, NJ.
- [11] LJUNGL, L. (1987). *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ.
- [12] HANG, C. C., LIM, K. W. and B. W. CHONG (1989). A dual-rate adaptive digital Smith predictor. *Automatica*, **20**, 1-16.
- [13] DAHLIN, D. B. (1968). Designing and tuning digital controllers. *Inst. Control Systems*, **42**, 77-73.
- [14] VIDYASAGAR, M. (1985). *Control System Synthesis: A Factorization Approach*. MIT Press, Cambridge M.A.
- [15] KUČERA, V. (1991). *Analysis and Design of Discrete Linear Control Systems*. Prentice-Hall, Englewood Cliffs, NJ.
- [16] KUČERA, V. (1993). Diophantine equations in control – a survey. *Automatica*, **29**, 1361-1375.
- [17] GUO, S. N., WANG, W. and L. S. SHIEH (2000). Discretisation of two degree-of-freedom controller and system with state, and output delays. *IEE Proceedings. Control Theory and Applications*, **147**, 87-96.
- [18] BOBÁL, V., BÖHM, J., FESSL, J. and J. MACHÁČEK (2005). *Digital Self-tuning Controllers: Algorithms, Implementation and Applications*. Springer-Verlag, London.
- [19] KULHAVÝ, R. (1987). Restricted exponential forgetting in real time identification. *Automatica*, **23**, 586-600.

prof. Ing. Vladimír Bobál, CSc.
Ing. Petr Chalupa, Ph.D.
prof. Ing. Petr Dostál, CSc.
Ing. Michal Brázdil

Tomas Bata University in Zlín
Faculty of Applied Informatics
Department of Process Control
Nad Stráněmi 4511
760 05 Zlín
Czech Republic
Tel.: +420 57603 5197
Fax: +420 57603 5279
e-mail: bobal@fai.utb.cz

Bounded Real Lemma Improved Forms

Anna Filasová and Dušan Krokavec

Abstract

The problem of the bounded real lemma equivalent representation for linear continuous-time systems is presented in the paper. The relationships between the terms of the system state equation using free weighting matrices is expressed, and improved LMI forms to bounded real lemma representation is derived. Possible extension to design a state feedback controller, which performs H_∞ properties of the closed-loop system, is formulated as a feasibility problem, and immediately expressed over a set of LMIs. Numerical example is included to illustrate the properties of the proposed representations.

Keywords: Continuous-time systems, linear matrix inequalities, bounded real lemma representation.

Introduction

Over the past decade, H_∞ theory seems to be one of the most sophisticated frameworks for robust control system design. Based on the concept of quadratic stability which attempts to find a quadratic Lyapunov function (LF), H_∞ norm computation problem is transferred into a standard linear matrix inequality (LMI) optimization task, which includes bounded real lemma (BRL) formulation [4], [9], [11]. A number of more or less conservative analysis methods are presented to assess robust stability for linear systems using a fixed Lyapunov function.

The first version of the BRL presents simple conditions under which a transfer function is contractive on the imaginary axis. Using this, it was possible to determine the H_∞ norm of the transfer function, and the BRL becomes a significant element to show and prove that the existence of feedback controllers (that results in a closed loop transfer matrix having the H_∞ norm less than a given upper bound), is equivalent to solutions existence of certain LMIs [1], [3].

Linear matrix inequality approach based on convex optimization algorithms is extensively applied to solve the above mentioned problem [5], [8] since it can be solved numerically efficiently by using developed interior-point algorithm. For time-varying parameters the quadratic stability approach is preferable utilized. Settin Lyapunov function be independent of uncertainties, this approach guaranties uniform asymptotic stability. To include this requirements equivalent LMI representations were introduced [10].

In this paper, enhanced LMI representations of BRL for linear continuous-time systems are introduced. Motivated by the underlying ideas in [2], [12] a simple technique for the BRL representation of linear systems is presented, and used modifications are explained in a context. The proposed LMI representations are proven to be necessary and sufficient, and their extensions to state feedback controller design, performing system H_∞ properties is immediate. Translating into LMI framework the closed-loop system stability is characterized in the terms of convex LMIs.

1. Problem Description

Through this paper the task is concerned with the computation of a state feedback control law $u(t)$, which control the linear dynamic system given by the set of equations

$$\dot{q}(t) = Aq(t) + Bu(t) \quad (1)$$

$$y(t) = Cq(t) + Du(t) \quad (2)$$

where $q(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^r$, and $y(t) \in \mathbb{R}^m$ are vectors of the state, input and measurable output variables, respectively, nominal system matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$, $C \in \mathbb{R}^{m \times n}$ and $D \in \mathbb{R}^{m \times r}$ are real matrices.

Problem of the interest is to design asymptotically stable closed-loop system with the linear memory-less state feedback controller of the form

$$u(t) = -Kq(t) \quad (3)$$

where matrix $K \in \mathbb{R}^{r \times n}$ is a gain matrix.

2. Basic Preliminaries

Proposition 1. (Bounded real lemma) System (1), (2) is asymptotically stable if there exist a symmetric positive definite matrix $P > 0$ and a positive scalar $\gamma > 0$ such that

$$\begin{bmatrix} A^T P + PA & PB & C^T \\ * & -\gamma^2 I_r & D^T \\ * & * & -I_m \end{bmatrix} < 0 \quad (4)$$

where $I_r \in \mathbb{R}^{r \times r}$, $I_m \in \mathbb{R}^{m \times m}$, are identity matrices, respectively.

Hereafter, * denotes the symmetric item in a symmetric matrix.

Proof. (see. e.g. [6]) Defining Lyapunov function as follows

$$v(q(t)) = q^T(t)Pq(t) + \int_0^t (y^T(r)y(r) - \gamma^2 u^T(r)u(r))dr > 0 \quad (5)$$

where $P = P^T > 0$, $P \in \mathbb{R}^{n \times n}$, $\gamma > 0$, and evaluating derivative of $v(q(t))$ with respect to t along a system trajectory then it yields

$$\dot{v}(q(t)) = \dot{q}^T(t)Pq(t) + q^T(t)P\dot{q}(t) + y^T(t)y(t) - \gamma^2 u^T(t)u(t) < 0 \quad (6)$$

Thus, substituting (1), (2) into (6) gives

$$\begin{aligned} \dot{v}(q(t)) &= (Aq(t) + Bu(t))^T Pq(t) + \\ &+ q^T(t)P(Aq(t) + Bu(t)) - \gamma u^T(t)u(t) + \\ &+ (Cq(t) + Du(t))^T (Cq(t) + Du(t)) < 0 \end{aligned} \quad (7)$$

and with the next notation

$$q_c^T(t) = [q^T(t) \quad u^T(t)] \quad (8)$$

it is obtained

$$\dot{v}(q(t)) = q_c^T(t)P_c q_c(t) < 0 \quad (9)$$

where

$$P_c = \begin{bmatrix} A^T P + PA & PB \\ * & -\gamma^2 I_r \end{bmatrix} + \begin{bmatrix} C^T C & C^T D \\ * & D^T D \end{bmatrix} < 0 \quad (10)$$

Since

$$\begin{bmatrix} C^T C & C^T D \\ * & D^T D \end{bmatrix} = \begin{bmatrix} C^T \\ D^T \end{bmatrix} \begin{bmatrix} C & D \end{bmatrix} \geq 0 \quad (11)$$

Schur complement property implies

$$\begin{bmatrix} 0 & 0 & C^T \\ * & 0 & D^T \\ * & * & -I_m \end{bmatrix} \geq 0 \quad (12)$$

and using (12) the LMI condition (10) can be written compactly as (4). This concludes the proof. ■

3. Improved BRL Representation

Theorem 1. System (1), (2) is asymptotically stable if there exist a symmetric positive definite matrix $P > 0$, $P \in \mathbb{R}^{n \times n}$, matrices $S_1, S_2 \in \mathbb{R}^{n \times n}$, and a positive scalar $\gamma > 0$, $\gamma \in \mathbb{R}$ such that

$$\begin{bmatrix} -S_1 A - A^T S_1^T & -S_1 B & P + S_1 - A^T S_2^T & C^T \\ * & -\gamma^2 I_r & -B^T S_2^T & D^T \\ * & * & S_2 + S_2^T & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (13)$$

Proof. Since (1) implies

$$\dot{q}(t) - Aq(t) - Bu(t) = 0 \quad (14)$$

then with arbitrary square matrices $S_1, S_2 \in \mathbb{R}^{n \times n}$ it yield

$$(q^T(t)S_1 + \dot{q}^T(t)S_2)(\dot{q}(t) - Aq(t) - Bu(t)) = 0 \quad (15)$$

Thus, adding (15), as well as its transposition to (6) and substituting (2) it can be written

$$\begin{aligned} \dot{v}(q(t)) &= -\gamma u^T(t)u(t) + \dot{q}^T(t)Pq(t) + q^T(t)P\dot{q}(t) + \\ &+ (Cq(t) + Du(t))^T (Cq(t) + Du(t)) + \\ &+ (\dot{q}(t) - Aq(t) - Bu(t))^T (S_1^T q(t) + S_2^T \dot{q}(t)) + \\ &+ (q^T(t)S_1 + \dot{q}^T(t)S_2)(\dot{q}(t) - Aq(t) - Bu(t)) < 0 \end{aligned} \quad (16)$$

and using the notation

$$q_c^T(t) = [q^T(t) \quad u^T(t) \quad \dot{q}^T(t)] \quad (17)$$

it can be obtained

$$\dot{v}(q(t)) = q_c^T(t)P_c^\circ q_c(t) < 0 \quad (18)$$

where

$$\begin{aligned} P_c^\circ &= \begin{bmatrix} C^T C & C^T D & 0 \\ * & D^T D & 0 \\ * & * & 0 \end{bmatrix} + \\ &+ \begin{bmatrix} -S_1 A - A^T S_1^T & -S_1 B & P + S_1 - A^T S_2^T \\ * & -\gamma^2 I_r & -B^T S_2^T \\ * & * & S_2 + S_2^T \end{bmatrix} < 0 \end{aligned} \quad (19)$$

Thus, analogously using (11), (12) the inequality (19) can be written compactly as (13). This concludes the proof. ■

Remark 1. Setting $S_1 = P$ then (13) is transformed in

$$\begin{bmatrix} PA + A^T P & PB & -A^T S_2^T & C^T \\ * & -\gamma^2 I_r & -B^T S_2^T & D^T \\ * & * & S_2 + S_2^T & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (20)$$

Thus, inserting $S_2 = -\delta I$, $\delta > 0$, $\delta \in \mathbb{R}$ where gives

$$\begin{bmatrix} PA + A^T P & PB & \delta A^T & C^T \\ * & -\gamma I_r & \delta B^T & D^T \\ * & * & -2\delta I_n & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (21)$$

$$\begin{bmatrix} PA + A^T P & PB & A^T & C^T \\ * & -\gamma I_r & B^T & D^T \\ * & * & -2\delta^{-1} I_n & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (22)$$

respectively. Then (22) can be written as

$$\begin{bmatrix} PA + A^T P & PB & C^T \\ * & -\gamma I_r & D^T \\ * & * & -I_m \end{bmatrix} + 0.5\delta \begin{bmatrix} A^T \\ B^T \\ 0 \end{bmatrix} \begin{bmatrix} A & B & 0 \end{bmatrix} < 0 \quad (23)$$

Choosing δ as a sufficiently small positive scalar satisfying the condition

$$0 < \delta < 2 \frac{\lambda_1}{\lambda_2} \quad (24)$$

$$\lambda_1 = \lambda_{\min} \left\{ \begin{bmatrix} PA + A^T P & PB & C^T \\ * & -\gamma^2 I_r & D^T \\ * & * & -I_m \end{bmatrix} \right\} \quad (25)$$

$$\lambda_2 = \lambda_{\max} \left\{ \begin{bmatrix} A^T A & A^T B & 0 \\ B^T A & B^T B & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\} \quad (26)$$

(21) be negative definite for a feasible P of (4).

Corollary 1. Setting $S_1 = -P$, and $S_2 = \delta I$, where $0 < \delta \in \mathbb{R}$ then (20)-(22) implies

$$\begin{bmatrix} A^T P + PA & PB & C^T \\ * & -\gamma^2 I_r & D^T \\ * & * & -I_m \end{bmatrix} - 0.5\delta \begin{bmatrix} -A^T \\ -B^T \\ 0 \end{bmatrix} \begin{bmatrix} -A & -B & 0 \end{bmatrix} < 0 \quad (27)$$

and a feasible solution P of (4) is also a feasible solution of (27) for all $\delta > 0, \delta \in \mathbb{R}$.

Theorem 2. System (1), (2) is asymptotically stable if there exist a symmetric positive definite matrix $P > 0$, $P \in \mathbb{R}^{n \times n}$, matrices $S_1, S_2 \in \mathbb{R}^{n \times n}$, and a positive scalar $\gamma > 0, \gamma \in \mathbb{R}$ such that

$$\begin{bmatrix} PA + A^T P & PB & P + S_1 - A^T S_2 & C^T \\ * & -\gamma^2 I_r & B^T S_2 & D^T \\ * & * & S_2 + S_2^T & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (28)$$

Proof. Defining the congruence transform matrix

$$T_1 = \begin{bmatrix} I & & & \\ & I & & \\ A & B & I & \\ & & & I \end{bmatrix} \quad (29)$$

and multiplying right-hand side of (13) by T_1 and left-hand side of (13) by T_1^T then after tedious calculation (28) is obtained. This concludes the proof. ■

Remark 2. Using $S_1 = -P$, $S_2 = -\delta P$ then (28) leads to

$$\begin{bmatrix} PA + A^T P & PB & -\delta A^T P & C^T \\ * & -\gamma^2 I_r & -\delta B^T P & D^T \\ * & * & -2\delta P & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (30)$$

$$\begin{bmatrix} PA + A^T P & PB & -A^T P & C^T \\ * & -\gamma^2 I_r & -B^T P & D^T \\ * & * & -2\delta^{-1} P & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (31)$$

respectively, and using Schur complement property then (31) can be rewritten as

$$\begin{bmatrix} PA + A^T P & PB & C^T \\ * & -\gamma^2 I_r & D^T \\ * & * & -I_m \end{bmatrix} + \frac{\delta}{2} \begin{bmatrix} -A^T P \\ -B^T P \\ 0 \end{bmatrix} P^{-1} \begin{bmatrix} -PA & -PB & 0 \end{bmatrix} < 0 \quad (32)$$

$$\begin{bmatrix} PA + A^T P & PB & C^T \\ * & -\gamma^2 I_r & D^T \\ * & * & -I_m \end{bmatrix} + \frac{\delta}{2} \begin{bmatrix} A^T PA & A^T PB & 0 \\ B^T PA & B^T PB & 0 \\ 0 & 0 & 0 \end{bmatrix} < 0 \quad (33)$$

respectively.

Choosing δ satisfying (24), then with (25) and with

$$\lambda_2 = \lambda_{\max} \left\{ \begin{bmatrix} A^T PA & A^T PB & 0 \\ B^T PA & B^T PB & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\} \quad (34)$$

(31) be negative definite for a feasible P of (4). This concludes the proof. ■

Corollary 2. Considering (32), (33) it is evident that the inequality

$$\begin{bmatrix} PA + A^T P & PB & A^T P & C^T \\ * & -\gamma^2 I_r & B^T P & D^T \\ * & * & -2\delta^{-1} P & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (35)$$

and (31) are equivalent.

4. Control Law Parameter Design

Theorem 3. Closed-loop system (1), (2), (3) is stable if there exists a symmetric positive definite matrix $X > 0$, $X \in \mathbb{R}^{n \times n}$, a regular square matrix $Z \in \mathbb{R}^{n \times n}$, a matrix $Y \in \mathbb{R}^{r \times n}$, and a scalar $\delta > 0, \delta \in \mathbb{R}$ such that

$$X = X^T > 0, \quad \gamma > 0 \quad (36)$$

$$\begin{bmatrix} \Pi_{11} & B & XA^T - Y^T B^T & XC^T - Y^T D^T \\ * & -\gamma^2 I_r & B^T & D^T \\ * & * & Z + Z^T & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (37)$$

$$\Pi_{11} = AX + XA^T - BY - Y^T B^T \quad (38)$$

The control law gain matrix is given as

$$K = YX^{-1} \quad (39)$$

Proof. Setting $S_1 = -P$ then (28) implies

$$\begin{bmatrix} PA + A^T P & PB & A^T S_2^T & C^T \\ * & -\gamma^2 I_r & B^T S_2^T & D^T \\ * & * & S_2 + S_2^T & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (40)$$

Supposing that $\det(S_2) \neq 0$ then it can be defined the congruence transform matrix

$$T_2 = \text{diag} \left[P^{-1} \quad I_r \quad S_2^{-1} \quad I_m \right] \quad (41)$$

and pre-multiplying right-hand side of (40) by T_2 , and left-hand side of (40) by T_2^T leads to

$$\begin{bmatrix} AP^{-1} + P^{-1}A^T & B & P^{-1}A^T & P^{-1}C^T \\ * & -\gamma^2 I_r & B^T & D^T \\ * & * & S_2^{-1} + S_2^{-T} & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \quad (42)$$

Thus, denoting

$$P^{-1} = X, \quad S_2^{-1} = Z \quad (43)$$

(42) can be written as

$$\begin{bmatrix} \mathbf{AX} + \mathbf{XA}^T & \mathbf{B} & \mathbf{XA}^T & \mathbf{XC}^T \\ * & -\gamma^2 \mathbf{I}_r & \mathbf{B}^T & \mathbf{D}^T \\ * & * & \mathbf{Z} + \mathbf{Z}^T & \mathbf{0} \\ * & * & * & -\mathbf{I}_m \end{bmatrix} < 0 \quad (44)$$

Inserting $\mathbf{A} \leftarrow \mathbf{A}_c = \mathbf{A} - \mathbf{BK}$, and $\mathbf{C} \leftarrow \mathbf{C}_c = \mathbf{C} - \mathbf{DK}$ it yields

$$\begin{bmatrix} \mathbf{\Pi}_{11} & \mathbf{B} & \mathbf{X}(\mathbf{A}^T - \mathbf{K}^T \mathbf{B}^T) & \mathbf{X}(\mathbf{C}^T - \mathbf{K}^T \mathbf{D}^T) \\ * & -\gamma^2 \mathbf{I}_r & \mathbf{B}^T & \mathbf{D}^T \\ * & * & \mathbf{Z} + \mathbf{Z}^T & \mathbf{0} \\ * & * & * & -\mathbf{I}_m \end{bmatrix} < 0 \quad (45)$$

$$\mathbf{\Pi}_{11} = \mathbf{AX} + \mathbf{XA}^T - \mathbf{BKX} - \mathbf{XK}^T \mathbf{B}^T \quad (46)$$

and with

$$\mathbf{Y} = \mathbf{KX} \quad (47)$$

(45), (46) implies (37), (38), respectively. This concludes the proof. ■

Remark 3. Setting $\mathbf{Z} = -\delta \mathbf{X}$ then with $\mathbf{D} = \mathbf{0}$ the control law design condition (36)-(38) can be rewritten as

$$\mathbf{X} = \mathbf{X}^T > 0, \quad \gamma > 0 \quad (48)$$

$$\begin{bmatrix} \mathbf{\Pi}_{11} & \mathbf{B} & \mathbf{XA}^T - \mathbf{Y}^T \mathbf{B}^T & \mathbf{XC}^T \\ * & -\gamma^2 \mathbf{I}_r & \mathbf{B}^T & \mathbf{0} \\ * & * & -2\delta \mathbf{I}_n & \mathbf{0} \\ * & * & * & -\mathbf{I}_m \end{bmatrix} < 0 \quad (49)$$

$$\mathbf{\Pi}_{11} = \mathbf{AX} + \mathbf{XA}^T - \mathbf{BY} - \mathbf{Y}^T \mathbf{B}^T \quad (50)$$

where feasible $\mathbf{X}, \mathbf{Y}, \delta$ implies gain matrix parameter (39).

Therefore, it is evident that the design standard form of BRL is

$$\begin{bmatrix} \mathbf{\Pi}_{11} & \mathbf{B} & \mathbf{X}(\mathbf{C}^T - \mathbf{K}^T \mathbf{D}^T) \\ * & -\gamma^2 \mathbf{I}_r & \mathbf{0} \\ * & * & -\mathbf{I}_m \end{bmatrix} < 0 \quad (51)$$

Note, other nontrivial solutions can be obtained using different setting of S_l , $l=1,2$.

5. Illustrative Example

The approaches given above are illustrated by the numerical example where the parameters of (1), (2) are

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -9 & -5 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 1 & 5 \end{bmatrix}, \quad \mathbf{C}^T = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix}$$

Solving (48), (49) with respect to LMI matrix variables $\mathbf{X}, \mathbf{Y}, \gamma$ and δ using SeDuMi (Self-Dual-Minimization) package for Matlab [7], given task was feasible with

$$\mathbf{X} = \begin{bmatrix} 3.7160 & -2.6784 & 1.2147 \\ -2.6784 & 3.0184 & -1.8970 \\ 1.2147 & -1.8970 & 3.2896 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 0.8937 & 2.1673 & -1.4078 \\ -0.0801 & -0.0207 & 0.5383 \end{bmatrix}$$

$$\gamma = 11.0242, \quad \delta = 6.7040$$

and results the control system parameters

$$\mathbf{K} = \begin{bmatrix} 2.2731 & 3.0405 & 0.4860 \\ 0.0152 & 0.1662 & 0.2538 \end{bmatrix}$$

$$\rho(\mathbf{A}_c) = \{-0.9398 \quad -3.1252 \quad -11.2561\}$$

It is evident, that the eigenvalues spectrum $\rho(\mathbf{A}_c)$ of the closed control loop is stable.

Solving (48), (51) with respect to LMI matrix variables \mathbf{X}, \mathbf{Y} and γ , given task was feasible, too. Obtained LMI variables were

$$\mathbf{X} = \begin{bmatrix} 2.7636 & -1.7983 & 0.6386 \\ -1.7983 & 2.2479 & -1.2081 \\ 0.6386 & -1.2081 & 3.0925 \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} 0.9127 & 1.6581 & -0.8163 \\ 0.2802 & 0.1304 & -0.2269 \end{bmatrix}$$

$$\gamma = 6.9412$$

and implies

$$\mathbf{K} = \begin{bmatrix} 1.7557 & 2.2852 & 0.2662 \\ 0.2837 & 0.2709 & -0.0261 \end{bmatrix}$$

$$\rho(\mathbf{A}_c) = \{-0.8968 \quad -5.8435 \pm 1.7282i\}$$

It is evident, that performance γ is less than one obtained with respect to (49) but this fetches worst dynamic properties.

It also should be noted, the cost value γ will not be a monotonously decreasing function with the decreasing of δ , if δ is chosen.

Concluding Remarks

This paper describes a simple technique for enhanced BRL representation and its application to the H_∞ control of linear continuous-time systems. Standard criterion is extended for a system with constant coefficient matrices by employing free weighting matrices to take the relationship between the terms of the system equation into account in the structure of BRL. The method is further extended to the design of an H_∞ state-feedback memory-free control policy. Numerical example demonstrates that principles described in the paper are effective, although some computational complexity is increases.

The advantage of this approach is that there in Theorem 1 Lyapunov matrix \mathbf{P} is separated from $\mathbf{A}, \mathbf{B}^T, \mathbf{C}$, and \mathbf{D}^T , i.e. there are no terms containing the product of \mathbf{P} and any of them. This enables a new structure of BRL to be derived for a system with polytopic uncertainties by using a parameter-dependent Lyapunov function, and to deal with linear systems with parametric uncertainties. It seems to be a useful extension to other control performance synthesis problems, too.

Acknowledgments

The work presented in this paper was supported by VEGA, Grant Agency of Ministry of Education and Academy of Science of Slovak Republic under Grant No. 1/0256/11. This support is very gratefully acknowledged.

References

- [1] BOYD, D., EL GHAOU, L., PERON, E. and BALAKRISHNAN, V.: *Linear Matrix Inequalities in System and Control Theory*, SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1994.
- [2] FILASOVÁ, A. and KROKAVEC, D.: Global asymptotically stable control design for time-delay systems, *AT&P Journal Plus*, 2, 89-92, 2009.
- [3] FILASOVÁ, A., GONTKOVIČ, D. and KROKAVEC, D.: Output feedback control design using unified algebraic approach. In *Proceedings of the 8th International Symposium on Applied Machine Intelligence and Informatics SAMI 2010*, Herľany, Slovakia, 259-262, 2010.
- [4] HERRMANN, G., TURNER, M.C. and POSTLETHWAITE, I.: Linear matrix inequalities in control. In *Mathematical Methods for Robust and Nonlinear Control*, Springer-Verlag, Berlin, 123-142, 2007.
- [5] JIA, Y.: Alternative proofs for improved LMI representations for the analysis and the design of continuous-time systems with polytopic type uncertainty: A predictive approach, *IEEE Transactions on Automatic Control*, 48:8, 1413-1416, 2003.
- [6] KROKAVEC, D. and FILASOVÁ, A.: *Discrete-Time Systems*. Elfa, Košice, 2008. (in Slovak)
- [7] PEAUCELLE, D., HENRION, D., LABIT, Y. and TAITZ, K.: *User's Guide for SeDuMi Interface 1.04*, LAAS-CNRS, Toulouse, 2002.
- [8] PIPELEERS, G., DEMEULENAEREA, B., SWEVERSA, J. and VANDENBERGHEB, L.: Extended LMI characterizations for stability and performance of linear systems, *Systems & Control Letters*, 58, 510-518, 2009.
- [9] VESELÝ, V. and ROSINOVÁ, D.: Robust output model predictive control design: BMI approach, *International Journal of Innovative Computing, Information and Control*, 5:4, 1115-1123, 2009.
- [10] WU, A.I. and DUAN, G.R.: Enhanced LMI representations for H_2 performance of polytopic uncertain systems: Continuous-time case, *International Journal of Automation and Computing*, 3, 304-308, 2006.
- [11] WU, M., HE, Y. and SHE, J.H.: *Stability Analysis and Robust Control of Time-Delay Systems*, Science Press, Beijing and Springer-Verlag, Berlin, 2010.
- [12] XIE, W.: An equivalent LMI representation of bounded real lemma for continuous-time systems, *Journal of Inequalities and Applications*, 5, 2010.

Abstrakt

V príspevku je uvedený prístup k vytváraniu ekvivalentnej reprezentácie vety o reálnom ohraničení pre lineárne spojité dynamické systémy. Základné relácie medzi prvkami stavového opisu, vnorené do riešenia pomocou voliteľných matic, sú využité na odvodenie vhodnejšieho tvaru vety o reálnom ohraničení v štruktúre lineárnych maticových nerovností. Potenciálne rozšírenie pre návrh stavového riadenia, ktoré zaručuje požadované frekvenčné vlastnosti uzavretého obvodu v zmysle ohraničenia H_∞ normou, je formulované ako optimalizačný problém. Použitý príklad ilustruje princíp návrhu a vlastnosti navrhovanej reprezentácie danej vety.

doc. Ing. Anna Filasová, CSc.
prof. Ing. Dušan Krokavec, CSc.

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
Fax: ++421 55 625 3574
E-mail: anna.filasova@tuke.sk; dusan.krokavec@tuke.sk

Eigenstructure Decoupling in State Feedback Control Design

Pavol Kocsis and Róbert Fónod

Abstract

The presented design method aim is to synthesize a state feedback control law in such way that with respect to the prescribed eigenvalues of the closed-loop system matrix the corresponding eigenvectors are as close as possible to a decoupled system eigenvectors. It is demonstrated that some degree of freedom existing in the control design, representing by the parametric vectors set as well as by the set of closed-loop eigenvalues, can be properly used to meet some desired specification requirement. An illustrative example and the simulation results show that the proposed design principle is effective and simple.

Keywords: State feedback, eigenstructure assignment, mode decoupling, singular value decomposition, orthogonal complement.

Introduction

The static and the dynamic pole placement belongs to the prominent design problems of modern control theory, and, although its practical usefulness has been continuously in dispute, it is one of the most intensively investigated in control system design. It seems that the state-feedback pole assignment in control system design is one from the preferred techniques. In the single-input case the solution to this problem, when it exists, is unique. In the multi-input multi output (MIMO) case various solutions may exist [6], [10], and to obtain a specific solution some additional conditions have to be supplied in order to eliminate the extra degrees of freedom in design strategy.

In the last decade significant progress has been achieved in this field, coming in its formulation closest to the algebraic geometric nature of the pole placement problem [11], [17]. The reason for the discrepancy in opinions about the conditioning of the pole assignment problem is that one has to distinguish among three aspects of the pole placement problem, the computation of the memory-less feedback control law matrix gain, the computation of the closed loop system matrix eigenvalues spectrum and the suppressing of the cross-coupling effect [16], where one manipulated input variable cause change in more outputs variables.

Thus, eigenstructure assignment seems to be a powerful technique concerned with the placing of eigenvalues and their associated eigenvectors via feedback control laws, to meet closed-loop design specifications. The eigenvalues are the principal factors that govern the stability and the rates of decay or rise of the system dynamic response. The right and left eigenvectors, on the other hand, are dual factors that together determine the relative shape of the system dynamic response [9], [12], [15].

The general problem of assigning the system matrix eigenstructure using the state feedback control is considered in this paper. Based on the classic algebraic methods [3], [4], [14], as well as on the algorithms for pole assignment using Singular Value Decomposition (SVD) [5], [13] the exposition of the pole eigenstructure assignment problem is generalized here to handle the specified structure of the left eigenvector set in state feedback control design for MIMO linear

systems. Extra freedom, which makes dependent the closed-loop eigenvalues spectrum, is used for closed-loop state variables mode decoupling.

The integrated procedure provides a straightforward methodology usable in linear control system design techniques when the memory-free controller in the state-space control structures takes the standard form. Presented application for closed-loop state variables mode decoupling is relative simple and its worth can help to disclose the continuity between eigenstructure assignment and system variable dominant dynamic specification.

1. Problem Statement

Linear dynamic system with n degree of freedom can be modelled by the state-space equations

$$\dot{\mathbf{q}}(t) = \mathbf{A}\mathbf{q}(t) + \mathbf{B}\mathbf{u}(t) \quad (1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{q}(t) \quad (2)$$

where $\mathbf{q}(t) \in \mathbf{R}^n$, $\mathbf{u}(t) \in \mathbf{R}^r$ and $\mathbf{y}(t) \in \mathbf{R}^m$ are vectors of the state, input and measurable output variables, respectively, and the system matrix parameters $\mathbf{A} \in \mathbf{R}^{n \times n}$, $\mathbf{B} \in \mathbf{R}^{n \times r}$, and $\mathbf{C} \in \mathbf{R}^{m \times n}$ are constant and finite valued.

Generally, to the controllable time-invariant linear MIMO system (1) a linear state-feedback controller can be defined by a control policy

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{q}(t) + \mathbf{L}\mathbf{w}(t) \quad (3)$$

$\mathbf{K} \in \mathbf{R}^{r \times n}$, $\mathbf{L} \in \mathbf{R}^{r \times m}$ to give rise to the closed-loop system description

$$\dot{\mathbf{q}}(t) = \mathbf{A}_c \mathbf{q}(t) + \mathbf{B}\mathbf{L}\mathbf{w}(t) \quad (4)$$

in such way that roots of the closed loop characteristic polynomial are eigenvalues of $\mathbf{A}_c = (\mathbf{A} - \mathbf{B}\mathbf{K})$, $\mathbf{A}_c \in \mathbf{R}^{n \times n}$.

Throughout the paper it is assumed that the pair (\mathbf{A}, \mathbf{B}) is controllable.

2. Basic Preliminaries

2.1 Orthogonal Complement

Definition 1. (Null space) Let $\mathbf{E}, \mathbf{E} \in \mathbf{R}^{h \times h}$, $\text{rank}(\mathbf{E}) = k < h$ be a rank deficient matrix. Then the null space $\mathcal{N}_{\mathbf{E}}$ of \mathbf{E} is the orthogonal complement of the row space of \mathbf{E} .

Proposition 1. Let $\mathbf{E}, \mathbf{E} \in \mathbf{R}^{h \times h}$, $\text{rank}(\mathbf{E}) = k < h$ be a rank deficient matrix. Then an orthogonal complement \mathbf{E}^\perp of \mathbf{E} is

$$\mathbf{E}^\perp = \mathbf{D}\mathbf{U}_2^T \quad (5)$$

where \mathbf{U}_2^T is the null space of \mathbf{E} and \mathbf{D} is an arbitrary matrix of appropriate dimension.

Proof. (see e.g. [8]) The SVD of $\mathbf{E}, \mathbf{E} \in \mathbf{R}^{h \times h}$, $\text{rank}(\mathbf{E}) = k < h$ gives

$$\mathbf{U}^T \mathbf{E} \mathbf{V} = \begin{bmatrix} \mathbf{U}_1^T \\ \mathbf{U}_2^T \end{bmatrix} \mathbf{E} \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} = \begin{bmatrix} \Sigma_1 & \mathbf{0}_{12} \\ \mathbf{0}_{21} & \mathbf{0}_{22} \end{bmatrix} \quad (6)$$

where $\mathbf{U} \in \mathbf{R}^{h \times h}$ is the orthogonal matrix of the left singular vectors, $\mathbf{V} \in \mathbf{R}^{h \times h}$ is the orthogonal matrix of the right singular vectors of \mathbf{E} and $\Sigma_1 \in \mathbf{R}^{k \times k}$ is the diagonal positive definite matrix

$$\Sigma_1 = \text{diag}[\sigma_1 \cdots \sigma_k], \quad \sigma_1 \geq \cdots \geq \sigma_k > 0 \quad (7)$$

which diagonal elements are the singular values of \mathbf{E} . Using orthogonal properties of \mathbf{U} and \mathbf{V} , i.e. $\mathbf{U}^T \mathbf{U} = \mathbf{I}_h$, $\mathbf{V}^T \mathbf{V} = \mathbf{I}_h$, $\mathbf{U}_2^T \mathbf{U}_1 = \mathbf{0}$, then

$$\mathbf{E} = \mathbf{U} \Sigma \mathbf{V}^T = \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0}_{12} \\ \mathbf{0}_{21} & \mathbf{0}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix} = \quad (8)$$

$$= \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{0}_2 \end{bmatrix} = \mathbf{U}_1 \mathbf{S}_1$$

where $\mathbf{S}_1 = \Sigma_1 \mathbf{V}_1^T$. Thus (8) implies

$$\mathbf{U}_2^T \mathbf{E} = \mathbf{U}_2^T \begin{bmatrix} \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{0}_2 \end{bmatrix} = \mathbf{0} \quad (9)$$

It is evident that for an arbitrary matrix \mathbf{D} is

$$\mathbf{D} \mathbf{U}_2^T \mathbf{E} = \mathbf{E}^\perp \mathbf{E} = \mathbf{0} \quad (10)$$

respectively, which implies (5). ■

2.2 System Model Canonical Form

Proposition 2. If $\text{rank}(\mathbf{CB}) = m$ then there exists a coordinates change in which $(\mathbf{A}^\circ, \mathbf{B}^\circ, \mathbf{C}^\circ)$ takes the structure

$$\mathbf{A}^\circ = \begin{bmatrix} \mathbf{A}_{11}^\circ & \mathbf{A}_{12}^\circ \\ \mathbf{A}_{21}^\circ & \mathbf{A}_{22}^\circ \end{bmatrix}, \quad \mathbf{B}^\circ = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_2^\circ \end{bmatrix}, \quad \mathbf{C}^\circ = \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \end{bmatrix} \quad (11)$$

where $\mathbf{A}_{11}^\circ \in \mathbf{R}^{(n-m) \times (n-m)}$, $\mathbf{B}_2^\circ \in \mathbf{R}^{m \times m}$ is a non-singular matrix, and $\mathbf{I}_m \in \mathbf{R}^{m \times m}$ is identity matrix.

Proof. (see e.g. [7]) Considering the state-space description of the system (1), (2) with $r = m$ and defining the transform matrix \mathbf{T}_1^{-1} such that

$$\mathbf{C}_1 = \mathbf{C} \mathbf{T}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \end{bmatrix}, \quad \mathbf{T}_1^{-1} = \begin{bmatrix} \mathbf{I}_{n-m} & \mathbf{0} \\ & \mathbf{C} \end{bmatrix} \quad (12)$$

then

$$\mathbf{B}_1 = \mathbf{T}_1^{-1} \mathbf{B} = \mathbf{T}_1^{-1} \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{CB} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{11} \\ \mathbf{B}_{12} \end{bmatrix} \quad (13)$$

If $\mathbf{CB} = \mathbf{B}_{12}$ is a regular matrix (in opposite case the pseudo inverse of \mathbf{B}_{12} is possible to use), then the second transform matrix \mathbf{T}_2^{-1} can be defined as follows

$$\mathbf{T}_2^{-1} = \begin{bmatrix} \mathbf{I}_{n-m} & -\mathbf{B}_{11} \mathbf{B}_{12}^{-1} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix} \quad (14)$$

$$\mathbf{T}_2 = \begin{bmatrix} \mathbf{I}_{n-m} & \mathbf{B}_{11} \mathbf{B}_{12}^{-1} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix} \quad (15)$$

This application results in

$$\mathbf{B}^\circ = \mathbf{T}_2^{-1} \mathbf{B}_1 = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_2^\circ \end{bmatrix} = \mathbf{T}_2^{-1} \mathbf{T}_1^{-1} \mathbf{B} = \mathbf{T}_c^{-1} \mathbf{B} \quad (16)$$

where

$$\mathbf{B}_{11} = \mathbf{B}_1, \quad \mathbf{B}_2^\circ = \mathbf{B}_{12} = \mathbf{CB}, \quad \mathbf{T}_c^{-1} = \mathbf{T}_1^{-1} \mathbf{T}_2^{-1} \quad (17)$$

and analogously

$$\mathbf{C}^\circ = \mathbf{C}_1 \mathbf{T}_2 = \mathbf{C} \mathbf{T}_c = \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \end{bmatrix} \mathbf{T}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_m \end{bmatrix} \quad (18)$$

Finally, it yields

$$\mathbf{A}^\circ = \mathbf{T}_c^{-1} \mathbf{A} \mathbf{T}_c = \mathbf{T}_2^{-1} \mathbf{T}_1^{-1} \mathbf{A} \mathbf{T}_1 \mathbf{T}_2 \quad (19)$$

Thus, (16), (18), and (19) implies (11). This concludes the proof. ■

Note, the structure of \mathbf{T}_1^{-1} is not unique and others can be obtained by permutations of the first $n-m$ rows in the structure defined in (12).

2.3 System Modes Properties

Proposition 3. Given system eigenstructure with distinct eigenvalues then for $j, k \in \{1, 2, \dots, n\}$, $l \in \{1, 2, \dots, m\}$, $m = r$

i. the k -th mode $(s - s_k)$ is unobservable from the l -th system output if the l -th row of matrix \mathbf{C} is orthogonal to the k -th eigenvector of the closed-loop system matrix \mathbf{A}_c , i.e. with $j \neq k$

$$\mathbf{c}_l^T \mathbf{n}_k = \mathbf{n}_j^T \mathbf{n}_k = 0, \quad \mathbf{C}^T = \begin{bmatrix} \mathbf{c}_1 & \cdots & \mathbf{c}_m \end{bmatrix} \quad (20)$$

ii. the k -th mode $(s - s_k)$ is uncontrollable from the l -th column of matrix \mathbf{B} is orthogonal to the k -th eigenvector of the closed-loop system matrix \mathbf{A}_c , i.e. with $j \neq k$

$$\mathbf{n}_k^T \mathbf{b}_l = \mathbf{n}_k^T \mathbf{n}_j = 0, \quad \mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_r \end{bmatrix} \quad (21)$$

Proof. (see e.g. [13]) Let \mathbf{n}_k is the k -th right eigenvector corresponding to the eigenvalue s_k , i.e.

$$\mathbf{A}_c \mathbf{n}_k = (\mathbf{A} - \mathbf{BK}) \mathbf{n}_k = s_k \mathbf{n}_k \quad (22)$$

By definition, the closed-loop system resolvent kernel is

$$\mathbf{Y} = (s\mathbf{I}_n - \mathbf{A}_c)^{-1} \quad (23)$$

If the closed-loop system matrix is with distinct eigenvalues, (22) can be written in the compact form

$$\mathbf{A}_c [\mathbf{n}_1 \cdots \mathbf{n}_n] = [\mathbf{n}_1 \cdots \mathbf{n}_n] \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{bmatrix} \quad (24)$$

$$\mathbf{A}_c \mathbf{N} = \mathbf{N} \mathbf{S}, \quad \mathbf{N}^{-1} = \mathbf{N}^T \quad (25)$$

respectively, where

$$\mathbf{S} = \text{diag}[s_1 \cdots s_n], \quad \mathbf{N} = [\mathbf{n}_1 \cdots \mathbf{n}_n] \quad (26)$$

Using the orthogonal property given in (25), the resolvent kernel of the system takes the next form

$$\mathbf{Y} = (s\mathbf{N}\mathbf{N}^{-1} - \mathbf{N}\mathbf{S}\mathbf{N}^{-1})^{-1} = \mathbf{N}(s\mathbf{I} - \mathbf{S})^{-1}\mathbf{N}^T \quad (27)$$

$$\mathbf{Y} = [\mathbf{n}_1 \cdots \mathbf{n}_n] \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{n}_1^T \\ \vdots \\ \mathbf{n}_n^T \end{bmatrix} \quad (28)$$

$$\mathbf{Y} = \sum_{h=1}^n \frac{\mathbf{n}_h \mathbf{n}_h^T}{s - s_h} \quad (29)$$

respectively. Thus, the closed loop transfer functions matrix takes form

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A}_c)^{-1}\mathbf{B}\mathbf{L} = \sum_{h=1}^n \frac{\mathbf{C}\mathbf{n}_h \mathbf{n}_h^T \mathbf{B}}{s - s_h} \mathbf{L} \quad (30)$$

It is obvious that (30) implies (20), (21). This concludes the proof. ■

3. Eigenstructure Assignment

In the pole assignment problem, a feedback gain matrix \mathbf{K} is sought so that the closed-loop system has a prescribed eigenvalues spectrum $\Omega(\mathbf{A}_c) = \{s_h : \Re(s_h) < 0, h = 1, 2, \dots, n\}$. Note, the spectrum $\Omega(\mathbf{A}_c)$ is closed under complex conjugation, and the observability and controllability of modes is determined by the closed-loop eigenstructure.

Considering the same assumptions as above then (22) can be rewritten as

$$[s_h \mathbf{I} - \mathbf{AB}] \begin{bmatrix} \mathbf{n}_h \\ \mathbf{K}\mathbf{n}_h \end{bmatrix} = \mathbf{L}_h \begin{bmatrix} \mathbf{n}_h \\ \mathbf{K}\mathbf{n}_h \end{bmatrix} = \mathbf{0} \quad (31)$$

where $\mathbf{L}_h \in \mathbf{R}^{n \times (n+r)}$,

$$\mathbf{L}_h = [s_h \mathbf{I} - \mathbf{AB}] \quad (32)$$

Subsequently, the singular value decomposition (SVD) of \mathbf{L}_h gives

$$\begin{bmatrix} \mathbf{u}_{h1}^T \\ \vdots \\ \mathbf{u}_{hn}^T \end{bmatrix} \mathbf{L}_h [\mathbf{v}_{h1} \cdots \mathbf{v}_{hn} \quad \mathbf{v}_{h,(n+1)} \cdots \mathbf{v}_{h,(n+r)}] = \quad (33)$$

$$= \begin{bmatrix} \sigma_{h1} & & & & \\ & \ddots & & & \\ & & \sigma_{hn} & & \\ & & & \mathbf{0}_{n+1} & \cdots & \mathbf{0}_{n+r} \end{bmatrix}$$

where $\{\mathbf{u}_{hl}^T, l = 1, 2, \dots, n\}$, $\{\mathbf{v}_{hk}, k = 1, 2, \dots, n+r\}$ are sets of the left and right singular vectors of \mathbf{L}_h associated with the set of singular values $\{\sigma_{hl}, l = 1, 2, \dots, n\}$.

It is evident that vectors $\{\mathbf{v}_{hj}, j = n+1, n+2, \dots, n+r\}$ satisfy (31), i.e.

$$\mathbf{L}_h \mathbf{v}_{hj} = [s_h \mathbf{I} - \mathbf{AB}] \mathbf{v}_{hj} = \mathbf{0} \quad (34)$$

The set of vectors $\{\mathbf{v}_{hj}, j = n+1, n+2, \dots, n+r\}$ is a non-trivial solution of (32), and results the null space of $\mathbf{L}_h, h = 1, 2, \dots, n$

$$\begin{bmatrix} \mathbf{n}_h \\ \mathbf{K}\mathbf{n}_h \end{bmatrix} \in \mathcal{N}[s_h \mathbf{I} - \mathbf{AB}] = \mathbf{0} \quad (35)$$

The null space (35) consists of the normalized orthogonal set of vectors. Any combination of these vectors (the span of null space) will provide a vector \mathbf{n}_h which used as an eigenvector produces the desired eigenvalue s_h in the closed-loop system matrix.

Proposition 4. The canonical form eigenstructure optimization provides optimal eigenstructure also for that model from which the canonical form was derived.

Proof. Using (16), (18), (19) and (22) it can be simply written

$$(\mathbf{A} - \mathbf{BK}) \mathbf{n}_k = (\mathbf{T}_c \mathbf{A}^\circ \mathbf{T}_c^{-1} - \mathbf{T}_c \mathbf{B}^\circ \mathbf{K} \mathbf{T}_c \mathbf{T}_c^{-1}) \mathbf{n}_h = \mathbf{T}_c (\mathbf{A}^\circ - \mathbf{B}^\circ \mathbf{K}^\circ) \mathbf{T}_c^{-1} \mathbf{n}_h = s_h \mathbf{n}_h \quad (36)$$

$$(\mathbf{A}^\circ - \mathbf{B}^\circ \mathbf{K}^\circ) \mathbf{T}_c^{-1} \mathbf{n}_h = s_h \mathbf{T}_c^{-1} \mathbf{n}_h \quad (37)$$

$$(\mathbf{A}^\circ - \mathbf{B}^\circ \mathbf{K}^\circ) \mathbf{n}_h^\circ = \mathbf{A}_c^\circ \mathbf{n}_h^\circ = s_h \mathbf{n}_h^\circ \quad (38)$$

respectively, where $\mathbf{K}^\circ = \mathbf{K} \mathbf{T}_c, \mathbf{n}_h = \mathbf{T}_c \mathbf{n}_h^\circ$.

Writing compactly the set $\{\mathbf{n}_h = \mathbf{T}_c \mathbf{n}_h^\circ, h = 1, 2, \dots, n\}$ as follows

$$\mathbf{N} = \mathbf{T}_c \mathbf{N}^\circ, \quad \mathbf{N}^{-1} = \mathbf{N}^{\circ T} \mathbf{T}_c^{-1} \quad (39)$$

then using (27), (30), (39) it yields

$$\begin{aligned} \mathbf{G}(s) &= \mathbf{C}\mathbf{Y}\mathbf{B}\mathbf{L} = \mathbf{C}\mathbf{N}(s\mathbf{I} - \mathbf{S})^{-1}\mathbf{N}^{-1}\mathbf{B}\mathbf{L} = \\ &= \mathbf{C}\mathbf{T}_c \mathbf{N}^\circ (s\mathbf{I} - \mathbf{S})^{-1} \mathbf{N}^{\circ T} \mathbf{T}_c^{-1} \mathbf{B}\mathbf{L} = \\ &= \mathbf{C}^\circ \mathbf{N}^\circ (s\mathbf{I} - \mathbf{S})^{-1} \mathbf{N}^{\circ T} \mathbf{B}^\circ \mathbf{L} \end{aligned} \quad (40)$$

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A}_c)^{-1}\mathbf{B}\mathbf{L} = \mathbf{C}^\circ (s\mathbf{I} - \mathbf{A}_c^\circ)^{-1} \mathbf{B}^\circ \mathbf{L} \quad (41)$$

respectively, and evidently

$$\mathbf{G}(s) = \sum_{h=1}^n \frac{\mathbf{C}\mathbf{n}_h \mathbf{n}_h^T \mathbf{B}}{s - s_h} \mathbf{L} = \sum_{h=1}^n \frac{\mathbf{C}^\circ \mathbf{n}_h^\circ \mathbf{n}_h^{\circ T} \mathbf{B}^\circ}{s - s_h} \mathbf{L} \quad (42)$$

It is obvious that optimizing product $\mathbf{C}^\circ \mathbf{n}_h^\circ$ then it is optimized also $\mathbf{C}\mathbf{n}_h$. This concludes the proof. ■

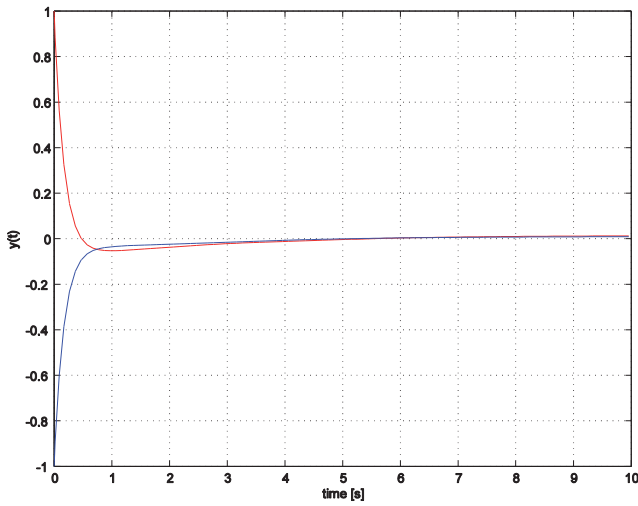


Fig. 1 System output response

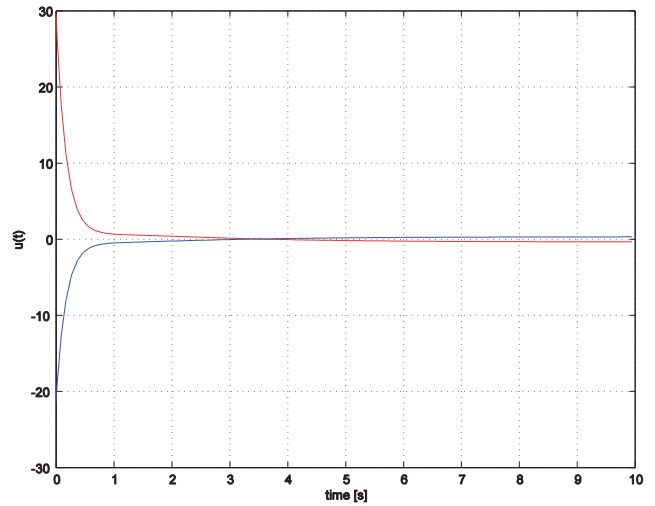


Fig. 2 Control actions

4. Parameter Design

Using eigenvector orthogonal properties, (22) can be rewritten for $h=1, 2, \dots, n$ as follows

$$(s_h \mathbf{I} - \mathbf{A}^\circ) \mathbf{n}_h^\circ = -\mathbf{B}^\circ \mathbf{K}^\circ \mathbf{n}_h^\circ = -\mathbf{B}^\circ \mathbf{r}_h^\circ \quad (43)$$

$$\mathbf{n}_h^\circ = -(s_h \mathbf{I} - \mathbf{A}^\circ)^{-1} \mathbf{B}^\circ \mathbf{r}_h^\circ = \mathbf{V}_h^\circ \mathbf{r}_h^\circ \quad (44)$$

respectively, where

$$\mathbf{r}_h^\circ = \mathbf{K}^\circ \mathbf{n}_h^\circ, \quad \mathbf{V}_h^\circ = -(s_h \mathbf{I} - \mathbf{A}^\circ)^{-1} \mathbf{B}^\circ \quad (45)$$

Subsequently, it can be obtained

$$\mathbf{r}_h^\circ = \mathbf{V}_h^{\circ T} \mathbf{n}_h^\circ \quad (46)$$

where

$$\mathbf{V}_h^{\circ T} = (\mathbf{V}_h^{\circ T} \mathbf{V}_h^\circ)^{-1} \mathbf{V}_h^{\circ T} \quad (47)$$

is Moore-Penrose pseudo inverse of \mathbf{V}_h° .

Of interest are the eigenvectors of the closed-loop system having minimal orthogonal projection to rows of the orthogonal complement $\mathbf{C}^{\circ T \perp}$ of the output matrix $\mathbf{C}^{\circ T}$ and associated with m element eigenvalues subset $\rho(\mathbf{A}^\circ)$ ($m = \text{rank}(\mathbf{C}^\circ)$, $\rho(\mathbf{A}^\circ) \subset \Omega(\mathbf{A}^\circ)$) of the desired closed-loop eigenvalues set $\Omega(\mathbf{A}^\circ) = \{s_h, \Re(s_h) < 0, h=1, 2, \dots, n\}$ $\Omega(\mathbf{A}^\circ) = \Omega(\mathbf{A})$. The rest $(n-m)$ eigenvalues can be associated with rows of the complement matrix \mathbf{C}^\bullet obtained in such way that all zero elements in \mathbf{C}° be changed to ones, and all ones to zeros. Note, direct use of \mathbf{C}° maximize matrix weights of modes.

Let $\rho(\mathbf{A}^\circ) = \{s_h, \Re(s_h) < 0, h=1, 2, \dots, n\}$, then

$$\mathbf{r}_h^* = \mathbf{V}_h^{\circ T} \mathbf{c}_h^{\circ T \perp T}, \quad h=1, 2, \dots, m \quad (48)$$

$$\mathbf{r}_h^\bullet = \mathbf{V}_h^{\circ T} \mathbf{c}_h^{\bullet T}, \quad h=m+1, \dots, n \quad (49)$$

Thus, computing

$$\mathbf{n}_h^* = \mathbf{V}_h^\circ \mathbf{r}_h^*, \quad \mathbf{n}_h^\bullet = \mathbf{V}_h^\circ \mathbf{r}_h^\bullet \quad (50)$$

it is possible to construct and to separate the matrix \mathbf{Q}° of the form

$$\mathbf{Q}^\circ = [\mathbf{v}_1^* \quad \dots \quad \mathbf{v}_m^* \quad \mathbf{v}_{m+1}^\bullet \quad \dots \quad \mathbf{v}_n^\bullet] = \begin{bmatrix} \mathbf{P}^\circ \\ \mathbf{R}^\circ \end{bmatrix} \quad (51)$$

with $\mathbf{P}^\circ \in \mathbf{R}^{n \times n}$, $\mathbf{R}^\circ \in \mathbf{R}^{r \times n}$ such that

$$\mathbf{K} = \mathbf{R}^\circ \mathbf{P}^{\circ -1}, \quad \mathbf{K} = \mathbf{K}^\circ \mathbf{T}_c^{-1} \quad (52)$$

4. Illustrative Example

The system under consideration was described by (1), (2), where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -9 & -5 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 5 \end{bmatrix}, \quad \mathbf{C}^T = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix}$$

Constructing the transformation matrices

$$\mathbf{T}_c^{-1} = \begin{bmatrix} 4.0 & 0.5 & -2.5 \\ 1.0 & 2.0 & 1.0 \\ 1.0 & 1.0 & 0.0 \end{bmatrix}, \quad \mathbf{T}_c = \begin{bmatrix} 1.0 & 2.5 & -5.5 \\ -1.0 & -2.5 & 6.5 \\ 1.0 & 3.5 & -7.5 \end{bmatrix}$$

the system model canonical form parameters were computed as

$$\mathbf{A}^\circ = \begin{bmatrix} -1 & 10.5 & 6 \\ 0 & -3.0 & -2 \\ 0 & 1.0 & -1 \end{bmatrix}, \quad \mathbf{B}^\circ = \begin{bmatrix} 0 & 0 \\ 7 & 10 \\ 3 & 4 \end{bmatrix}, \quad \mathbf{C}^\circ = [\mathbf{0} \quad \mathbf{I}_2]$$

Thus, considering $\Omega(\mathbf{A}^\circ) = \{-0.5, -1.2, -6\}$ it is

$$\mathbf{V}_1^\circ = \begin{bmatrix} -37.3846 & -54.4615 \\ 0.7692 & 0.9231 \\ -4.4615 & -6.1538 \end{bmatrix}$$

$$\mathbf{V}_2^\circ = \begin{bmatrix} -10.0610 & -5.4815 \\ 4.5122 & 6.0976 \\ -7.5610 & -10.4878 \end{bmatrix}$$

$$\mathbf{V}_3^\circ = \begin{bmatrix} -5.2059 & -7.3059 \\ 2.4118 & 3.4118 \\ 0.1176 & 0.0176 \end{bmatrix}$$

and with $\mathbf{c}^{\circ T \perp} = [1 \ 0 \ 0]$, $\mathbf{c}_1^{\circ T} = [1 \ 0 \ 1]$ it yields

$$\mathbf{r}_1^{\circ} = \begin{bmatrix} 0.3891 \\ -0.2854 \end{bmatrix}, \quad \mathbf{r}_2^{\circ} = \begin{bmatrix} -0.1645 \\ 0.1194 \end{bmatrix}, \quad \mathbf{r}_3^{\circ} = \begin{bmatrix} 18.4978 \\ -13.2737 \end{bmatrix}$$

$$\mathbf{n}_1^{\circ T} = [0.9983 \ 0.0358 \ 0.0205]$$

$$\mathbf{n}_2^{\circ T} = [0.9997 \ -0.0144 \ -0.0082]$$

$$\mathbf{n}_3^{\circ T} = [0.9983 \ 0.0358 \ 0.0205]$$

Constructing the matrix \mathbf{Q}°

$$\mathbf{Q}^{\circ} = \begin{bmatrix} 0.9983 & 0.9997 & 0.6788 \\ 0.0358 & 0.0144 & -0.6745 \\ 0.0205 & -0.0082 & 0.6146 \\ 0.3891 & -0.1645 & 18.4978 \\ -0.2854 & 0.1194 & -13.2737 \end{bmatrix} = \begin{bmatrix} \mathbf{P}^{\circ} \\ \mathbf{R}^{\circ} \end{bmatrix}$$

the control law parameters satisfying (52) are

$$\mathbf{K}^{\circ} = \begin{bmatrix} -0.0062 & -3.7944 & 25.9402 \\ 0.0036 & 2.6301 & -18.7151 \end{bmatrix}$$

$$\mathbf{K} = \begin{bmatrix} 22.1212 & 18.3483 & -3.7990 \\ -16.0707 & -13.4532 & 2.6211 \end{bmatrix}$$

It is possible to verify that closed-loop system matrix eigenvalues belongs to the desired one.

In the presented Fig. 1, 2 the example is shown of the unforced closed-loop system output response, as well as control actions, where nonzero initial state was considered.

	$\mathbf{c}_1^{\circ T} \leftarrow -0.5$	$\mathbf{c}_1^{\circ T} \leftarrow -1.2$	$\mathbf{c}_1^{\circ T} \leftarrow -6$
$\ \mathbf{K}^{\circ}\ $	17.2524	15.9878	32.3181
$\ \mathbf{K}\ $	46.2521	43.7478	35.8664
ISE	1.8746	0.8183	0.1712
IAE	3.7033	1.6238	0.5651
ITAE	7.0822	1.8607	0.7077

Tab.1 Comparison of performance indicators for different combination of eigenvalue assignment

Concluding remarks

This paper provides a design method for memory-free controllers where the general problem of assigning the eigenstructure for state variable mode decoupling in state feedback control design is considered. The method exploits standard numerical optimization procedures to manipulate the system feedback gain matrix as a direct design variable. The manipulation is accomplished in a manner that produces desired system global performance by pole placement and output dynamics by modification of the mode observability.

With generalization of the known algorithms for pole assignment the modified exposition of the problem is presented here to handle the optimized structure of the left eigenvector set in state feedback control design. Presented method makes full use of the freedom provided by eigenstructure assignment to find a controller which stabilizes the closed-loop system. Therefore, the feedback control law has a clear physical meaning and provides a valid design me-

thod of the controller for real systems. It is shown by appropriately assigning closed-loop eigenstructure in state feedback control the overall stability is achieved. Finally the design methodology is illustrated by an example.

Acknowledgment

The work presented in this paper was supported by VEGA, Grant Agency of Ministry of Education and Academy of Science of Slovak Republic under Grant No. 1/0256/11, as well by Research & Development Operational Programme Grant No. 26220120030 realized in Development of Centre of Information and Communication Technologies for Knowledge Systems. These supports are very gratefully acknowledged.

References

- [1] BACHELIER, O., BOSCHE, J., and MEHDI, D.: On pole placement via eigenstructure assignment Approach, *IEEE Transactions on Automatic Control*, 51:9, 1554-1558, 2006.
- [2] BURNS, R.S.: *Advanced Control Engineering*, Butterworth-Heinemann, Oxford, 2001.
- [3] DATTA, B.N.: *Numerical Methods for Linear Control Systems: Design and Analysis*, Elsevier, London, 2004.
- [4] GOLUB, G.H. and LOAN Van C.F.: *Matrix Computations*, The John Hopkins University Press, Baltimore, 1989.
- [5] FILASOVÁ, A.: Robust control design for large-scale uncertain dynamic system, In *New Trends in Design of Control Systems*, STU, Bratislava, 247-432, 1997.
- [6] FILASOVÁ, A.: Robust control design: An optimal control approach, In *Proceeding of the IEEE International Conference on Intelligent Engineering Systems INES'99*, Stará Lesná, Slovakia, 515-518, 1999.
- [7] FILASOVÁ, A. and KROKAVEC, D.: On sensor faults estimation using sliding mode observers, In *Conference on Control and Fault-Tolerant Systems SysTol'10*, Nice, France, 2010, 44-49.
- [8] FILASOVÁ, A. and KROKAVEC, D.: State estimate based control design using the unified algebraic approach, *Archives of Control Sciences*, 20:1, 5-18, 2010.
- [9] FÓNOD, R. and KOCSIS, P.: State feedback control design using eigenstructure decoupling, In *Proceeding of the 18th International Conference on Process Control PC'11*, Tatranská Lomnica, Slovakia, 268-272, 2011.
- [10] IPSEN, I.C.S.: *Numerical Matrix Analysis. Linear Systems and Least Squares*, SIAM, Philadelphia, 2009.
- [11] KAUTSKY, J., NICHOLS, N. and DOOREN Van, P.: Robust pole assignment in linear state feedback, *International Journal of Control*, 41:5, 1129-1155, 1985.
- [12] KOCSIS, P. and KROKAVEC, D.: State variables mode decoupling in state control design for linear MIMO systems, In *International Conference Cybernetics and Informatics 2009*, Ždiar, Slovak Republic, 44.1-44.8, 2009.
- [13] KROKAVEC, D. and FILASOVÁ, A.: *Dynamic Systems Diagnosis*, Elfa, Košice, 2007. (in Slovak)
- [14] POZNYAK, A.S.: *Advanced Mathematical Tools for Automatic Control Engineers: Deterministic Techniques*, Elsevier, London 2008.
- [15] SOBEL, K.M. and LALLMAN, F.J.: Eigenstructure assignment for the control of highly augmented aircraft, *Jour-*

nal of Guidance, Control and Dynamics, 12:3, 318-324, 1989.

[16] WANG, Q.G.: *Decoupling Control*, Springer-Verlag, Berlin, 2003.

[17] WONHAM, W.M.: *Linear Multivariable Control: A Geometric Approach*, Springer-Verlag, New York, 1985.

[18] XU, X.H. and XIE, X.K.: Eigenstructure assignment by output feedback in descriptor systems, *IMA Journal of Mathematical Control and Information*, 12, 127-132, 1995.

Abstrakt

Cieľom návrhu je syntetizovať stavové riadenie tak, aby vzhľadom na predpísané vlastné hodnoty matice dynamiky uzavretého obvodu boli odpovedajúce vlastné vektory čo najviac podobné vlastným vektorom systému s rozviazaným vstupom a výstupom. V príspevku je ukázané, že existujúci stupeň voľnosti v návrhu, reprezentovaný množinou parametrických vektorov a množinou pólov uzavretého obvodu, možno využiť na dosiahnutie vopred zadaných charakteris-

tík uzavretého obvodu. Príklad a číselné simulácie ukazujú, že takýto spôsob návrhu je efektívny a jednoduchý.

Ing. Pavol Kocsis
Ing. Róbert Fónod

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 2749
Fax: ++421 55 625 3574
E-mail: robert.fonod@student.tuke.sk, pavol.kocsis@tuke.sk

Discrete-time disturbance decoupling of coupled tanks

Miroslav Halás

Abstract

Mathematical technicalities, involved in the modern theory of nonlinear control systems, many times prevent a wider use of the impressive theoretical results in practice. Attempts to overlap this gap between theory and practice are usually more than welcome and form the main scope of our interest in this work. An important control problem given by the disturbance decoupling is studied for a real laboratory model of coupled tanks. Since the theoretical solution to the disturbance decoupling problem does not satisfy practical control requirements it is modified accordingly. Experiments on the real plant are included as well and show that the disturbances practically do not affect the system output.

Keywords: nonlinear discrete-time systems, applications, algebraic methods, disturbance decoupling, coupled tanks

Introduction

The modern theory of nonlinear control systems all, continuous-time, discrete-time and time-delay, owes a large part of its success to the systematic use of differential algebraic methods. Since early 80's of the last century this has been forming the scope of interest of many authors in a number of works, see for instance [1,2,3,4,5,6,11,14] and references therein. Nowadays, such methods offer solutions to a wide range of nonlinear control problems including feedback linearization, model matching, disturbance decoupling, realization problem, non-interacting control, observer design and many others.

However, a price one has to pay for such impressive and elegant solutions is given by a necessity to involve many mathematical technicalities. Obviously, this prevents a wider use of the theoretical results in practice, making the big gap between control theory and control practice even bigger in this case. It is generally known that in practice the way of dealing with nonlinear control systems is many times based just on the linearization in a fixed operating point and then methods of linear control systems are applied. Therefore, attempts to overlap the gap are usually more than welcome and form the main scope of our interest in this work. In particular, an important control problem given by the disturbance decoupling, which is quite frequent control problem in practice, is studied. We begin with the theoretical solution of [4] and apply it to the laboratory model of coupled tanks, which is a demonstrative and well known system having contact points to many real control processes, for instance from chemical engineering. It is shown that the theoretical solution cannot be directly applied and additional problems, related for instance to the difference between model and real system, have to be considered as well. Similar solution as discussed in this paper has recently been given in [12] for continuous-time case, while here the discrete-time counterpart is treated. Certain contact points exist also to the non-interacting problem studied in [8]. Finally, for additional existing results of the disturbance decoupling problem for nonlinear discrete-time systems the reader is referred to, for instance, [10] where a simple inversion-based solution is

given and to [7] where a more advanced differential geometric solution can be found.

1. Disturbance decoupling

We begin with an introduction to the disturbance decoupling problem of nonlinear control systems as discussed in [4] to which the reader is referred for additional details and references. The ideas can easily be carried over to the discrete-time systems.

For the sake of simplicity we introduce the following notation. For any variable $\xi(t)$ we write only ξ and for its time shifts $\xi(t+T)$, $\xi(t+2T)$ we write ξ^+ , ξ^{++} respectively, or, in general, ξ^{k+} for $\xi(t+kT)$, where T is a sampling period.

Using the above notation the systems considered in this paper are objects of the form

$$\begin{aligned}x^+ &= f(x, u) \\ y &= g(x)\end{aligned}$$

where $x \in \mathbf{R}^n$, $u, y \in \mathbf{R}$ and the entries of f and g are meromorphic functions from the difference field denoted by \mathcal{K} . For more details see [2,9,11].

In the disturbance decoupling our task is to design, if possible, a control law such that the disturbances do not affect the system output. Technically speaking, the solution consists of finding a feedback under which a subspace of the state space, affected by the disturbances, becomes unobservable in the compensated system. This situation can be explained by the following introductory system

$$\begin{aligned}x_1^+ &= x_2 u \\ x_2^+ &= u \\ y &= x_1\end{aligned}$$

where u is the disturbance.

As can be seen, through x_2 the disturbance u affects the system output

$$y^{++} = u^+ u$$

However, the state feedback $u = v/x_2$, with v being an input to the compensated system, makes x_2 unobservable in the compensated system and thus decouples the disturbance w from the system output

$$y^+ = v$$

The general solution follows the same idea. That is, if possible, make unobservable the subspace of the state space affected by the disturbance.

Problem statement: Consider the SISO system

$$\begin{aligned} x^+ &= f(x, u) + p(x)u \\ y &= g(x) \end{aligned}$$

where the state $x \in \mathbb{R}^n$, the disturbance $w \in \mathbb{R}^s$ and the entries of f , g and p are elements of the difference field of meromorphic functions \mathcal{K} . Find, if possible, a static state feedback

$$u = \alpha(x, v)$$

such that

$$dy^{[i]} \in \text{span}_{\mathcal{K}}\{dx, dv, \dots, dv^{[i]}\}$$

for any $i \in \mathbb{N}$.

Theorem 1: Let $\mathcal{X} = \text{span}_{\mathcal{K}}\{dx\}$ and $\mathcal{Y} = \text{span}_{\mathcal{K}}\{dy^{[i]}; i \geq 0\}$. The disturbance decoupling problem is solvable if and only if $p(x) \perp \mathcal{X} \cap \mathcal{Y}$.

Proof: The proof follows the same line as in [4], however, carried over to the discrete-time case.

2. Coupled tanks

Coupled tanks are well-known and illustrative system having contact points to many real control processes. For that reason practically each laboratory which activities are related to the system and control theory possesses such a plant. In this section, the mathematical model of the laboratory plant is built up, from its identification to the nonlinear discrete-time state space model. Then, the disturbance decoupling is applied.

2.1 System identification

We restrict our attention to a standard coupled two-tank system, however, with all three valves active. The structure of such a system is depicted in Fig. 1. Our aim is to control the level in the first tank which is, however, coupled with the second tank by a valve with the flow coefficient c_{12} . Each of the tanks is equipped by a valve itself, having the flow coefficients c_{11} and c_{22} respectively. However, the valve c_{22} is considered here as the disturbance w . Thus, we deal here with a SISO system which can be modeled by the following state-space equations

$$\begin{aligned} \dot{x}_1 &= \frac{1}{A}u - c_{12} \text{sign}(x_1 - x_2) \sqrt{|x_1 - x_2|} - c_{11} \sqrt{x_1} \\ \dot{x}_2 &= c_{12} \text{sign}(x_1 - x_2) \sqrt{|x_1 - x_2|} - wc_{22} \sqrt{x_2} \\ y &= x_1 \end{aligned} \quad (1)$$

where x_1 and x_2 are levels in tank 1 and tank 2 respectively and A is a cross-section of the tanks, see Fig. 1. Note that both tank 1 and tank 2 have the identical cross-sections here. The disturbance $w \in \{0, 1\}$, depending on whether the valve c_{22} is switched off or on respectively. In this case the level of a liquid in tank 2 might be greater than in tank 1. For that reason a more general model (1) has to be used.

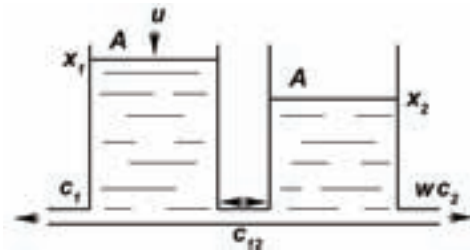


Fig.1 Coupled tanks.

To identify the system we have to find, besides the cross-sections A , the values of flow coefficients c_{11} , c_{22} and c_{12} .

The usual methods to treat the identification are based on applying a couple of certain experiments and measurements. Then the coefficients are computed by using either the steady-states of the system or the system linearization in a fixed operating point. However, both of them are rather slow. In addition, it is, usually, recommended to find a set of values in different steady states or operating points respectively and take their average finally. For that reason we, in what follows, suggest a different approach to the system identification which is based on finding a solution to the reduced nonlinear differential equations of the system (1). As a result we will be able to compute all the coefficients only by measuring the time of the respective experiments.

To identify the flow coefficient c_{11} suppose that all valves are closed and the pump is inactive. Let $x_{10} \neq 0$ be a level of a liquid in tank 1. The experiment consists of opening the valve c_{11} only and measuring the time τ that it takes to empty the tank from an initial value x_{10} to a final value x_{11} . Obviously, this situation can be modeled by the reduced nonlinear differential equation

$$\dot{x}_1 = -c_{11} \sqrt{x_1}$$

Even though the equation is nonlinear the solution can easily be found as follows

$$\begin{aligned} \frac{dx_1}{dt} &= -c_{11} \sqrt{x_1} \\ \int_{x_{10}}^{x_{11}} \frac{1}{\sqrt{x_1}} dx_1 &= - \int_0^\tau c_{11} dt \\ \left[2\sqrt{x_1} \right]_{x_{10}}^{x_{11}} &= [-c_{11}t]_0^\tau \\ 2(\sqrt{x_{11}} - \sqrt{x_{10}}) &= -c_{11}\tau \end{aligned}$$

Finally

$$c_{11} = \frac{2(\sqrt{x_{10}} - \sqrt{x_{11}})}{\tau}$$

If the final value x_{11} is chosen to be 0, which is usually the most reasonable choice, then the formula reduces to

$$c_{11} = \frac{2\sqrt{x_{10}}}{\tau} \quad (2)$$

where τ is the time it takes to empty tank 1 completely from the initial value x_{10} .

Clearly, the analogous experiment can be repeated for the second tank giving us the formula

$$c_{22} = \frac{2\sqrt{x_{20}}}{\tau} \quad (3)$$

where this time τ is the time it takes to empty tank 2 completely from its initial value x_{20} .

To identify the flow coefficient c_{12} a more advanced experiment is needed. Suppose that all valves are closed and both pumps inactive. Let $x_{10} \neq 0$ be a level of a liquid in tank 1

and $x_{20} < x_{1c}$ be a level of a liquid in tank 2. This time the experiment consists of opening the valve e_{12} only and measuring the time τ it takes the level in tank 1 decrease from the initial value x_{1c} to the final value x_{11} . Such a situation can be modeled by the following nonlinear differential equations

$$\begin{aligned}\dot{x}_1 &= -c_{12}\sqrt{x_1 - x_2} \\ \dot{x}_2 &= c_{12}\sqrt{x_1 - x_2}\end{aligned}$$

However, one can use either of them to find the solution. For instance the first equation yields

$$\begin{aligned}\frac{dx_1}{dt} &= -c_{12}\sqrt{x_1 - x_2} \\ \frac{1}{\sqrt{x_1 - x_2}} dx_1 &= -c_{12} dt\end{aligned}$$

Note that the situation during the experiment implies that $x_{10} - x_1 = x_2 - x_{2c}$ and thus substituting $x_2 = x_{10} + x_{20} - x_1$ gives us

$$\begin{aligned}\int_{x_{10}}^{x_{11}} \frac{1}{\sqrt{2x_1 - x_{10} - x_{20}}} dx_1 &= -\int_0^\tau c_{12} dt \\ \left[\sqrt{2x_1 - x_{10} - x_{20}} \right]_{x_{10}}^{x_{11}} &= [-c_{12}t]_0^\tau \\ \sqrt{2x_{11} - x_{10} - x_{20}} - \sqrt{2x_{10} - x_{10} - x_{20}} &= -c_{12}\tau\end{aligned}$$

Finally

$$c_{12} = \frac{\sqrt{2x_{10} - x_{10} - x_{20}} - \sqrt{2x_{11} - x_{10} - x_{20}}}{\tau}$$

Here, if the initial value x_{2c} is chosen to be 0 and the final value x_{11} is chosen to be $x_{10}/\sqrt{2}$, that is the levels in both tanks finally equal each other (note that the tanks have the identical cross-sections) the formula reduces to

$$c_{12} = \frac{\sqrt{x_{10}}}{\tau} \quad (4)$$

where τ is the time it takes the levels in both tanks equal each other.

Using the above formulas (2), (3) and (4) the flow coefficients e_1 , e_2 and e_{12} of the laboratory plant were identified as 1.17×10^{-5} , 1.17×10^{-5} and 2.65×10^{-5} respectively. Finally, the cross sections of both tanks are approximately $A = 10.18 \times 10^{-4} \text{ m}^2$.

2.2 Discrete-time state-space representation

To find a discrete-time state-space representation of the system (1) one needs to find a solution to the set of nonlinear differential equations and sample it by the sampling period T . Since the system equations (1) involve nonlinear functions we are, in general, not able to find any. In such a case one usually has to rely on approximations only. One of the possibilities is to employ the Taylor series expansion.

Assume that

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

where f is analytic. Then one can write

$$x(t+T) = x(t) + \dot{x}(t)T + \frac{\ddot{x}(t)}{2!}T^2 + \frac{x^{(3)}(t)}{3!}T^3 + \dots$$

However, it is usually sufficient to consider only the first two terms of the Taylor series expansion to approximate the system behaviour in which case one gets the well-known Euler approximation

$$\dot{x}(t) \approx \frac{x(t+T) - x(t)}{T}$$

Using such an approximation one can find the discrete-time state space model from (1) as

$$\begin{aligned}x_1^+ &= x_1 + \frac{T}{A}u - c_{12}T \text{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} \\ &\quad - c_1 T \sqrt{x_1} \\ x_2^+ &= x_2 + c_{12}T \text{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} \\ &\quad - u c_2 T \sqrt{x_2} \\ y &= x_1\end{aligned} \quad (5)$$

2.3 Disturbance decoupling problem

The standard theoretical solution to the disturbance decoupling problem, as outlined in Section 1, does not meet basic practical control requirements, as shown in what follows, and thus it is necessary to modify it accordingly.

To proceed with the disturbance decoupling we compute

$$y^+ = x_1 + \frac{T}{A}u - c_{12}T \text{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} - c_1 T \sqrt{x_1}$$

Since y^+ directly depends on the input u , that is the relative degree of the system is 1, and it is not affected by the disturbance w , it can be decoupled. Note that in according to Theorem 1 we have $\mathcal{X} \cap \mathcal{Y} = \text{span}_{\mathbb{R}}\{dx_1\}$ and, therefore, any $v \in \mathcal{X} \cap \mathcal{Y}$ is a vector of the form $v = \alpha dx_1 = (\alpha \ 0) \cdot (dx_1 \ dx_2)^T$ where $\alpha \in \mathbb{K}$. Thus, the scalar product of $(\alpha \ 0)$ and $p(x) = (0, -c_2\sqrt{x_2})^T$ equals 0 meaning that $p(x)$ is orthogonal to $\mathcal{X} \cap \mathcal{Y}$.

By solving for u the equation

$$y^+ = \mathbf{u}$$

one gets

$$u = \frac{A}{T}v - \frac{A}{T}x_1 + c_{12}A \text{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} + c_1 A \sqrt{x_1} \quad (6)$$

where \mathbf{u} represents input to the compensated system which is reduced to the first order linear system $y^+ = \mathbf{u}$ with the transfer function

$$F(z) = \frac{1}{z} \quad (7)$$

However, from a practical point of view the compensated system cannot respond in one sampling period T , like its transfer function (7) says,

at least for lower sampling periods T , for we have a controller output constraint $u \in (0, q_{max}]$ where q_{max} is upper limit of the pump capacity. On the other side for higher sampling periods T the discrete-time approximation (5) of the continuous-time system (1) might no longer be sufficient. In addition, there obviously exist additional differences between the real plant and its continuous-time model (1) that have not been considered. For that reason, the real compensated system will possess oscillations even for not that high sampling periods T when the discrete-time approximation (5) is still accurate. Last but not least, the feedback (6) is not a controller at all. Obviously, it is only a static state feedback achieving the disturbance decoupling, however, with no intention to track the reference signal or to eliminate unmodeled disturbances. For all the aspects listed above, such a solution is practically not applicable and needs to be modified accordingly.

There exist several possibilities how to overcome the problems. One of them, discussed in [12] for continuous-time case, suggests to modify the feedback (6) such that the whole second tank, together with the disturbance, becomes unobservable. Then, one only has to design a controller for a one-tank system which is, obviously, a trivial task and plenty of solutions have been given. This seems to be a reasonable choice also in the discrete-time case. The feedback (6) can easily be modified to the form

$$u = v + c_{12}A \operatorname{sign}(x_1 - x_2) \sqrt{|x_1 - x_2|} \quad (8)$$

under which the compensated system takes now the form of one-tank system with the discrete-time state-space model

$$\begin{aligned} x_1^+ &= x_1 + \frac{T}{A}v - c_1T\sqrt{x_1} \\ y &= x_1 \end{aligned} \quad (9)$$

Then the controller can easily be designed by the system linearization in a fixed operating point (x_{10}, v_0, y_0) which reads

$$\begin{aligned} \Delta x_1^+ &= \Delta x_1 + \frac{T}{A}\Delta v - \frac{c_1T}{2\sqrt{x_{10}}}\Delta x_1 \\ \Delta y &= \Delta x_1 \end{aligned}$$

where $\Delta x_1 = x_1 - x_{10}$, $\Delta v = v - v_0$ and $\Delta y = y - y_0$. It has the transfer function

$$F(z) = \frac{K}{z - D}$$

where $K = T/A$ and $D = 1 - \frac{c_1T}{2\sqrt{x_{10}}}$.

If one wants the transfer function of the compensated system

$$G(z) = \frac{R(z)F(z)}{1 + R(z)F(z)}$$

to take the form of a first order linear system with the time constant T_1 , then the solution is given by a linear discrete-time **PI** controller with the transfer function

$$R(z) = \frac{(1 - \lambda)(z - D)}{K(z - 1)} = \frac{1 - \lambda}{K} \left(1 + \frac{1 - D}{z - 1} \right)$$

where $\lambda = e^{-T/T_1}$.

Remark: Note that more advanced solution, dealing also with the system linearization, the controller output constraint and two different disturbances to decouple, has been suggested in [13].

The closed loop structure is depicted in Fig. 2. The responses of the real laboratory plant are shown in Fig. 3 where one can observe the differences between the linear **PI**-controller with and without the disturbance decoupling

(8). In the latter the disturbances practically do not affect the system output. However, since we have the constrained controller output and since only a standard **PI**-controller has been used to control the system both solutions admit an overshoot. A non-overshooting solution has been suggested in [12].

The parameters were chosen as follows: $T = 0.25s$, $T_1 = 5s$ and $x_{10} = 0.2m$.

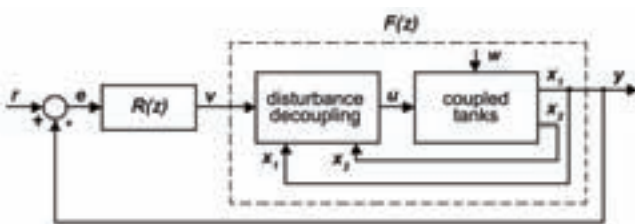


Fig.2 Closed loop.

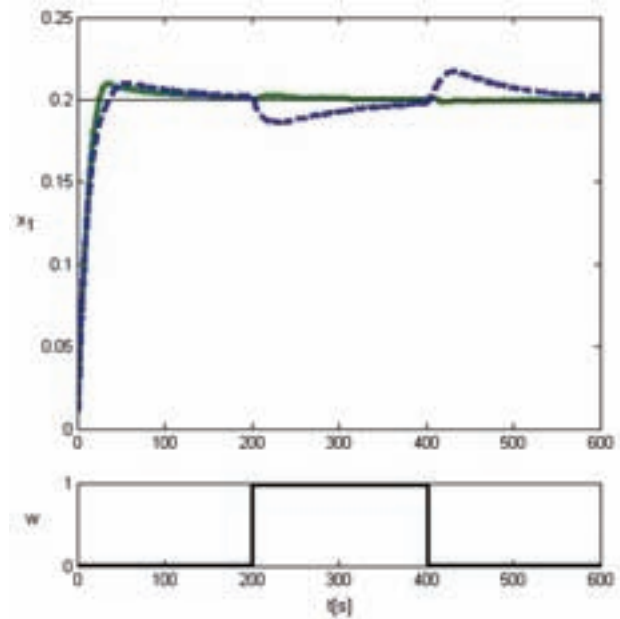


Fig.3 Closed loop responses: **PI**-controller with (solid, green line) and without (dashed, blue line) the disturbance decoupling.

Our final note is related to the slight modification of the disturbance decoupling (8) which is appropriate from a practical point of view and has been implemented in our solution. When the valve e_2 is closed the equations (1) imply that in a steady state one, theoretically, has $x_1 = x_2$. However, in practice, there are differences between x_1 and x_2 caused at least by sensors calibration and noise. Therefore the term $\operatorname{sign}(x_1 - x_2)$ in (8) oscillates between 1 and -1 and thus produces small oscillations of the controller output especially in steady states, which is, of course, inconvenient. The problem can and has been overcome easily by adding a deadzone to the disturbance decoupling making it inactive whenever the difference between x_1 and x_2 is less than $2 \times 10^{-3}m$.

Conclusions

In this work, an attempt to overlap the gap between control theory and control practice was studied. An important practical control problem given by the disturbance decoupling problem were applied to coupled tanks. It was shown that the initial theoretical solution to the disturbance decoupling problem does not satisfy the basic control requirements. For that reason, the solution was modified accordingly. This resulted in the **PI**-controller with the disturbance decoupling. As a result the disturbances practically did not affect the system output of the real laboratory plant.

Acknowledgements

This work was supported by the Slovak Grant Agency grants No. VG-1/0656/09 and VG-1/0369/10 and by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

References

[1] ARANDA-BRICAIRE, E., MOOG, C., POMET, J.: A linear algebraic framework for dynamic feedback lineariza-

- tion. *IEEE Transactions on Automatic Control*, 40, pp 127-132, 1995.
- [2] ARANDA-BRICAIRE, E., KOTTA, Ü., MOOG, C.: Linearization of discrete-time systems. *SIAM Journal of Control Optimization*, 34, pp 1999-2023, 1996.
- [3] CONTE, G., PERDON, A.M., MOOG, C.: The differential field associated to a general analytic nonlinear dynamical system. *IEEE Transactions on Automatic Control*, 38, pp 1120-1124, 1993.
- [4] CONTE, G., MOOG, C., PERDON, A.M.: *Algebraic Methods for Nonlinear Control Systems. Theory and Applications*. Communications and Control Engineering. Springer-Verlag, London, 2nd edition, 2007.
- [5] FLIESS, M.: A new approach to the noninteracting control problem in nonlinear systems theory. *23rd Allerton Conference*, Monticello, 1985.
- [6] FLIESS, M.: Reversible linear and nonlinear discrete-time dynamics. *IEEE Transactions on Automatic Control*, 37, pp 1144-1153, 1992.
- [7] GRIZZLE, J. W.: Controlled invariance for discrete-time nonlinear systems with an application to the disturbance decoupling problem. *IEEE Transactions on Automatic Control*, 30:868-873, 1985.
- [8] HALÁS, M., ŽILKA, V.: Noninteracting control of coupled tanks: from theory to practice. *Eurocast*, Gran Canaria, Spain, 2011.
- [9] HALÁS, M., KOTTA, Ü., LI, Z., WANG, H., YUAN, C.M.: Submersive rational difference systems and formal accessibility. *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, Seoul, Korea, 2009.
- [10] KOTTA, Ü.: *Inversion Method in the Discrete-Time Nonlinear Control Systems Synthesis Problems*. Springer, Berlin, 1995.
- [11] KOTTA, Ü., ZINOBER, A.S.I., LIU, P.: Transfer equivalence and realization of nonlinear higher order input-output difference equations. *Automatica*, 37, pp 1771-1778, 2001.
- [12] ŽILKA, V., HALÁS, M.: Disturbance decoupling of coupled tanks: from theory to practice. *IFAC Symposium on System, Structure and Control*, Ancona, Italy, 2010.
- [13] ŽILKA, V., HALÁS, M., HUBA, M.: Nonlinear controllers for a fluid tank system. *R. Moreno-Diaz, F. Pichler, A. Quesada-Arencia: Computer Aided Systems Theory, Lecture Notes in Computer Science*, Springer, 2009.
- [14] XIA, X., MARQUEZ-MARTINEZ, L.A., ZAGALAK, P., MOOG, C.: Analysis of nonlinear time-delay systems using modules over non-commutative rings. *Automatica*, 38, pp 1549-1555, 2002.

Doc. Ing. Miroslav Halás, PhD.

Slovak University of Technology
 Faculty of Electrical Engineering and IT
 Institute of Control and Industrial Informatics
 Ilkovičova 3
 812 19 Bratislava
 Slovakia
 E-mail: miroslav.halas@stuba.sk

Mathematical Modeling and Implementation of the Airship Navigation

František Jelenčíak, Ivan Masár

Abstrakt

The main source of errors for airship navigation is that the airship body is not solid. For this reason a standard fixed calibration for a navigation system is not the best solution. This article provides an overview of the proposed navigation system for airships with compensation of errors due to resilience.

Klíčové slová: navigation, airship, strapdown, Kalman filter

Introduction

Navigation systems and flight control are basic equipments of unmanned aerial systems (UAS). For modeling of the navigation system (see references) it is necessary to select appropriate coordinate systems and to select description methods for body orientation. The next step is to apply transformations between coordinate frames, and to relate them to kinematics and dynamics theory. Stability and reliability of the navigation system can be achieved by combining the on-board IMU (Inertial Measurement Unit) with external observation units such as GPS (Global Positioning System), barometric and magneto-compass units.

The proposed model of airship navigation uses coordinate systems like ECEF (Earth-Centered-Earth-Fixed) and NED (North-East-Down). For the description of the airship's orientation Euler angles as well as roll, pitch, and yaw are used. All measured data are joined together by a special type of direct Kalman filter. This filter and the strapdown modification are designed in such a way that together with other measurement units it is possible to compensate the problem, which is caused by the non-solid body of airships. Navigation systems implemented on airships show characteristic errors due to this fact. The following facts can be considered as sources of these errors: an airship body changes its shape depending on several parameters, e.g. the helium pressure in the hull; the ambient temperature; relative position of inner and outer hull; aerodynamic forces. The shape changes are in general nonlinear. For these reasons a fixed calibration for navigation system is not the optimal solution. This article provides an overview of the proposed navigation system for the airship which compensates the aforementioned errors.

The development of the airship navigation algorithm was carried out in the programming environment Matlab-Simulink. The toolboxes used are "Target Support Package" and "Code Composer Studio" was designed Simulink-scheme, which is algorithm interpretation of the airship model navigation. This Simulink-scheme can be transferred as program code into the microprocessor TMS320F28335, which together with sensors of the IMU (Inertial Measurement Unit), barometric unit, GPS, and magneto-compass are core of the navigation system (http://prt.fernuni-hagen.de/ARCHIV/2010/fernsehen_2010.html, <http://www.derwesten.de/staedte/hagen/FernUni-Luftschiff-auf-Minensuche-id2359263.html>).



Fig.1 Electronic board of the navigation system (design by Dr.-Ing. Ivan Masár).

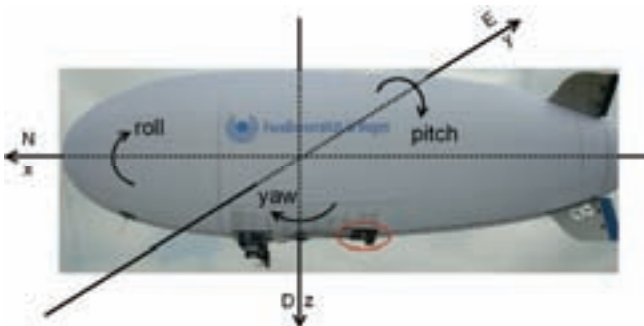


Fig.2 Airship photo with selected area for storing of the navigation system (two hulls, length 9m, diameter 2.5m, volume 27m³).

The article is structured as follows: In chapter 2, the basic scheme of the navigation algorithm is shown. In chapters 3-7, the essential parts of the navigation algorithm are described in detail. In the final chapter 8, conclusions and results are given.

2. Basic scheme

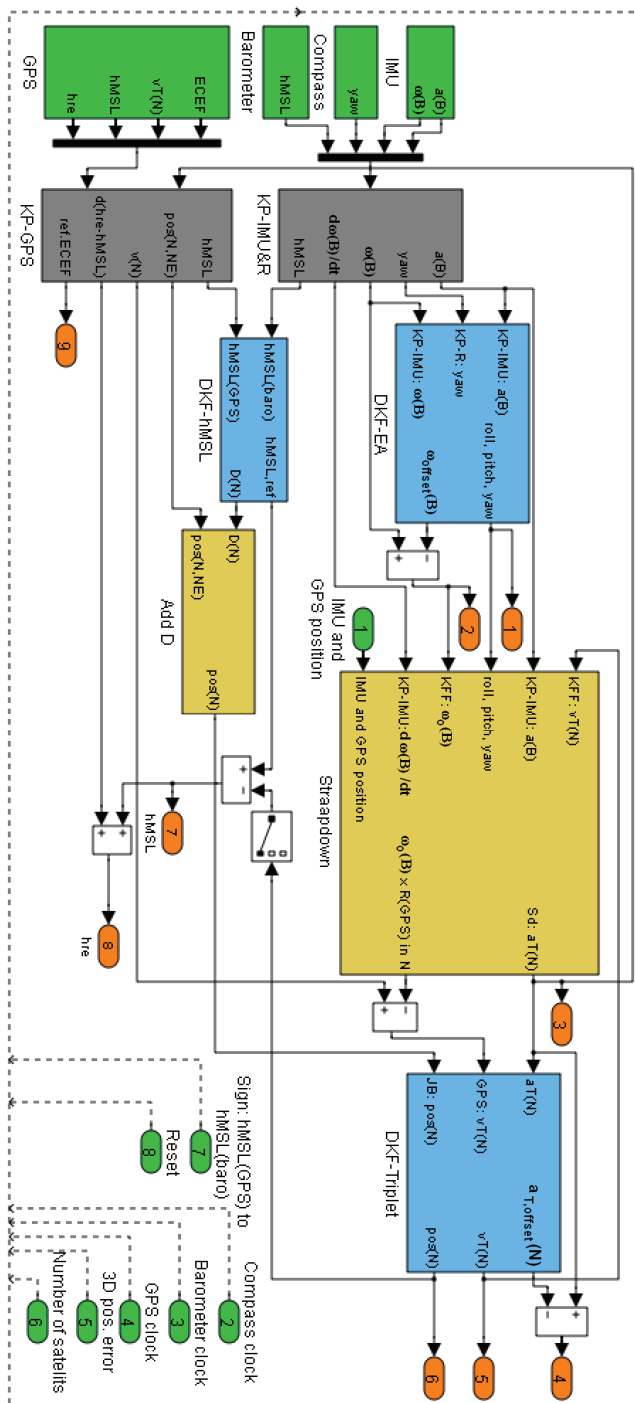


Fig. 3. Basic scheme of the navigation algorithm.

At the beginning it is necessary to divide the input navigation data into two groups. The first group represents data from the IMU. The second group represents data from other sensors.

From the IMU (strapdown platform), it is possible to load data such as acceleration and angular velocity of rotation. The advantage of these data is that they may be available at high sampling rates with relatively high sensitivity and accuracy. The disadvantage of these data is, that velocity, position or orientation determined by direct time integration are loaded with error, which in time always increases.

A second group is represented by data from GPS, barometer, and magneto-compass. The advantage of these data is, that the error with which they are loaded is not dependent

on time. The disadvantage of these data is that they often show less accuracy, sensitivity, and their availability with a smaller sampling rate. Advantages of both groups can be obtained by some form of Kalman filter.

3. Signals and Blocks

In Fig. 3, all inputs into the navigation algorithm are marked green. These are signals, which are directly necessary for the navigation algorithm. Output signals are marked in orange. Algorithms which use pre-filter based on Kalman filter are marked in gray. Algorithms which use direct Kalman filters are marked in blue. Other blocks in which partial calculations are carried out are marked with yellow or white. The argument B represents the body frame and the argument N represents the navigation frame. Both frames have orientation as NED coordination system.

4. Algorithms with Kalman pre-filter

In Fig. 3, two blocks are using Kalman pre-filters: “KP- IMU&R” and “KP-GPS”. Block “KP-IMU&R” processes two types of data: direct data from the IMU, which are represented by the body acceleration $a(B)$ and the body angular velocity $\omega(B)$, and data from the magneto-compass (angle yaw γ_R) and barometer (h_{MSL} – height above mean sea level). Task of this block is to make Kalman signal pre-filtering. By this pre-filter also the value of angular acceleration $\frac{d\omega(B)}{dt}$

Block “KP-GPS” processes data from GPS and strapdown algorithm. This block calculates the following values: h_{MSL} , position and velocity in the navigation frame N, the difference between the height of the reference ellipsoid h_{RE} and h_{MSL} , and the ECEF reference point which defines zero of the navigation frame N. These parameters are calculated using Kalman filtering. The ratio between the covariance matrix of process noise and the covariance matrix of measurement noise is dependent on the total translational acceleration of the body in the navigation frame N $S_d: a_T(N)$ determined by strapdown algorithm.

5. Algorithms with direct Kalman Filter

In Fig. 3, three blocks are applying direct Kalman filters: “DKF-EA”, “DKF-hMSL”, and “DKF-Triplet”.

Block “DKF-EA” processes two types of data:

- data from the IMU as total acceleration $a(B)$ and angular velocity of rotation $\omega(B)$;
- reference data about the azimuth γ_R from the magneto-compass.

In this block, reference values of Euler angles are computed based on the azimuth from the magneto-compass and acceleration values from the accelerometer. These reference values of Euler angles are used for estimation of the real Euler angles using a direct Kalman filter. The mathematical model of the direct Kalman filter is divided into two parts:

- direct computation of Euler angles using the RM (rotation matrix) with respect to the sequence of rotation roll, pitch, and yaw
- computation of Euler angles using quaternions.

For smaller values of Euler angles, direct computation using the rotation matrix is considered. For larger values of Euler angles, computing over quaternions is applied.

The direct Kalman filter works as follows: if reference data in the actual sample time interval are not known, the filter performs angular velocity integration, resulting in Euler angles or quaternions. If reference data in the actual sample time interval are known, the filter performs estimation and computed offset of angular velocity. This offset of angular velocity is then subtracted of the angular velocity

$$\omega_o(B) = \omega(B) - \omega_{\text{offset}}(B) \cdot$$

Block "DKF-hMSL" processes two types of data:

- height above mean sea level from barometer $h_{\text{MSL}}(\text{baro})$
- height above mean sea level from GPS $h_{\text{MSL}}(\text{GPS})$

In this case, $h_{\text{MSL}}(\text{GPS})$ is the reference signal. The reason is, that GPS can operate in several modes (for example differential GPS). The accuracy of $h_{\text{MSL}}(\text{GPS})$ is generally dependent on many factors. For this reason auxiliary variables as $3D_{\text{ERR}}$ (3D position error), N_S (number of the satellites) and $S_{\text{GPS2B}}(h_{\text{MSL}}(\text{baro}))$ value is replaced by $h_{\text{MSL}}(\text{GPS})$ value with respect to weight of both values) are used. According to the value of the logical variable S_{GPS2B} the direct Kalman filter can perform the estimation. If the estimation, then the direct Kalman filter carried out determines also the offset $h_{\text{MSL,offset}}$, which is subtracted of the $h_{\text{MSL}}(\text{baro})$. The result $d = h_{\text{MSL}}(\text{baro}) - h_{\text{MSL,offset}}$ is then subsequently subtracted from the reference $h_{\text{MSL,ref}}$. $h_{\text{MSL,ref}}$ corresponds with the ECEF reference point.

The final result from this block is then an estimated value

$$D = h_{\text{MSL,ref}} - d = h_{\text{MSL,ref}} - h_{\text{MSL}}(\text{baro}) + h_{\text{MSL,offset}} \cdot$$

Under certain assumptions, this value of D may replace the D value in navigation frame N computed by GPS. The advantage of this method is that the value of D is known from barometric measurements even if no GPS signal is available. This method divides one position vector in the navigation frame N into two independent parts: part NE and part D.

Block "DKF-Triplet" processes three types of data:

- translational acceleration $a_T(N)$ computed from strapdown algorithm
- velocity in navigation frame N $v_T(N)$ from GPS
- position in the navigation frame N obtained as combination of elements NE of the position vector N from GPS with element D from "DKF-hMSL" block

This block uses a direct Kalman filter as estimator in the following cases:

- in the current sampling period a reference vector NED or only NE of the position in frame N is known, or
- in the current sampling period a reference vector of the translational velocity from GPS $v_T(N)$ is known, or
- in the current sampling period only a reference value D as one element from position vector in navigation frame N is known

If in the current sampling period no reference value of the position in frame N as N or E or D, or $v_T(N)$ is known, the direct Kalman filter works as integrator for translational acceleration $S_d: a_T(N)$ from the strapdown algorithm. If the direct Kalman filter works as estimator, then it computes also the acceleration offset $a_{T,offset}(N)$, with

$$a_{TO}(N) = a_T(N) - a_{T,offset}(N) \cdot$$

6. Strapdown algorithm

The mathematical model of the strapdown algorithm, which is used for airship navigation purpose, expresses the following relationship

$$a(B) = \frac{d\omega(B)}{dt} \times R_{\text{IMU}} + \omega(B) \times v_T(B) + \omega(B) \times (\omega(B) \times R_{\text{IMU}}) - g(B) + a_T(B) + a_R(B)$$

where

$a(B)$ is the measured acceleration vector, $v_T(B)$ is the translational velocity vector of the body, R_{IMU} is the IMU position on the body of the turn-point, $g(B)$ is the gravitation vector in the body frame, $a_T(B)$ is the translational vector of the body, $a_R(B)$ is the residual acceleration vector of others forces in the body frame which are not include in the strapdown algorithm.

In this case in the strapdown algorithm is $a_R(B) = 0$, because errors of unmodeling acceleration are solving by $\omega_{\text{offset}}(B)$, $h_{\text{MSL,offset}}$ and $a_{T,offset}(N)$. The reason why it is possible to accept this assumption is, that all errors caused by this assumption are small in relation to errors caused by the behaviour of the airship body hull.

In case of $a_R(B) = 0$, the strapdown equation is the equation of the „Flat Earth Navigator“.

In the strapdown block there is also the computation of $(\omega_o(B) \times R(\text{GPS}))_N$, which defines the transformation to the navigation frame N with $R(\text{GPS})$ as GPS antenna position on the airship body of the airship turn-point. This part is necessary to subtract of the $v_T(N)$ obtained from GPS receiver.

Main output from the strapdown algorithm is the translational acceleration in the navigation frame N: $a_T(N)$.

7. Other blocks

In the 'other' blocks of the algorithm (see figure 3), auxiliary inputs and characteristic outputs of the navigation algorithm are calculated.

The clocks for magneto-compass, barometer, and GPS are used as auxiliary inputs. The logical value of these signals is ONE if magneto-compass, barometer, or GPS have an actual value in the actual sample time. This is important for decision of the Kalman filtering.

The following parameters are outputs from the navigation algorithm:

- roll, pitch, and yaw angles
- the angular velocity $\omega_o(B)$ or $\omega_o(N)$, adjusted by the angular velocity offset $\omega_{\text{offset}}(B)$

- the translational acceleration from the strapdown algorithm $S_d: \mathbf{a}_T(N)$
- the translational acceleration $\mathbf{a}_T(N)$ which is result with respect to all measurement units and which is adjusted by offset $\mathbf{a}_{T,offset}(N)$
- the translational velocity $\mathbf{v}_T(N)$;
- the position $\mathbf{pos}(N)$;
- the height above mean sea level h_{MSL} ;
- the height above the reference ellipsoid h_{RE} ;
- the ECEF reference point

Conclusions and results

To show the results, a comparison of a commercial navigation system with a system that was developed at our department is accomplished. The latter is used as one part this algorithm (the full mathematical model of the navigation algorithm can not be presented here due to limited space). For this comparison we used a commercial navigation system in the price range of 4000 €. For a comparison, 25 journeys by car were made. The mounting of both navigation systems were done in a way that the conditions were as similar as possible as on an airship hull (see figure 4).

Both navigation systems are mounted on a special fixture which is placed in the trunk of a car. This fixture consists of a top platform, a lower platform, and four foam springs. The navigation systems are mounted on the top platform. The lower platform is based in the trunk. When the car is running, the springs cause movements of the top platform contrary to the movement of the car. This is the main objective of the fixture. This mechanism emulates an airship with implemented navigation system. The airship hull is a non-solid body with spring effects. Another reason why the comparison was done on the car is, that at the time of article writing our navigation system did not offer a DGPS mode. Driving a car and matching with Google maps is an alternative strategy.

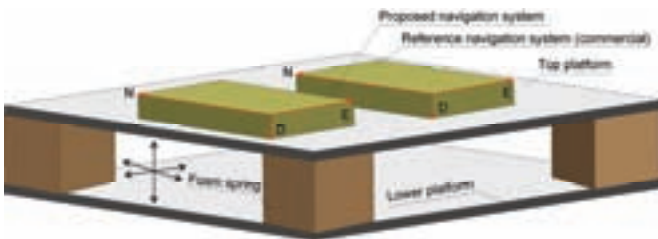


Fig. 4. Fixture of the navigation systems in the car.

Figures 5 and 6 show Google maps with plotted averaged paths (from 25 journeys by car) obtained from the navigation system. The real paths are driven with respect to the traffic rules. In figures 5 and 6, two characteristic points are marked. Point 1 is located at traffic lights. Point 2 is a parking position. From figures 5 and 6, it is easy to see that if the car stopped the navigation algorithm begins "walking". This is caused by the GPS data, which are obtained in normal user mode. If the body is in motion, it is possible to eliminate this "walking" of the GPS by taking into account the orientation of the body (car).

The precision of the proposed navigation system on the defined path is better than the commercial navigation system (4000 €) about from 0,5 to 2,5m.

Literatúra

- [1] Britting, K.R.: Inertial Navigation System Analysis. Wiley Interscience, New York, 1971.
- [2] Zhu, J.: Conversion of Earth-centered Earth-fixed coordinates to geodetic coordinates. In: Aerospace and Electronic Systems, IEEE Transactions. Vol. 30. pp. 957-961, 1994.
- [3] Brant, A., Gardner, J.F.: Constrained navigation algorithms for strapdown inertial navigation systems with reduced set of sensors. In: American Control Conference. Vol. 3. pp. 1848-1852, 1998.
- [4] Farrell, J., Barth, M.: The Global Positioning System and Inertial Navigation. 1 edition. McGraw-Hill Professional, 1998.
- [5] Bose, C., Sam, C.: GPS/INS Integrated Navigation Systems. Technalytics Inc. Cangoa Park Kalifornien, USA, 2000.
- [6] Titterton, D.: Strapdown inertial navigation technology. In: The American Institute of Aeronautics and Astronautics and the Institution of Electrical Engineers, 2000.
- [7] Barbour, N., Schmidt, G.: Inertial Sensor Technology Trends. In: IEEE Sensors Journal. Vol. 1. pp. 332-339, 2001.
- [8] Dissanayake, G., Sukkarieh, S., Nebot, E., Durrant – Whyte, H.: The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. In: IEEE Transactions on Robotics and Automation. 17(5). pp. 731-747, 2001.
- [9] Foppe, K.: Kombination von inertialen und satellitengestützten Beobachtungsverfahren zur ingenieurgeodätischen Überwachung von Brückenbauwerken. In: Wissenschaftliche Arbeiten der Fachrichtung Vermessungswegen der Universität Hannover. Nr. 242, Hannover, (2001).
- [10] Mohinder, S.G., Weill, R.L., Andrews, A.P.: Global Positioning Systems, Inertial Navigation and Integration. John Wiley and Sons, 2001.
- [11] Woodman, O.O.J.: An introduction to inertial navigation. Technical Report 696. University of Cambridge, 2007.

Abstract

The main source of errors for airship navigation is that the airship body is not solid. For this reason a standard fixed calibration for a navigation system is not the best solution. This article provides an overview of the proposed navigation system for airships with compensation of errors due to resilience.

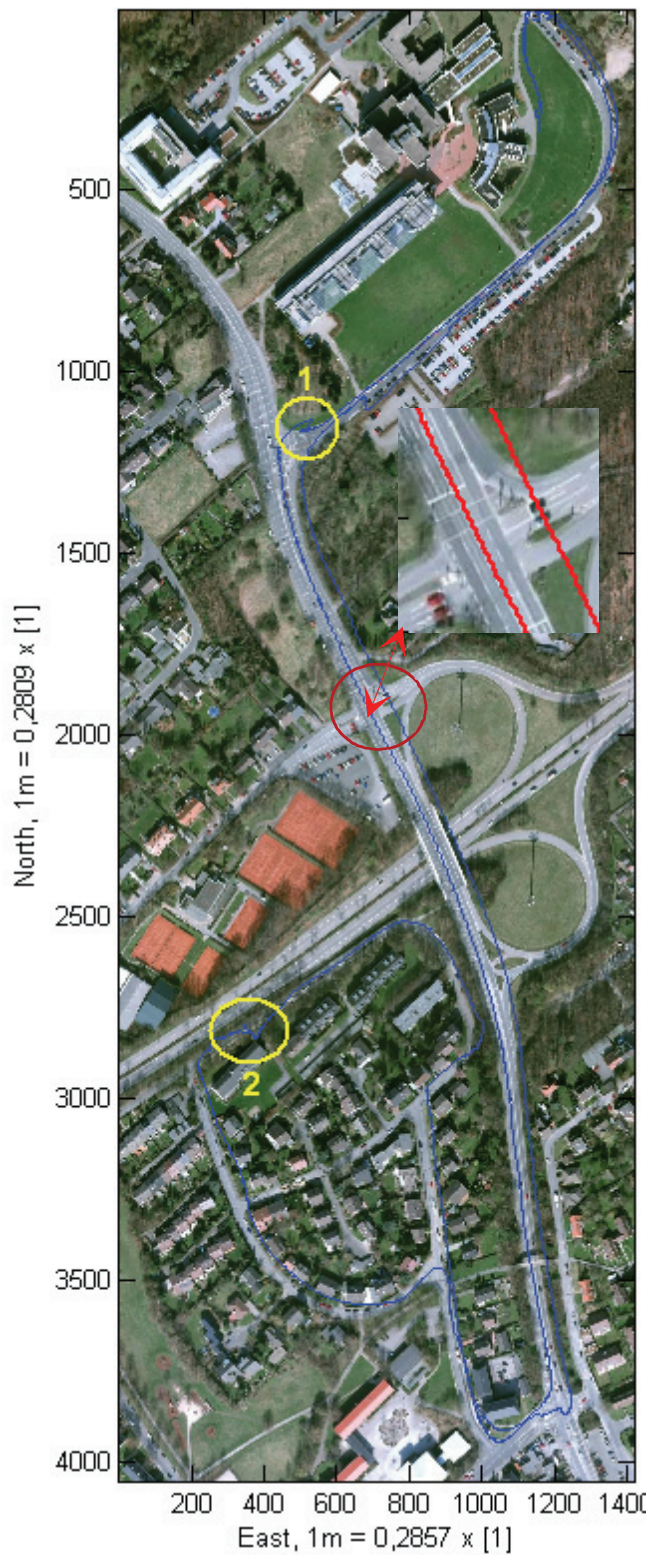


Fig. 5. Reference navigation system (commercial).

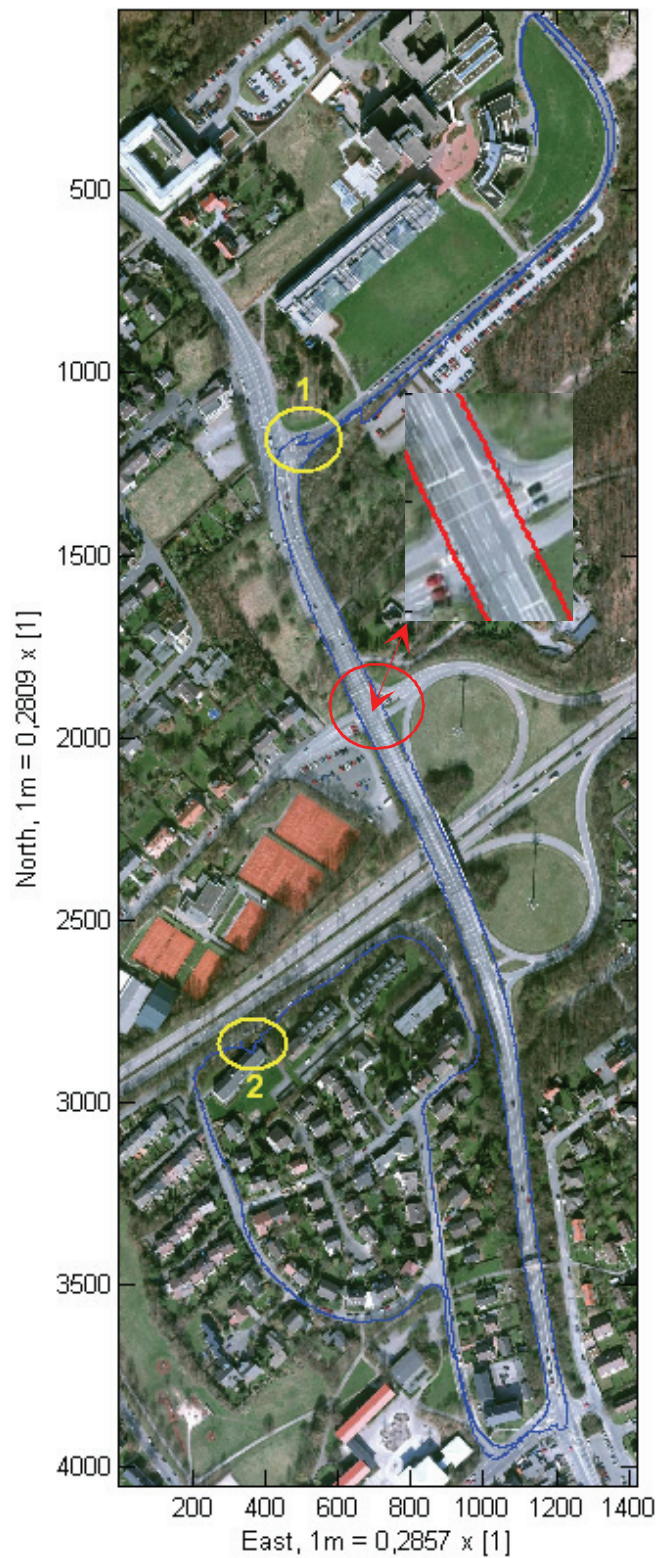


Fig. 6. Proposed navigation system.

Ing. František Jelenčík, Dr.-Ing. Ivan Masár

FernUniversität in Hagen

MI / PRT

Universitätstrasse 27

58097 Hagen

Tel.: + 49 23 31 987 111 9

E-mail: Frantisek.Jelenciak@fernuni-hagen.de

Remote experiment in control education

Martin Kalúz, Ľuboš Čírka and Miroslav Fikar

Abstract

This paper describes a software solution used for realization of remote experiment. The main aim of this work was to develop the computer software for remote access and control of thermo-optical device uDAQ28/LT. Our solution connects several information technologies which operate on client and server side. For client side application development, we choose the software programming platform Adobe Flash and for server side solution we choose technologies PHP, MySQL, and MATLAB.

Keywords: remote, control, laboratory, experiment

Introduction

Today, in a field of automation and process control education, it is necessary to create quality conditions, also by enabling students to use real experimental devices, on which they can carry out their experiments and improving their skills. This aim can be fulfilled by several different ways. One of the most used solutions is real technological laboratory. However, from the students view, it can be used only during school opening hours and it depends on accessibility of teachers. But there is another way how to provide students access to real experiments. It is remote laboratory that can be used for control of real technological plants from any location with access to Internet.

There are some solutions of remote laboratories over the Internet, created by software developers from other universities.

Interesting solution comes from FEI STU in Bratislava, where remote control software for the same thermo-optical device uDAQ28/LT has been developed. This solution is based on Java Applet and Java Servlet communication, and MATLAB used as main control software. Used technology is described in detail in paper by Bisták and Beránek [1]. This solution was later extended by new reservation system [2] and has been applied to another type of device (hydraulic plant) [3].

A DSP (Digital Signal Processor)-based remote control laboratory was proposed by Hercog et al [4]. Their solution is based on in-house developed control hardware with MATLAB as control software and client application created in LabVIEW graphical programming environment. It is used for control of real systems like electric motor, robot mechanism, power converter, etc.

Another remote control laboratory was created at Department of Industrial Systems Engineering of University Miguel Hernández in Elche, Alicante, Spain. Puerto et al. [5] suggested remote control of two mechanical devices, a DC motor and sliding cylinder. The client side of software is built on HTML/PHP technologies. The communication part of software is based on Common Gateway Interface (CGI) and the main control software is MATLAB.

We develop a remote laboratory that can be accessed by Internet, using a web browser, but propose a different combination of technologies than above mentioned. Our remote

control application is created as Adobe Flash program, framed in regular web page, accessible through e-learning system Moodle [6]. It also provides a video stream from remote web camera that is aimed at controlled device, so the remote user can observe the behavior of the experiment in real time. The Flash application is used on the client side. The server side with the real controlled device uDAQ28/LT consists of PHP/MySQL Apache-type server containing programs serving communication between every part of our software solution, MATLAB that directly controls remote device, and Java database software drivers for providing access to MySQL from MATLAB.

In comparison with client applications from Bisták et al., and Puerto et al., based on HTML/PHP and Java Applet, the Adobe Flash programming platform is more suitable for graphical design development and also it provides more options for graphical user interface (GUI) customization. From the simplicity point of view and software/hardware requirements, Java and HTML/PHP solutions seems to be better, but not so far. Hercog et al. used the LabVIEW environment for building GUI, which is directly designed for such use, but it is questionable if LabVIEW is an appropriate environment for building features like interface for communication between users, dynamically generated forms and fully animatable GUI. The main advantage of our solution is the possibility of direct storage and data management from experiments. This advantage stems from the using MySQL database system, also as communication channel. Another difference is that we use technology based on asynchronous communication, so we can prevent the crash of experiment due to short-time losses of connection between client and server. MySQL, PHP layer, and MATLAB are located on the same physical server, therefore in the case of link outage between client PC and server, the local communication channels on server side stay unaffected by this issue, so the measurement remains running and data are collected in MySQL. In this case, user does not lose the measured data. The experiment is turned off only in case of longer-time link outage. Using PHP is fast and efficient way to secure communication between Flash application on client side and MySQL.

This is not our first attempt of using Adobe Flash platform in software development. Recently, we created a virtual laboratory based on Flash [7], and MATLAB computation online laboratory based on Java [8].

uDAQ28/LT device

For purposes of remote laboratory and application testing, we chose a thermo-optical device uDAQ28/LT (Fig. 1) created by DigiCon Corporation [9]. It is originally designed for education in fields of automation and process control. Device can be connected directly to computer by universal serial bus (USB) and after simple driver installation it can communicate with MATLAB. Manufacturer provides with device the installation CD including software drivers, Simulink schemes, and user manual. Plant can be controlled by 3 analog inputs (bulb voltage, fan voltage, and LED voltage), and has 4 primary outputs (light intensity, temperature inside light tube, fan rpm, and voltage taken by fan).



Fig.1 Thermo-optical device uDAQ28/LT

Software solutions

Our software solution combines several technologies, which were chosen depending on selected criteria. The main criteria were data storing ability, connection robustness, program processing rate, easy-to-use graphical user interface, and low system and hardware requirements. In this section, we

will explain the realization of links between every part of chosen technology and its principles. Every user instruction from Flash application is processed on server side by PHP scripts.

PHP is used in three different ways.

- The first is a group of scripts, which serves application timing and simple computations with results directly sent back to application.
- The second group of PHP scripts is used to serving connection between Flash application and MySQL database system.
- The third group of scripts is used for executing of server system commands.

We use MySQL system for storing data like device states, running experiment states, MATLAB states, connection states, user accounts, and other. The uDAQ28/LT device on server-side is directly controlled by MATLAB/Simulink. We have created a Simulink block function, which sends the measured data to database in every sample time period of running experiment. These data are collected by Flash application and sent to user. The connection between MATLAB and MySQL is performed by Java database connection driver (JDBC).

The communication between every part is shown in Fig. 2. Technology of experiment execution and data collecting is as follows. MATLAB is executed by PHP script that sends execution command directly to operating system command line. This command is built in client-side application and consists of two parts. The first is system command for MATLAB execution with program setup parameters, and the second is command for MATLAB command line. This part contains all instructions for experiment execution, as names of Simulink schemes and m-files to run, input parameters, and other instructions for MATLAB. When MATLAB/Simulink program is running, the specific m-file script writes to MySQL information about its state (1 – MATLAB is running). Data collecting is based on asynchronous technology. Every new measured state from device is sent by Simulink to MySQL and periodically collected by client application.

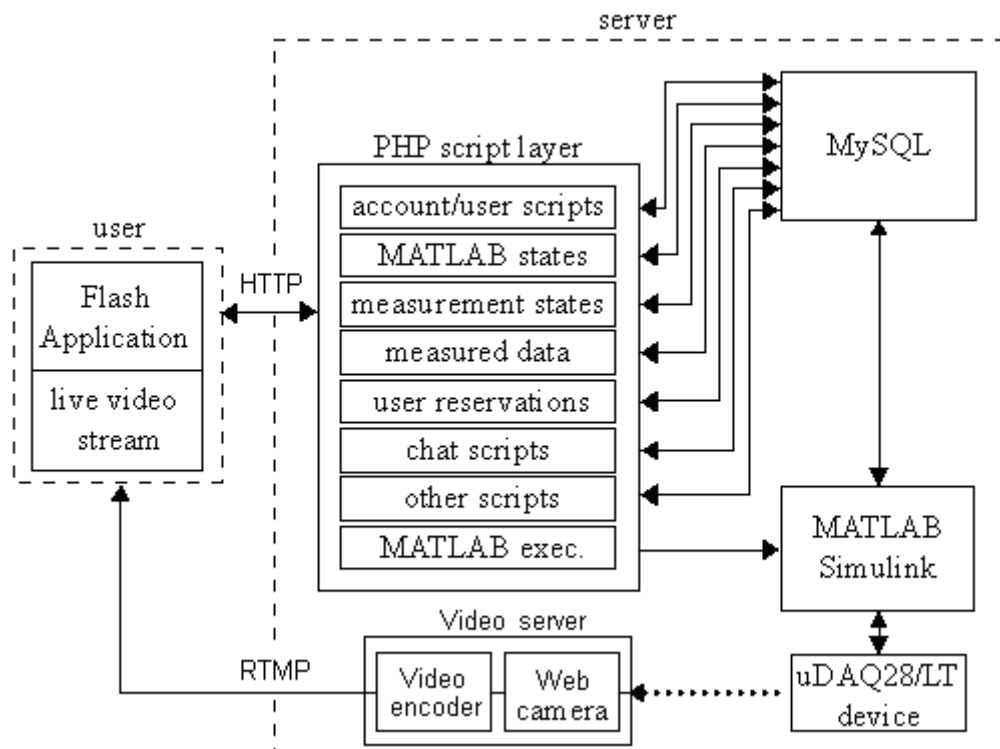


Fig.2 Scheme showing connections between client and server side of software solution and controlled device

Sampling period for Simulink experiment is set up by user and for data collection by Flash application is set to 0.2 seconds. MATLAB uses JDBC for database connection and Flash application connects to database through PHP layer. After experiment is finished, MATLAB is automatically turned off by its script. It can be turned off also manually by the user from client application.

Application overview

The remote laboratory can be used by anyone who is registered as user of Moodle. After sign in to remote laboratory course, user fills a simple registration form to create a new account. After registration, user receives a confirmation e-mail with account activation link and can sign in to remote control application.

The Flash application is accessible through any web browser, because it is embedded into regular web page. The only requirement to run the application is Adobe Flash Player (downloadable software plug-in). The application web site is located on e-learning system LMS Moodle, and it is accessible through the main web site of our department.

Application GUI consists of several screens. The first is login interface (Fig. 3), where user has to put his login name and password to sign up for experiment.

In the actual version of our solution, it is necessary to use account login option directly in client application, because the actual version of application is aimed to be independent on web location. Due to fact that our solution is a part of remote laboratory course, located on Moodle, in future we plan to create extension that will provide Access to application user account directly through Moodle, by using its session cookies.

On the second screen (Fig. 4) user can reserve the time session for experiment and communicate with other signed up users by chat. Application provides option for reservations management, like creation, removal, and sort.

Before proceeding to the experiment itself a new time reservation has to be created. Flash program scripts run in background of application to compare new reservation with those which are saved in database to avoid time collision between

experiments. One user can register maximum three reservations with maximal duration of 30 minutes (sufficient time for performing the experiment on uDAQ28/LT).

If session reservation is finished, user can switch to another screen, where he can set up the experiment. On setup screen (Fig. 5) user chooses Simulink scheme which will be executed in remote MATLAB. All Simulink schemes are located at server and registered in MySQL database system. Content of each scheme is automatically detected by parsing software. This software part is necessary for gathering informations about Simulink model files (MDL-files) and their content. In application, when user selects scheme for measurement, the parsing PHP script reads the MDL file and detect all important objects in it. The script looks for Simulink blocks and their parameters, and sends information about them back to Flash application, where the input form is dynamically generated. This way, user can view and edit all important parameters that model requires. The script detects most commonly used blocks and their parameters, like constant, step, transfer function, and PID controller blocks. User can change their values before measurement starts and also select parameters that may be changed during experiment.

Application contains all default simulation schemes provided by device manufacturer, but there is also an option to upload custom scheme created by user. For this purpose we have created upload form that provides all necessary features important for keeping software security and operability.

Before a new schema is uploaded to server and saved to database, the security PHP script checks if file has proper extension, size, and if contains all necessary parts, like database connector block and block for data export. In last step PHP script reads file to look for dangerous content (for example system commands written by user).

When all input parameters for experiment are correctly filled, user can execute the experiment. After connection between user-side application and MATLAB is established and MATLAB/Simulink is properly running, the application switches to experiment screen (Figs. 6 and 8). User can observe experiment results and remote device states and also can send commands for setting up new values of device inputs (Fig. 7).



Fig.3 First screen of Flash application, showing the picture of remote device and login form



Fig.4 Second screen of Flash application, showing reservation system for experiment

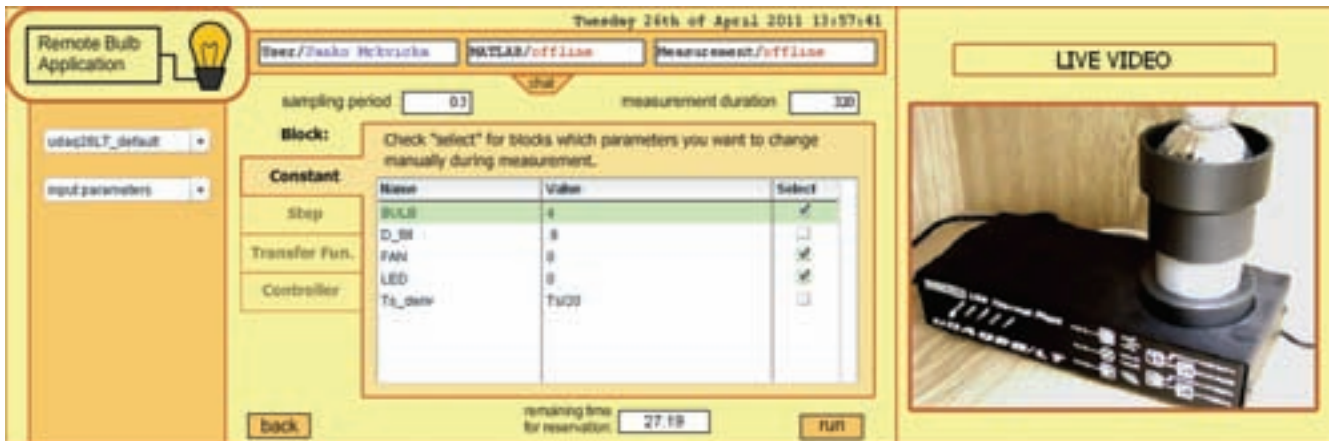


Fig.5 Input form for experiment parameters associated to chosen scheme



Fig.6 Running experiment (input/output view)



Fig.7 Running experiment (input form where user can change parameters in real time)

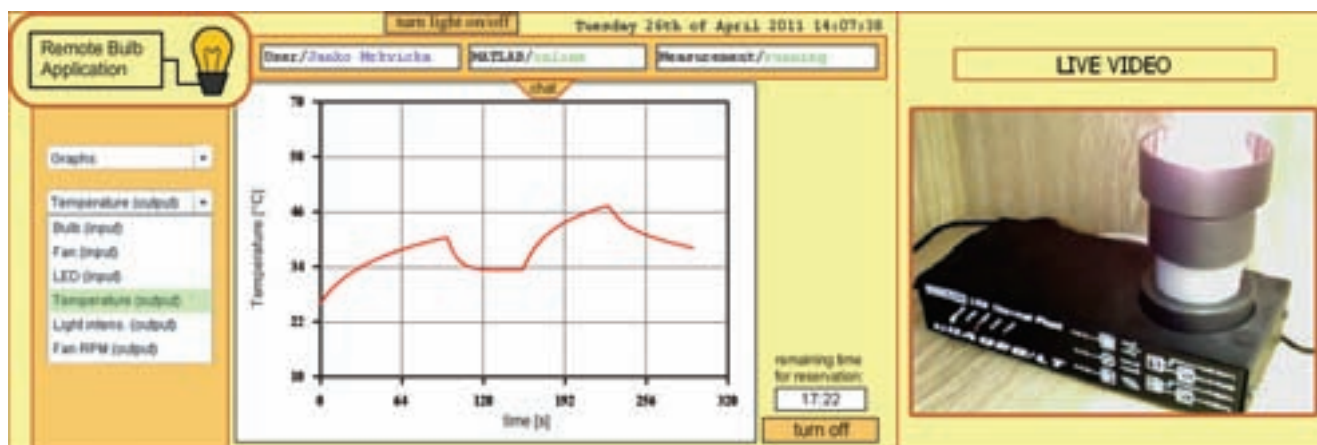


Fig.8 Running experiment (selected variable graph view)

When experiment is finished, user can download all measured data from database in chosen format. Application can provide data in the form of structured XML file, plain text file or MATLAB m-file. Every experiment result is stored in database with unique identifier, and can be accessed later. Web module for data export is shown in Fig. 9.

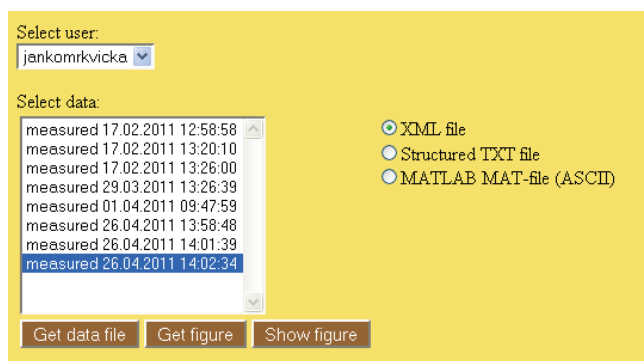


Fig.9 Data export form. Results are sorted by user name and time of measurement

Conclusion

Remote laboratories can be a suitable way to improve education in field of automation engineering and process control. Our proposed solution presents one of many different ways how can remote laboratory be realized. We have chosen technologies which can easily handle features that are required for this kind of solution. MySQL database system is suitable for fast storage and data management and it is also robust enough for programs with high database access traffic. Adobe Flash Application on user side provides easy-to-use GUI and can be run directly through web browser without any installation procedure.

Acknowledgments

The authors are pleased to acknowledge the financial support of the Cultural and Educational Grant Agency KEGA of the Slovak Republic under grant No. 3/7245/09 and of the Scientific Grant Agency of the Slovak Republic under the grant 1/0071/09.

The paper is supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

References

- [1] BISTÁK, P., BERÁNEK, M.: Java client server applications for remote laboratory based on matlab and com. In *7th Int. Conference Virtual University VU06*, 135–139. Bratislava, Slovakia, 2006
- [2] BISTÁK, P., ČIRKA, Ľ.: Reservation of remote laboratory using Moodle LMS. In *Proceedings of 10th International Conference Virtual University*. FEI STU in Bratislava, E-academia Slovaca, Bratislava, Slovakia, 2009.
- [3] ŽILKA, V., BISTÁK, P., KURČÍK, P.: Hydraulic plant remote laboratory. *International Journal of Online Engineering*, 4, 69–73, 2008.
- [4] HERCOG, D., GERGIČ, B., URAN, S., JEZERNIK, K.: A DSP-based remote control laboratory. *IEEE Transactions on Industrial Electronics*, 54, 3057–3068, 2007.
- [5] PUERTO, R., JIMÉNEZ, L. M., REINOSO, O.: Remote control laboratory via internet using matlab and simulink. In *Computer Applications in Engineering Education*, volume 18, 694–702, 2010.
- [6] DOUGIAMAS, M.: Moodle – a free, open source course management system for online learning. <http://moodle.org>, 2005.
- [7] ČIRKA, Ľ., KALÚZ, M., KVASNICA, M., FIKAR, M.: Virtual laboratory. In *Proceedings of the 9th International Scientific-Technical Conference Process Control 2010, C029a_1–C029a_8*. University of Pardubice, Kouty nad Desnou, Czech Republic, 2010.
- [8] KALÚZ, M., ČIRKA, Ľ., FIKAR, M.: MATLAB builder JA in control engineering education at FCFT STU. In *Technical Computing Bratislava 2010*, volume 18, 1–5, 2010.
- [9] HUBA, M., KURČÍK, P., KAMENSKÝ, M.: Thermo-optical device uDAQ28/LT. STU Bratislava, Illkovičova 3, Bratislava, Slovakia, 2006.

Martin Kalúz, Ing.

Ľuboš Čirka, Ing., PhD.

Miroslav Fikar, Prof. Ing., DrSc.

Slovak University of Technology
Faculty of Chemical and Food Technology
Institute of Information Engineering,
Automation and Mathematics
Radlinského 9, 812 37 Bratislava
E-mail: martin.kaluz@stuba.sk

Air/Fuel Ratio Model Predictive Control of a Real-world Gasoline Engine

Matúš Kopačka, Peter Šimončič, Jozef Csambál, Marek Honek, Sławomir Wojnar, Tomáš Polóni, Boris Rohal-Ilkiv

Abstrakt

The following paper deals with the air/fuel ratio (AFR) controller applied on spark ignition engine. It utilizes the analytical model predictive controller based on the multi-model approach which employs the autoregressive model (ARX) network and the weighting of local models, coming from the sugeno-type fuzzy logic. The weighted ARX models are identified in the particular working points and are creating a global engine model, covering its nonlinearity. Awaited improvement of a proper air/fuel mixture combusted in a cylinder is mostly gained in the transient working regimes of an engine. In these regimes, the traditional control approach loses its quality, compared to steady state working regimes of an engine. This leads to higher fuel consumption and level of emissions from an engine. Real-world experiments related to the VW Polo 1390cm³ engine, at which the original electronic control unit (ECU) has been replaced by a dSpace system executing the model predictive controller. Results acquired during the researches proves, that the proposed controller is suitable for the air/fuel ratio control giving sufficiently good and steady system output.

Keywords: model predictive control, analytical solution, air/fuel ratio, SI engine, ARX models

Introduction

A run of a spark ignition engine (SI) is highly dependent on the mixture of the sucked air and injected fuel present in the cylinder, waiting to be ignited by the spark. Incorrect ratio of this two components may lead to the poor engine power, ineffective functionality of the catalytic converter resulting in higher level of emissions polluting the environment and in the extreme case this can lead to the engine stoppage. Due to this reason it is crucial to keep the air/fuel ratio (AFR) at the stoichiometric level, which means, that both, the air and the fuel are completely combusted. Due to above mentioned reasons and all the time tightening emission standards the car producers are improving the control of the air/fuel ratio. Traditional control of air/fuel ratio is based on a feed-forward control using predefined tables determining how much fuel has to be injected into a cylinder, based on the information from the mass air flow meter. This fuel amount is subsequently corrected using the information from the lambda probe, so the stoichiometric mixture can be reached. Due to a lambda probe position (at the engine exhaust) a delay arises, causing an improper feedback correction at the unstable engine regimes, like acceleration, or deceleration. On the other side, this kind of control guarantees stability and robustness at all conditions and therefore is still preferred by car producers, despite its disadvantages in control. The academic field have started to publish other kinds of air/fuel control, mostly model-based ones. The model-based approaches are bringing good quality of control, but are also more sensitive to the model precision and issues with stability and robustness appear. A survey through popular "mean value engine modeling" is described in Bengtsson et al. (2007). This analytical way of engine modeling is very clear, but

requires exact knowledge of the system and the model error has to be taken into account explicitly. Other ways of a model acquisition are based on the experimental identification (black box modeling). Works of Zhai et al. (2010), Zhai and Yu (2009) and Hou (2007) are specialized in employment of neural networks, while Mao et al. (2009) uses for engine modeling CARIMA models. In the engine control itself became popular fuzzy logic (Hou (2007)), neural network control (Arsie et al. (2008)) and model predictive control (MPC) approaches (Loriniet al. (2006) and Muske and Jones (2006)). General topics on an issue of stability and robustness in MPC can be found in Mayne et al. (2000), or Zeman and Rohal-Ilkiv (2003). Our approach, introduced in Poloni et al. (2007) is utilizing an analytical model predictive controller with a penalization of a terminal state. It uses a multi-model approach using a weighted net (sugeno-type fuzzy logic) of autoregressive models (ARX) as a system model. The ARX models were identified in the particular working points of the engine as black box models. This method of engine modeling offers an easy way of "global nonlinear system model" acquisition with subsequent utilization in the model based system control. The preliminary real-time predictive control results presented in this paper indicate that the proposed controller could be suitable alternative toward the air/fuel ratio control through the look-up tables.

1. Air/fuel ratio

The model of the air/fuel ratio dynamics λ of a spark ignition engine is based on the mixture, defined as a mass ratio of the air and fuel present in a cylinder at a time instance k .

Due to the fact, that the air mass flow is measured as an absolute value, it was necessary to integrate this amount during the particular time and express the air and fuel quantity as relative mass densities $\left(\frac{\text{grams/cylinder}}{\text{grams/cylinder}}\right)$. Hence, the air/fuel ratio is defined, as:

$$\lambda(k) = \frac{m_a(k) \cdot 1}{m_f(k) L_{th}} \quad (1)$$

Where $m_a(k)$ and $m_f(k)$ are relative mass amounts of air and fuel in a cylinder and $L_{th} = 14.64$ is the theoretical amount of air necessary for the ideal combustion of a unit amount of fuel. The L_{th} constant normalizes the ideal value of λ to be 1.0.

2. Si engine modeling using ARX models

The engine modeling is based on the weighted linear local model with single input single output (SISO) structure (Poloni et al., 2008). The parameters of local linear ARX models with weighted validity (Murray-Smith and Johanssen, 1997) are identified to model the nonlinear dynamics of the AFR. The principle of this nonlinear modeling technique is in partitioning of the engine's working range into smaller working points.

A net of local ARX models weighted for a particular working point ϕ is defined, as:

$$\sum_{h=1}^{n_M} \rho_h(\phi(k)) A_h(q) y(k) = \sum_{h=1}^{n_M} \rho_h(\phi(k)) B_h(q) u(k) + \sum_{h=1}^{n_M} \rho_h(\phi(k)) c_h + e(k) \quad (2)$$

defined by polynomials A_h and B_h :

$$A_h(q) = 1 + a_{h,1}q^{-1} + \dots + a_{h,n_y}q^{-n_y} \quad (3)$$

$$B_h(q) = b_{h,1+d_h}q^{-1-d_h} + \dots + b_{h,n_u+d_h}q^{-n_u-d_h}$$

where symbolics q^{-1} denotes a sample delay, e.x. $q^{-1}y(k) = y(k-1)$, $a_{h,i}$ and $b_{h,(j+d_h)}$ are parameters of h^{th} local function and d_h is its delay. Parameter n_M represents the number of local models.

The ρ_h denotes a weighting function of a particular ARX model (see Sec. 2.1) and the $e(k)$ is a stochastic term with a white noise properties. The engine working point itself is defined by engine revolutions n_{en} and the throttle valve position t_r , hence: $\phi(k) = [n_{en}(k), t_r(k)]^T$. The absolute term \hat{c}_h of the equation is computed from the steady state values of the system output $y_{e,h}$ and the system input $u_{e,h}$, as:

$$\hat{c}_h = y_{e,h} + y_{e,h} \sum_{i=1}^{n_y} \hat{a}_{h,i} - u_{e,h} \sum_{j=1}^{n_u} \hat{b}_{h,j} \quad (4)$$

The model output is computed from the equation:

$$y_s(k) = \sum_{h=1}^{n_M} \rho_h(\phi(k)) * \left(\sum_{i=1}^{n_y} \hat{a}_{h,i} q^{-i} y_s(k) - \sum_{j=1}^{n_u} \hat{b}_{h,(j+d_h)} q^{-j-d_h} u(k) + \hat{c}_h \right) \quad (5)$$

which after the introduction of the estimated parameter vector $\hat{\theta}_h$ and the regression vector $\gamma(k)$, becomes:

$$y_s(k) = \gamma^T(k) \sum_{h=1}^{n_M} \rho_h(\phi(k)) \hat{\theta}_h + \sum_{h=1}^{n_M} \rho_h(\phi(k)) \hat{c}_h \quad (6)$$

2.1. Weighting functions

The full working range of the engine has been covered by a discrete amount of local linear models (LLMs), identified at particular working points. The LLMs are being weighted by a weighting functions defining validity of each local model according to an instantaneous working point of the engine. Due to a request of a smooth and continuous global engine

model, design of those weighting functions was crucial. There were designed particular interpolation functions for every LLM, assigning it 100% validity exactly at the belonging working point with a decreasing tendency in the directions of the deviation of the throttle valve opening Δt_r and the engine revolutions Δn_{en} from the particular working point. The "three dimensional" Gaussian functions:

$$\tilde{\rho}_h(\phi(k)) = \exp \left[-[\Delta n_{en}(k) \Delta t_r(k)] \begin{bmatrix} \frac{1}{\sigma_{h,1}^2} & 0 \\ 0 & \frac{1}{\sigma_{h,20}^2} \end{bmatrix} \begin{bmatrix} \Delta n_{en}(k) \\ \Delta t_r(k) \end{bmatrix} \right] \quad (7)$$

were used as the local weighting functions, due to their suitable shape fulfilling the approximation properties. The choice of tuning parameters $\sigma_{h,1} = 250$ and $\sigma_{h,2} = 0.8$ used in the weighting functions has been chosen experimentally, awaiting continuous and smooth output of the modeled system. At the same time the experiments have shown, that there can be used identical weighting functions for weighting of the air and fuel path parameters. All the weighting functions were at the end normalized by creating normalized weighting functions:

$$\rho_h(\phi(k)) = \frac{\tilde{\rho}_h(\phi(k))}{\sum_{h=1}^{n_M} \tilde{\rho}_h(\phi(k))} \quad (8)$$

so the sum of values of all weighting functions belonging to a particular working point (Fig. 1), equals exactly one: $\sum_{h=1}^{n_M} \rho_h(\phi(k)) = 1$

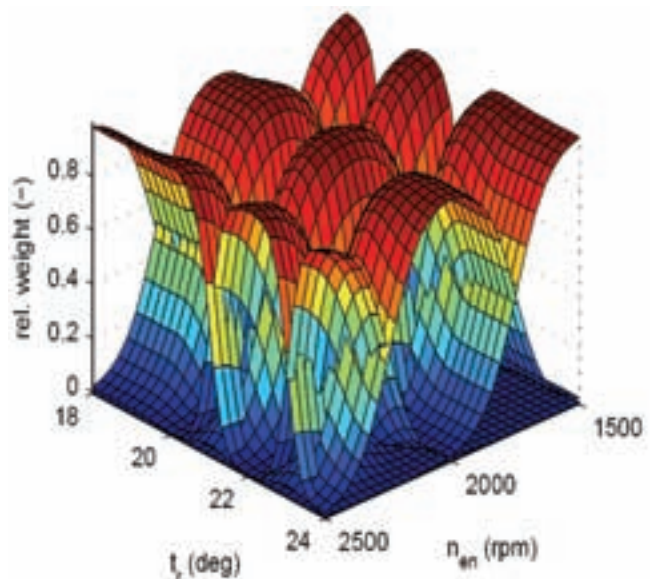


Fig.1 Relative weighting Gaussian functions

2.2. Model identification

Considering the $\lambda(k)$ modeling, the engine has been divided into two subsystems with independent inputs, namely into:

- Air path- with the throttle valve position as the disturbance input
- Fuel path- with the input of fuel injector opening time

Another disturbance like acting quantity in the air path were engine revolutions, implicitly included in the engine model, particularly for each working point.

Parameters of the local ARX models have been estimated from the data acquired from the exhaust gas oxygen sensor and an air flow sensor. The identification has been designed so, that the dynamics of the air path and fuel path stayed uncoupled, hence the dynamics of both paths were measured indirectly.

2.2.1. Air path identification

The first experiment started at the stoichiometric value of λ_a in the operation point ϕ . To excite the air path dynamics, the throttle valve position was oscillating around its steady position according to a pseudo random binary signal (PRBS), while the fuel injectors were delivering constant fuel mass $m_{f,e}$. The change in λ_a value has been recorded. During the experiment the engine was braked at constant revolutions.

2.2.2. Fuel path identification

The identification of the fuel path dynamics has been done similarly, but with the fixed throttle valve delivering a constant air mass $m_{a,e}$. The PRBS was varying the fuel injectors' opening time and the value of λ_f had been measured again.

In both experiments it was necessary to wisely propose a PRBS, so that the air/fuel mixture is always ignitable. The local ARX models can be subsequently determined from the measured values of instantaneous $\lambda_a(k)$ and $\lambda_f(k)$ belonging to the air path and fuel path, utilizing relative air and fuel mass densities:

$$m_a(k) = m_{a,e}(\phi)\lambda_a(k) \quad (9)$$

and

$$m_f(k) = \frac{m_{f,e}(\phi)}{\lambda_f(k)} \quad (10)$$

The final formula describing the air/fuel ratio dynamics is built up of local linear ARX models of the air and fuel paths is in the form:

$$\lambda_s(k) = \frac{1}{L_{th}} * \left[\frac{\gamma_a^T(k) \sum_{h=1}^{n_A} \rho_{a,h}(\phi(k)) \hat{\theta}_{a,h} + \sum_{h=1}^{n_A} \rho_{a,h}(\phi(k)) \hat{c}_{a,h}}{\gamma_f^T(k) \sum_{h=1}^{n_F} \rho_{f,h}(\phi(k)) \hat{\theta}_{f,h} + \sum_{h=1}^{n_F} \rho_{f,h}(\phi(k)) \hat{c}_{f,h}} \right] \quad (11)$$

Where:

- γ is the regression vector of system inputs and outputs
- n_A is the amount of working points
- ρ is the interpolation function
- ϕ is the vector of a working point
- θ is the vector of ARX parameters
- c is the absolute term of an ARX model

In accordance with the general model structure presented, the key variables are defined in the Table 1.

general symbol	air-path model	fuel-path model	operating point
$y(k)$	$m_a(k)$	$m_f(k)$	
$u(k)$	$t_r(k)$	$u_f(k)$	
$\gamma(k)$	$\gamma_a(k)$	$\gamma_f(k)$	
$\hat{\theta}_h$	$\hat{\theta}_{a,h}$	$\hat{\theta}_{f,h}$	
$\rho_h(\phi(k))$	$\rho_{a,h}(\phi(k))$	$\rho_{f,h}(\phi(k))$	
\hat{c}_h	$\hat{c}_{a,h}$	$\hat{c}_{f,h}$	
$\phi(k)$			$[n_{en}(k), t_r(k - \delta)]^T$

Tab.1 Symbol connection between the general expression and the model

3. Predictive Control

The strategy of an "exceeding oxygen amount" control using a predictive controller is based on a prediction of a controlled quantity λ and subsequent minimization of a chosen cost function on the horizon N_p expressed in a standard

quadratic form. The value of λ is predicted by utilization of partially linear models of the air and fuel path. Through the independent air path model the proper amount of fuel is predicted and enters the cost function J . Hence, the target of the cost function minimization is to determine such a control law, that the measured system output λ is stoichiometric. The second modeled subsystem, the fuel-path, is an explicit component of the objective function where the amount of the fuel is the function of optimized control action (Poloni et al. (2008)).

3.1. Predictive model

The applied control strategy is based on the knowledge of the internal model (IM) of air-path, predicting the change of air flow through the exhaust pipe, and consequently, setting the profile of desired values of the objective function on the control horizon. In this case we will consider the state space (SS) formulation of the system and therefore it is necessary to express linear local ARX models in the SS structure with time varying parameters:

$$x_{(a,f)}(k+1) = A_{(a,f)}(\phi)x_{(a,f)}(k) + B_{(a,f)}(\phi)u_{(a,f)}(k)$$

$$m_{s,(a,f)}(k) = C_{(a,f)}x_{(a,f)}(k) \quad (12)$$

The weighted parameters of multi-ARX models are displayed in matrices $A_{(a,f)}$ and $B_{(a,f)}$ for both subsystems. This is a non-minimal SS representation whose advantage is, that no state observer is needed. The "fuel pulse width control" is tracking the air mass changing on a prediction horizon from IM of the air-path, by changing the amount of injected fuel mass. Due to tracking offset elimination, the SS model of the fuel-path (12) (index f), with its state space vector x_f , is written in augmented SS model form to incorporate the integral action:

$$\tilde{x}_f(k+1) = \tilde{A}_f(\phi)\tilde{x}_f(k) + \tilde{B}_f(\phi)\Delta u_f(k) \quad (13)$$

or

$$\begin{bmatrix} x_f(k+1) \\ u_f(k) \end{bmatrix} = \begin{bmatrix} A_f(\phi) & B_f(\phi) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_f(k) \\ u_f(k-1) \end{bmatrix} + \begin{bmatrix} B_f(\phi) \\ 1 \end{bmatrix} \Delta u_f(k)$$

$$m_{s,f}(k) = \tilde{C}_f\tilde{x}_f(k) + D_f\Delta u_f(k) \quad (14)$$

or

$$m_{s,f}(k) = [C_f D_f]\tilde{x}_f(k) + D_f\Delta u_f(k)$$

The prediction of the air mass (\overline{m}_a) on the prediction horizon (N_p) is dependent on the throttle position (\overline{t}_r) and is computed as:

$$\overline{m}_a(k) = \Gamma_a(\phi)x_a(k) + \Omega_a(\phi)\overline{t}_r(k-1) \quad (15)$$

where the x_a denotes the state space vector of the air path.

Due to the imprecise modeling (IM strategy), the biased predictions of the air mass future trajectory and consequently biased fuel mass might occur. This error is compensated incorporation the term $L[\hat{m}_f(k) - m_{s,f}(k)]$ into the fuel mass prediction equation:

$$\overline{m}_f(k) = \Gamma_f(\phi)\tilde{x}_f(k) + \Omega_f(\phi)\Delta\overline{u}_f(k-1) + L[\hat{m}_f(k) - m_{s,f}(k)] \quad (16)$$

The matrices of free response Γ_a , Γ_f and forced response Ω_a , Ω_f are computed from the SS model (12), respectively (Maciejowski, 2000). Since there is only $\lambda(k)$ measurable in equation (1), the value of $m_a(k)$ needs to be substituted using IM of the air-path, then:

$$\hat{m}_f(k) = \frac{1}{L_{th}} \frac{m_{s,a}(k)}{\lambda(k)} \quad (17)$$

The estimate $\hat{m}_f(k)$ is used to compensate for possible bias errors of predicted $\bar{m}_f(k)$ in (16).

3.2. Analytical solution

The analytical solution is based on the cost function (18), encompassing deviations of predicted fuel mass amounts between the air and fuel path (based on (1)); a penalization of control increments r ; and a penalization p of a deviation between a predicted and desired end state.

$$J_\lambda = \left\| \frac{\bar{m}_a(k)}{L_{ch}} - \bar{m}_f(k) \right\|_2^2 + r \left\| \Delta \bar{u}_f(k-1) \right\|_2^2 + p \left\| \tilde{x}_f(N) - \tilde{x}_{f,r}(N) \right\|_2^2 \quad (18)$$

The chosen MPC approach utilizes the state space representation with an integral control for the correction of the prediction. Due to a disturbance $d(k)$, the steady state values of u and x have to be adapted so, that the assumption $J = 0$ is valid. This problem solves an explicit inclusion of the disturbance into the model. The fuel injectors are controlled by a fuel pulse width, what is at the same time the control u_f . The optimal injection time can be computed by minimization of a cost function (18), which has after expansion by the fuel path prediction equation, form:

$$J_\lambda = \left\| \frac{\bar{m}_a(k)}{L_{ch}} - \Gamma_f \tilde{x}_f(k) + \Omega_f \Delta \bar{u}_f(k-1) \right\|_2^2 + r \left\| \Delta \bar{u}_f(k-1) \right\|_2^2 + p \left\| \tilde{x}_f(N) - \tilde{x}_{f,r}(N) \right\|_2^2 \quad (19)$$

An analytical solution of $\frac{dJ_\lambda}{du} = 0$ of (19) without constraints leads to an expression determining the change of "fuel injector opening time" in a step (k), as:

$$\Delta u = (\Omega^T \Omega + I r + p \Omega_{xN}^T \Omega_{xN})^{-1} * [\Omega^T [w(k) - \Gamma \tilde{x}(k)] - p \Omega_{xN}^T \Delta \tilde{x}_f(N) + p \Omega_{xN}^T \tilde{x}_{f,r}(N)] \quad (20)$$

Hence, the absolute value of the control action in a step k is given by a sum of a newly computed increment in a control (20) and an absolute value of the control in a step ($k-1$):

$$u_f(k) = u_f(k-1) + \Delta u_f(k) \quad (21)$$

4. Rapid Control Prototyping System

The computational unit necessary for the real-time implementation of the MPC control is based on a powerful and freely programmable control system based on *dSpace* and *RapidPro* units; or "Rapid Control Prototyping System" (RCP), (Fig. 2, dSPACE GmbH. (2009)).

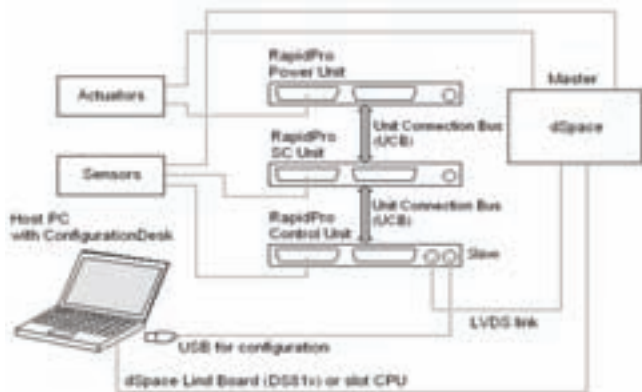


Fig.2 Scheme of the Rapid Control Prototyping System

It is built-up on the processor board *ds1005* and hardware-in-loop platform *ds2202 HIL*. The RCP ensures sufficient head-

room for the real-time execution of complex algorithms (Arsie et al. (2008)) and lets all engine tasks to be controlled directly. Also, the customized variants of the controller can be performed immediately.

Typical RCP system consists of:

- A math modeling program (prepared in Simulink)
- Symbolic input/output blocks
- A real-time target computer (embedded computer with an analog and digital I/O)
- A host PC with communication links to target computer
- A graphical user interface (GUI) which enables to control the real time process

The RCP system enables to use a support in the form of embedded functions which make the preparation of algorithms easy and fast. It is a great help, because one can then concentrate on significant problems (development and debugging of algorithms) without the spending time on not so important tasks (how to handle features of RCP system at low level programming).

5. Real-time Application of a Predictive Control

The ability to control the mixture concentration at stoichiometric level using MPC is demonstrated through the real-time SI engine control (Fig. 3).

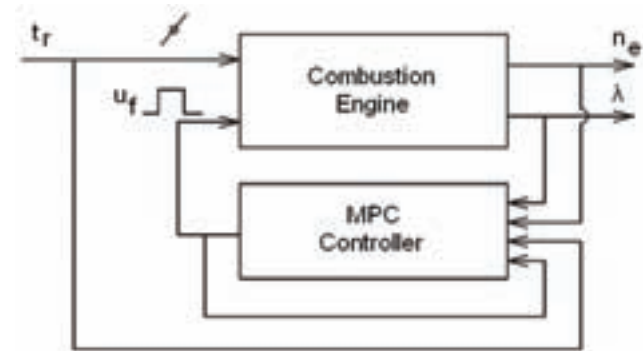


Fig.3 Control scheme

This has been performed using the AFR predictive control strategy described in the previous section, designed in *Matlab/Simulink* environment and compiled as a real-time application for a *dSpace* platform. It has been applied to the VW Polo engine (Fig. 4), 1390 cm³ with 55kW at 5000 rpm, not equipped with a turbocharger or an exhaust gas recirculation system.



Fig.3 Spark ignition engine VW Polo 1.4

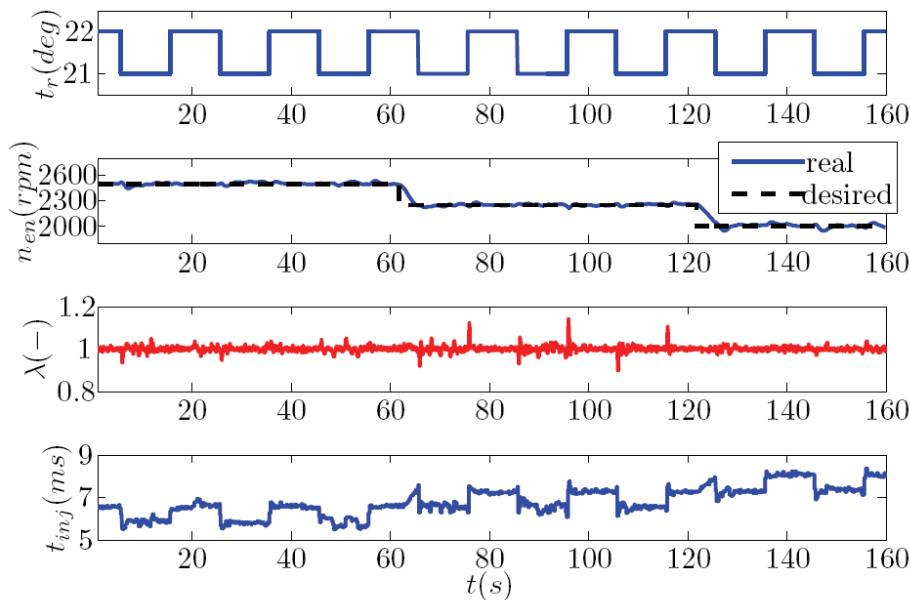


Fig.3 Results of an AFR SI engine control

The control period was 0.2s. The result of an identification are 9 local linear models (LLM) for each, air and fuel path, dependent on a throttle valve opening and engine revolutions.

The primary target of a control (Fig. 5) was to hold the air/fuel ratio in a stoichiometric region ($\lambda = 1$), in the worst case to keep the mixture ignitable ($0.7 \leq \lambda \leq 1.2$). During the experiment, the change in throttle valve opening, between 21 and 22 degrees (Fig. 5, variable t_r) and the change of engine revolutions (Fig. 5, variable n_{en}), has been performed several times. These changes simulate varying working regimes of an engine, which is adapting its run to a daily traffic. Changes in t_r and n_{en} quantities are determining the engine load, at the same time, ensuring, that the engine passes through several working points during its operation. As mentioned in Section 2, the engine revolutions are not included among explicit variables of local models, but they build together with a delayed throttle valve position a vector of an working point $\phi(k)$.

The quality of control is sufficient (Fig. 5, variable λ), with exceptional acceptable overshoots in both directions. These overshoots of the controlled variable λ have been caused by smaller model precision, due to its distance from the working point, at which the system identification has been performed. This effect is caused by the approximation of a particular model from the other working points' models.

The corresponding control (fuel injection time) computed by the controller is shown in (Fig. 5, variable t_{inj}). The initial engine warm-up (to 80 °C) eliminated model-plant mismatch caused by temperature dependent behavior of the engine.

The control has been performed by choosing the penalization $r = 0.1$. Utilizing the member $p \|\tilde{x}_f(N) - \tilde{x}_{f,r}(N)\|_2^2$ of a cost function by setting $p = 1.0$ allowed us to shorten the control horizon to $N_p = 20$ what significantly unloaded the computational unit and stabilized the controlled output of the engine on this shortened horizon, as well. The best control has been achieved in the neighborhood of working points, what is logically connected to the most precise engine model at those points. In other working points the control is still good enough, with small deviations from the stoichiometric mixture.

6. Conclusion

Considering the preliminary results from the real-time experiments at the engine, it can be concluded, that the idea of the AFR model predictive control based on local ARX models is suitable and applicable for the SI engine control. The proposed flexible design of a predictive controller offers easy tuning possibilities and a potential for the model accuracy improvement by the extension of the global engine model to other working regimes of the engine. The next project step shall be the overshoot elimination in the λ -control by the identification of wider net of "local linear engine models" and implementation of constraints. Another task which has to be done is a comparison of the quality of control gained by the MPC controller with a baseline electronic control unit. This goal has been not yet achieved, as the original ECU has been replaced by the dSpace system running our controller.

Acknowledgments

The work has been supported by the Slovak Research and Development Agency under grant LPP-0075-09, LPP-0118-09 and LPP-0096-07. This research is also supported by the grant from Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. This support is very gratefully acknowledged.

References

- [1] ARSIE, I., IORIO, S.D., NOSCHESI, G., PIANESE, C., and SORRENTINO, M. (2008). Optimal air-fuel ratio. *dSpace Magazine*, (1), 20-23.
- [2] BENGTTSSON, J., STRANDH, P., JOHANSSON, R., TUNESTAL, P., and JOHANSSON, B. (2007). Hybrid modeling of homogenous charge compression ignition (HCCI) engine dynamics -a survey. *International journal of control*, 80(11), 1814-1847.
- [3] dSPACE GmbH. (2009). *HelpDesk Application*.
- [4] HOU, Z. (2007). Air fuel ratio control for gasoline engine using neural network multi-step predictive model. 3rd inter-

national conference on intelligent computing, Qingdao, China.

[5] LORINI, G., MIOTTI, A., and SCATTOLINI, R. (2006). Modeling, simulation and predictive control of a spark ignition engine. In Predimot (ed.), *Predictive control of combustion engines*, 39-55. TRAUNER Druck GmbH & CoKG.

[6] MACIEJOWSKI, J.M. (2000). *Predictive control with constraints*. University of Cambridge.

[7] MAO, X., WANG, D., XIAO, W., LIU, Z., WANG, J., and TANG, H. (2009). Lean limit and emissions improvement for a spark-ignited natural gas engine using a generalized predictive control (GPC)-based air/fuel ratio controller. *Energy & Fuels*, (23), 6026-6032.

[8] MAYNE, D.Q., RAWLINGS, J.B., RAO, C.V., and SOKKAERT, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, (36), 789-814.

[9] MURRAY-SMITH, R. and JOHANSEN, T.A. (1997). *Multiple model approaches to modelling and control*. Taylor & Francis.

[10] MUSKE, K.R. and JONES, J.C.P. (2006). A model-based SI engine air fuel ratio controller. American Control Conference, Minneapolis, USA.

[11] POLONI, T., JOHANSEN, T.A., and ROHAL-ILKIV, B. (2008). Identification and modeling of air-fuel ratio dynamics of a gasoline combustion engine with weighted arx model network. Transaction of the ASME (*Journal of Dynamic Systems, Measurement, and Control*), 130(6). 061009.

[12] POLONI, T., ROHAL-ILKIV, B., and JOHANSEN, T.A. (2007). Multiple ARX model-based air-fuel ratio predictive control for SI engines. In *IFAC Workshop on advanced fuzzy and neural control*. Valenciennes, France. Conference paper MO5-3.

[13] ZEMAN, J. and ROHAL-ILKIV, B. (2003). Robust min-max model predictive control of linear systems with constraints. 930 - 935. IEEE International Conference on Industrial Technology.

[14] ZHAI, Y.J., DING-WENYU, HONG-YUGUO, and D.L.YU (2010). Robust air/fuel ratio control with adaptive DRNN model and AD tuning. *Engineering Applications of Artificial Intelligence*, (23), 283-289.

[15] ZHAI, Y.J. and YU, D.L. (2009). Neural network model-based automotive engine air/fuel ratio control and robustness evaluation. *Engineering Applications of Artificial Intelligence*, (22), 171-180.

Abstract

The following paper deals with the air/fuel ratio (AFR) controller applied on spark ignition engine. It utilizes the analytical model predictive controller based on the multi-model approach which employs the autoregressive model (ARX) network and the weighting of local models, coming from the sugeno-type fuzzy logic. The weighted ARX models are identified in the particular working points and are creating a global engine model, covering its nonlinearity. Awaited improvement of a proper air/fuel mixture combusted in a cylinder

is mostly gained in the transient working regimes of an engine. In these regimes, the traditional control approach loses its quality, compared to steady state working regimes of an engine. This leads to higher fuel consumption and level of emissions from an engine. Real-world experiments related to the VW Polo 1390cm³ engine, at which the original electronic control unit (ECU) has been replaced by a dSpace system executing the model predictive controller. Results acquired during the researches prove, that the proposed controller is suitable for the air/fuel ratio control giving sufficiently good and steady system output.

Ing. Matúš Kopačka

Slovak University of Technology
Faculty of Mechanical Engineering
Institute of Automation, Measurement and Applied Informatics
Nám. slobody 17
812 31 Bratislava 1
matus.kopacka@stuba.sk

Ing. Peter Šimončíč

peter.simoncic@stuba.sk

Ing. Jozef Csambál

jozef.csambal@stuba.sk

Ing. Marek Honek

marek.honek@stuba.sk

mgr. inž. Sławomir Wojnar

slawomir.wojnar@stuba.sk

Ing. Tomáš Polóni, PhD.

tomas.poloni@stuba.sk

prof. Ing. Boris Rohal-Ilkiv, CSc.

boris.rohal-ilkiv@stuba.sk

PIDTOOL 2.0 – Software for Step-Response-Based Identification and PID Controller Tuning

Juraj Oravec, Monika Bakošová

Abstract

The main aim of this paper is to present a novel version of software for PID controller tuning called PIDTOOL 2.0. The software represents a user friendly tool for simple step-response-based identification of a process model, fast PID controller tuning, and effective checking the quality of control. The software can be used as useful and visual tool for teaching purposes in laboratory exercises oriented on process control. In PIDTOOL 2.0, user can easily change a language of the graphic user interface. Nowadays, there is a possibility to choose between English and Slovak. The software has been developed in the MATLAB-Simulink programming environment using its graphic user interface

Key words: PID controller, controller tuning, identification, filtration, graphic user interface

Introduction

The aim of this paper is to present a new version of software for PID controller tuning called PIDTOOL 2.0 [1], [2]. It has been developed at the Institute of Information Engineering, Automation, and Mathematics of the FCFT STU in Bratislava [1], [3] in the MATLAB - Simulink programming environment and uses its graphic user interface (GUI).

As PID controllers belong to the most used types of controllers in industry [4], PIDTOOL 2.0 is oriented mainly on PID controller tuning. The software enables to tune PID controllers using various analytical and experimental methods, and new methods described in [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] were added. If a transfer function of the controlled process is unknown, the software enables to identify the controlled process from its step response [16].

The identified step response can be either damped periodic or aperiodic. The step-response data can be set directly or loaded from the data file. If noisy step-response data have been loaded, designed software enables to run filtration. Properties of the tuned controller can be judged visually and analytically, as PIDTOOL 2.0 displays simulated control response, time behavior of the manipulated variable and values of various integral performance indexes. So, it is easy to compare several closed-loop step responses generated using various PID controllers with different values of set-points, disturbances and constraints on manipulated variables.

The software PIDTOOL 2.0 can be used especially for teaching purposes in laboratory exercises oriented on process control. It can be also useful for those who need to identify controlled process from its step-response data, to filter noisy data, to tune PID controllers or to compare various types of control algorithms with simple PID control.

1. PIDTOOL 2.0

PIDTOOL 2.0 solves two basic problems, identification and controller tuning (Fig. 1).

PIDTOOL 2.0 enables to identify a controlled process from its step response. The software distinguishes identification from an aperiodic or a damped periodic step response.



Fig. 1 Basic window of PIDTOOL 2.0

The Strejc method [6], [7], [10] is applied for identification from the aperiodic step response and the method described in [7] is used for identification from the damped periodic step response. The result of identification is a controlled process model described by the transfer function (1) for aperiodic or (2) for damped periodic step response

$$G(s) = \frac{K}{(Ts + 1)^n} e^{-Ds} \quad (1)$$

$$G(s) = \frac{K}{T^2 s^2 + 2\zeta Ts + 1} e^{-Ds} \quad (2)$$

where n is the order of the system, K is the gain, T is the time constant, ζ is the damping coefficient and D is the time delay.

The identification can be simply started using button *Identification* located in the basic window (Fig. 1). The next window (Fig. 2) offers three identification possibilities.

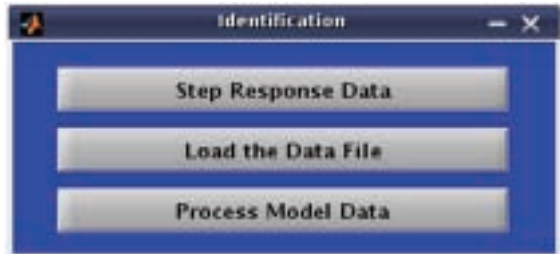


Fig. 2 Basic window of identification

The first button *Step Response Data* enables to identify the controlled process directly from data obtained from the measured and recorded step response.

The second button *Load the Data File* (Fig. 2) opens the new window (Fig. 3), where user can comfortably find out a required data file containing recorded step-response data. The considered structure of the data file is as follows, the first column vector represents a time and the second column vector represents associated measured values of output variable.

PIDTOOL 2.0 enables to load the data file which includes also the third column vector of values of manipulated variable (Fig. 4). If several step responses are included in the loaded data file, they are automatically recognized and the nominal step response is evaluated (Fig. 5). It enables to reach the nominal transfer function of identified non-linear controlled process. This possibility makes the identification from step response using this software even user-friendlier.

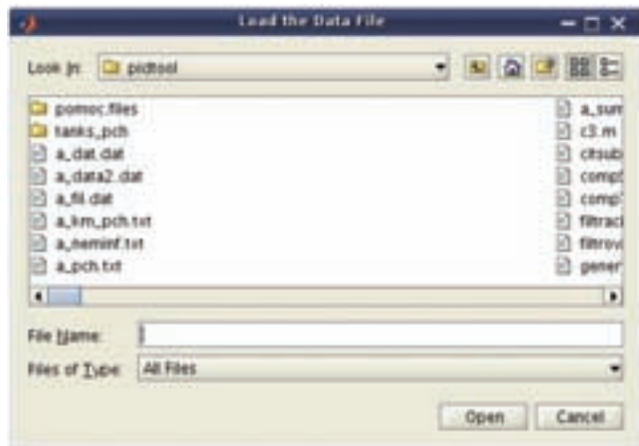


Fig. 3 Window for loading the datafile

In the next window, there is a choice of data processing (Fig. 6). To obtain the aperiodic model of controlled process, user can directly use the button *Identification*.

When the damped periodic model is required, user can simply activate the checkbox *Periodic process* and then use the button *Identification*. If the controlled process has been identified using the Strejc method, the tangent to the step response is also depicted and its equation is given (Fig. 7). In the new window, the parameters of model (1) or (2) of identified process are shown (Fig. 8, Fig. 9).

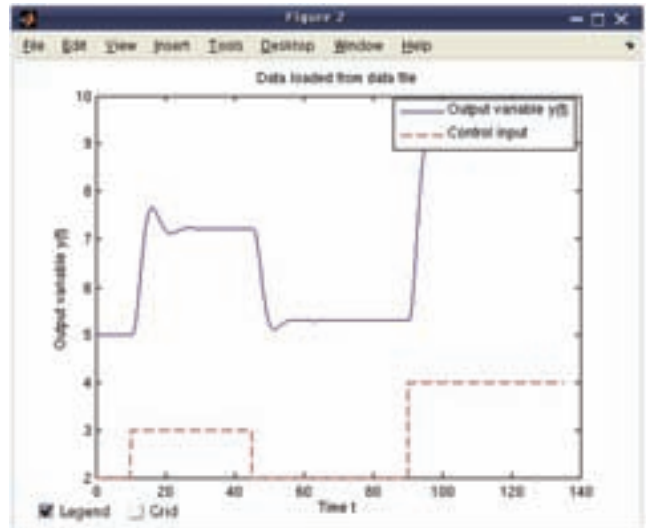


Fig. 4 Loaded data of step response

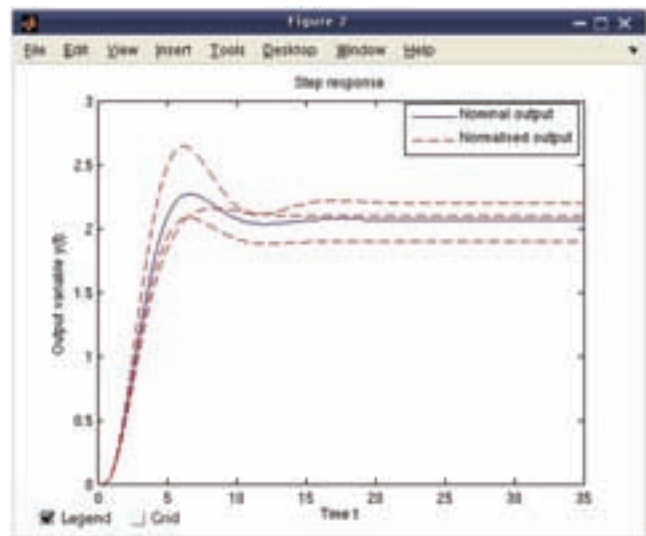


Fig. 5 Nominal and normalized step responses



Fig. 6 Choice of processing of loaded data

PIDTOOL 2.0 enables to use button *Identification Tuning* (Fig. 10) to receive the transfer function, which generates the step response that covers the original one more precisely. In the new window (Fig. 11) the step response of identified transfer function can be simply modified by changing the slope of its tangent [10]. The new parameters of identified transfer function are directly shown. It helps to check whether the identified transfer function has still required properties, e.g. the order n . If the loaded data are noisy, the user can use the filtration before identification, simply using the button *Filtration* (Fig. 6). Then the new window for filtration is opened (Fig. 12). By using the button *Save* (Fig. 12), user can simply store reached filtered data into the new data file for later usage. After filtration, identification can be started.

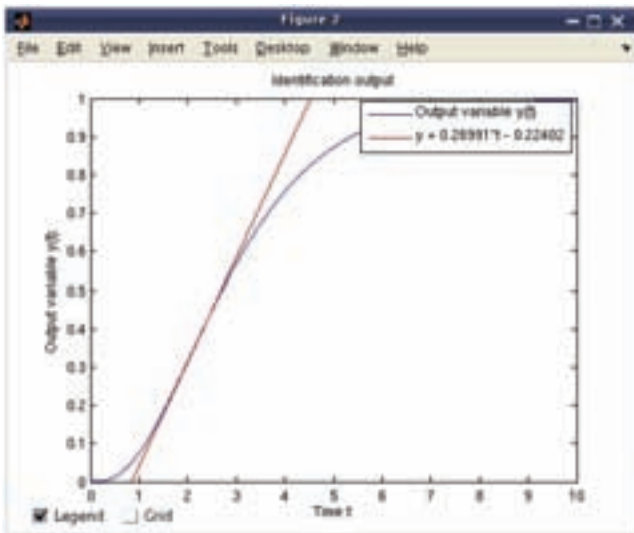


Fig. 7 Step response of identified process



Fig. 8 Identified parameters of aperiodic system



Fig. 9 identified parameters of damped periodic system

The third button *Process Model Data* (Fig. 12) enables to identify process model with required properties of transfer function. Using this button, the new window shown in Fig. 13 will be opened. In this window, the parameters of the model described by the transfer function (3) can be simply set.

$$G(s) = \frac{Num(s)}{Den(s)} e^{-Ds} \quad (3)$$

If the checkbox *Periodic process* has been activated (Fig. 13), given model (3) will be approximated by the transfer function (2), otherwise by the transfer function (1). The approximation is started using the button *Identification* (Fig. 13). This approximation of the given model can be

useful in the case when chosen PID controller tuning method requires controlled model described by the transfer function (1) or (2).

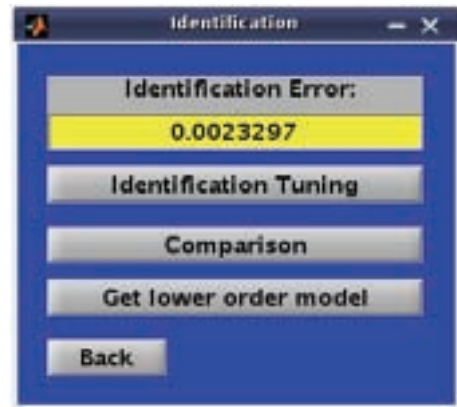


Fig. 10 Additional options of identification

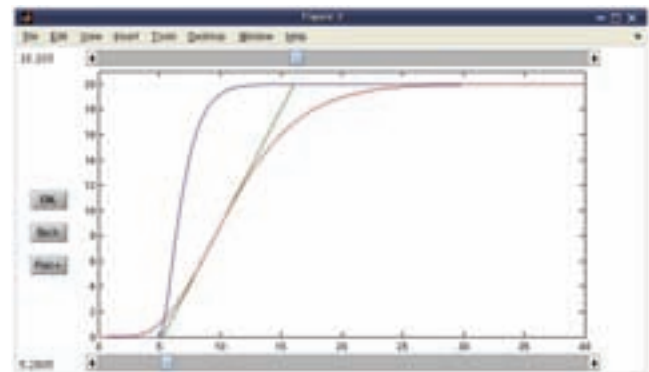


Fig. 11 Window for identification tuning

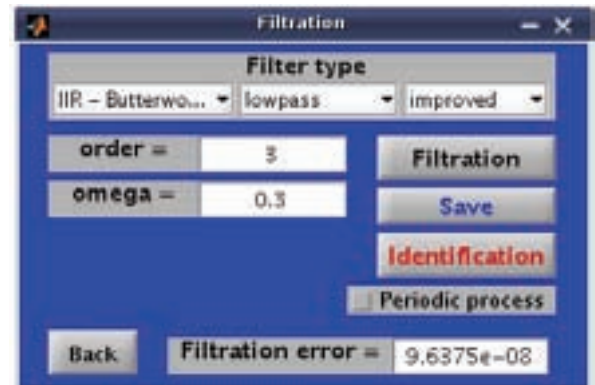


Fig. 12 Window of filtration

The software PIDTOOL 2.0 is oriented mainly on PID controller tuning. PID controllers can be designed for controlled process models with either damped periodic or aperiodic step responses described by the transfer functions (1) or (2). To run direct controller tuning, user can use the button *Controller Tuning* located in the basic window (Fig. 1), or use this button after the identification (Fig. 8, Fig. 9). Then, a window is opened where the user can choose a required type of PID controller and a type of a tuning method (Fig. 14).

It is possible to choose a P, PI, PID or PD controller. For simpler handling, the methods for controller tuning are divided into two main groups: analytical and experimental methods. Various types of experimental (Fig. 15) and analytical methods (Fig. 16) can be used for controller tuning. Calculated parameters of the tuned controller are shown in the new window (Fig. 17), where Z_R is the gain, T_I is the reset time and T_D is the derivative time of the controller.

In the top right part of the opened window (Fig. 17), the parameters of the transfer function of the controlled process

are also shown. These parameters can be modified and the tuned controller can be so tested in the presence of model uncertainty. The properties of the closed loop with the tuned controller can be judged by simulation of control. The standard control law (4) is supposed as a default control law.

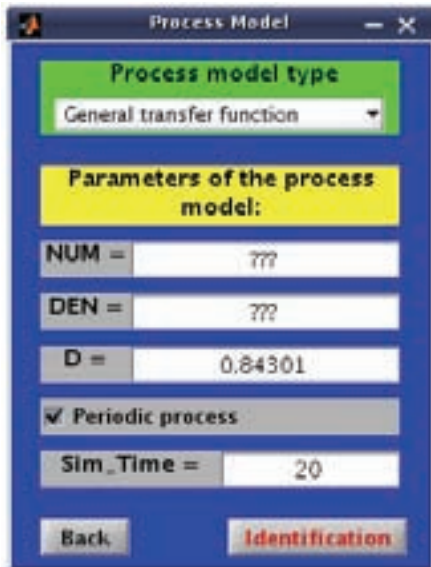


Fig. 13 Process model data

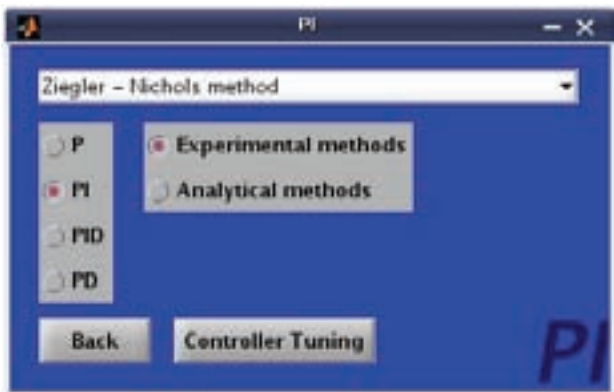


Fig. 14 Window for PID controller tuning

Setting the parameters b_w , T_R and T_F enables to use improved form of control law. The proportional part of control law is then modified by the value of the parameter b_w . Setting this parameter enables to change the weight of set-point $w(t)$ in the control error evaluation (5). The non zero value of the parameter T_R modifies the integral part of control law (6). Setting this parameter enables to prevent integral windup. The parameter T_F modifies the derivative part of control law (7). The non zero value of this parameter represents a filter of derivative part of control law to obtain the proper transfer function of the derivative part of the controller.

$$u(t) = Z_R e(t) + \frac{Z_R}{T_I} \int_0^t e(t) dt + Z_R T_D \frac{de(t)}{dt} \quad (4)$$

$$u_P(t) = Z_R (b_w w(t) - y(t)) \quad (5)$$

$$u_I(t) = \frac{Z_R}{T_I} \int_0^t \left(e(t) - \frac{T_I}{Z_R} T_R (u(t) - u_{SAT}(t)) \right) dt \quad (6)$$

$$u_D(s) = \frac{Z_R T_D s}{1 + \frac{T_D}{T_F} s} \quad (7)$$

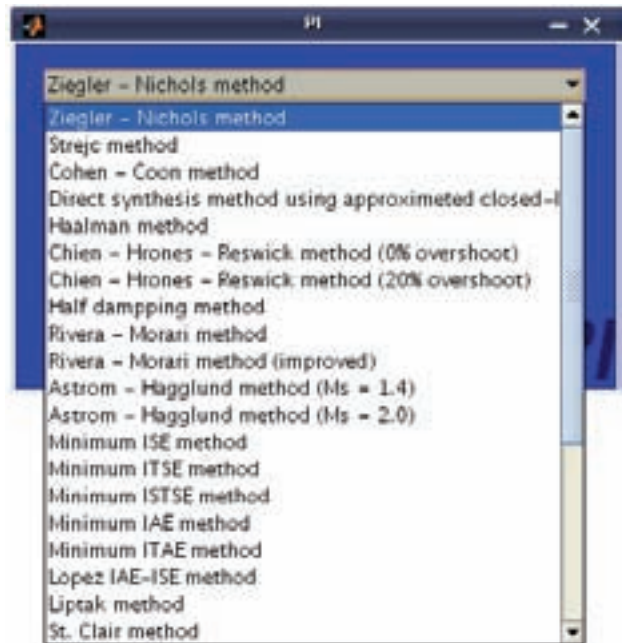


Fig. 15 List of experimental methods used for PI controller tuning



Fig. 16 List of analytical methods used for PID controller tuning

Using the button *Step Response* (Fig. 17) runs quality evaluation of control. The set-point tracking and the disturbance rejection can be simulated in the presence of boundaries on the control input. For simpler handling, the parameters have preset default values. In the case, the manipulated variable is constrained, user can compare the closed-loop step responses and the time behaviors of the manipulated variable before and after the saturation. After simulation, the closed-loop step responses (Fig. 18) and the time behaviors of the manipulated variable (Fig. 19) are shown. In the case the legend overlaps the displayed graph (Fig. 18), it is possible to deactivate the checkbox *Legend* and to hide the shown legend. Using the checkbox *Grid* leads to displaying the grid of the shown graph. Both of these possibilities are included in all displayed graphs, generated by this software. The quality of control performance (Fig. 20) can be also judged by calculating several integral performance indexes [1], [3], [5], [7], [13].

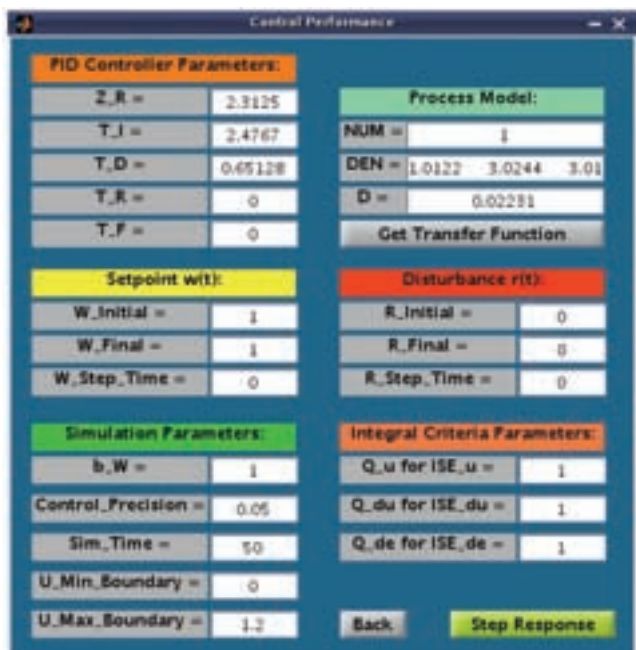


Fig. 17 Window for evaluating the quality of control

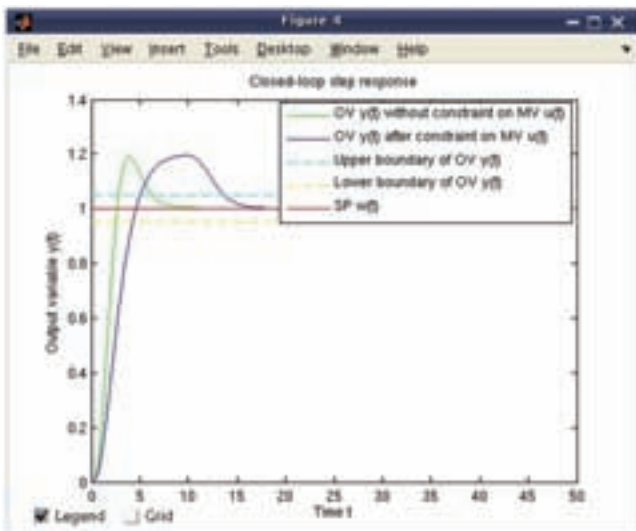


Fig. 18 Closed-loop step response

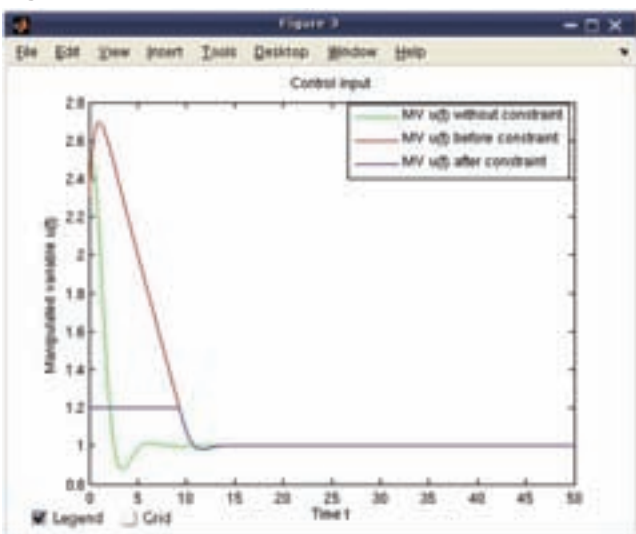


Fig. 19 Control input

Using the possibilities of window shown in Fig. 17, it is easy to compare several step responses and values of performance indexes reached with different values of set-points, disturbances and constraints on the manipulated variables.

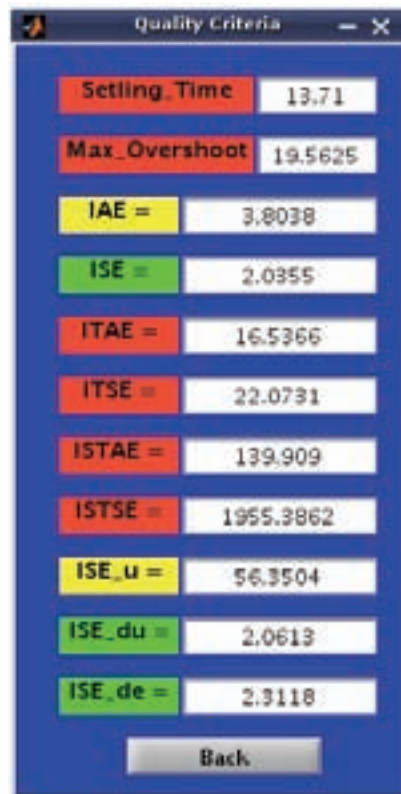


Fig. 20 Values of performance indexes

The values of all the parameters, which are necessary for simulations, are stored by PIDTOOL 2.0. Calculated values of performance indexes are also stored. These stored data can be simply shown by using *Setup/Show results* located in basic window (Fig. 21). The stored data are transformed into *html*-file. The stored data are shown in simple summary table with date and time of simulation. This new ability can be helpful in the case, when many simulations at different conditions have been evaluated and user wants to compare obtained results to make decision, which controller is the most suitable for control.

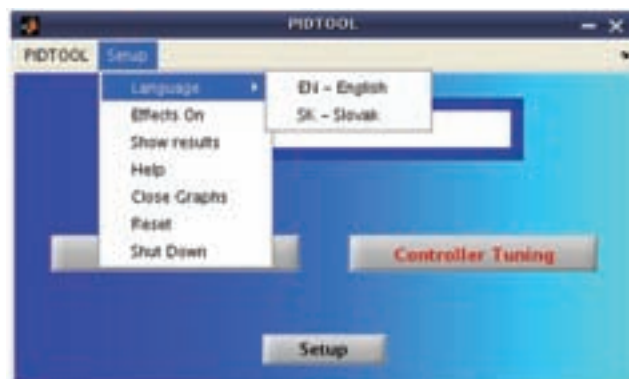


Fig. 21 Setup menu of PIDTOOL 2.0

It is easy to change a language of graphic user interface. Actually, there is a possibility to choose between English and Slovak language (Fig. 21). Also other languages can be simply added into the software PIDTOOL 2.0.

Conclusion

The software PIDTOOL 2.0 has been developed in the MATLAB - Simulink programming environment using its Graphical User Interface and offers to the user a comfortable and visual environment for fast identification, simple PID

controller tuning and effective evaluating the quality of control in various conditions. The values necessary for evaluating the quality of control and calculated quality criteria are stored and can be displayed in a simply summary table. Using this new ability helps to decide, which controller is the most suitable for control. This software has been tested by students at the FCFT in two courses; in the course Process Dynamics and Control that is taught in the first year of the Master study and the course Integrated Control in the Process Industry that is taught in the third year of the Bachelor study. Using the software is limited only for teaching purposes. To obtain the software contact the authors.

There exist various software applications, which enables to design the PID controller, for example see [10] or [17]. The comparison of the available applications will be subject of our next work.

Acknowledgments

The authors are pleased to acknowledge the financial support of the Cultural and Educational Grant Agency KEGA of the Slovak Republic under grant No. 3/7245/09 and of the Scientific Grant Agency VEGA of the Slovak Republic under the grants 1/0537/10.

References

- [1] ORAVEC, J. (2010). Design of a Program System for Controller Tuning (in Slovak). Master's thesis, FCFT, SUT in Bratislava.
- [2] BAKOŠOVÁ, M. and ORAVEC, J. (2009). Software for PID controller tuning. Proceedings of the 17th Interantional Conference on Porcess Control'09, FCFT SUT in Bratislava, 524–527.
- [3] ČEMANOVÁ, L. (2007). Design of GUI for controller tuning (in Slovak). Master's thesis, FCFT SUT in Bratislava.
- [4] ÅSTRÖM, K. and HÄGGLUND, T. (1995). PID controllers: theory, design and tuning. Triangle Park, 2nd ed. edition.
- [5] XUE, D. and CHEN, Y.Q. (2008). Analysis and Design with MATLAB (Advanced in Design and Control). Society for Industrial Mathematics, 1st ed. Edition
- [6] BAKOŠOVÁ, M. and FIKAR, M. (2008). Process Control (in Slovak). STU Press, Bratislava.

- [7] MIKLEŠ, J. and FIKAR, M. (2007). Process Modelling, Identification, and Control. Springer Verlag, Berlin Heidelberg.
- [8] OGUNNAIKE, B. and RAY, W.H. (1994). Process dynamics, modeling and control. Oxford University Press, New York.
- [9] VÍTEČKOVÁ, A. and VÍTEČEK, M. (2006). Automatic control fundamentals (in Czech). TU Ostrava.
- [10] LOŠONSKÝ, J. (2006). System for support of control loops (in Slovak). Master's thesis, Trnava: MTF SUT in Bratislava.
- [11] WENG, K.H., CHANG, C.H., and LISHENG, S.C. (1995). Tuning of PID Controllers Based on Gain and Phase Margin Specifications. Pergamon, Automatica, 31(3), 497–502.
- [12] SHAW, J.A. (2006). The PID Control Algorithm [online]. <http://www.docstoc.com/docs/20312583>, 2nd ed., version 2.2 edition.
- [13] BAKOŠOVÁ, M., L. ČIRKA, and FIKAR, M. (2003). Automatic control fundaments (in Slovak). Vydavateľstvo STU, 1st ed. edition.
- [14] KUBÍK, S. and KOTEK, Z. (1983). Automatic control theory (in Czech). SNTL, 1st ed. edition.
- [15] ŠULC, B. and VÍTEČKOVÁ, M. (2004). Theory and raxis of control design (in Czech). ČVUT Press, Praha.
- [16] FIKAR, M. and MIKLEŠ, J. (1999). System identification (in Slovak). STU Press, Bratislava.
- [17] SCHLEGEL, M. and ČECH, M. (2004). Internet PID controller design : www.pidlab.com. IBCE 04, 1–6.

Ing. Juraj Oravec,
Doc. Monika Bakošová, CSc.

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.: (02)59325365
E-mail {juraj.oravec, monika.bakosova@stuba.sk

Real-Time Implementation of Explicit MPC Using PLC

Ivana Rauová, Richard Valo, Michal Kvasnica and Miroslav Fikar

Abstract

We consider the problem of real-time implementation of explicit Model Predictive Control (MPC) feedback laws using Programmable Logic Controller (PLC) platform. A controller defined by a continuous Piecewise Affine (PWA) function represents a solution of the MPC problem. The challenge becomes how to make browsing through the regions of the controller easier and more applicable on memory-restricted devices. The proposed design procedure is illustrated on a real-time control of a laboratory heat exchange plant.

Keywords: *model predictive control, programmable logic controllers*

Introduction

Model predictive control (MPC) is an attractive approach widely used in industry to control a broad range of the systems due to its ability to provide optimal performance while taking process constraints into account [5]. In MPC the control objectives are translated into an optimization problem, which is formulated over a finite prediction horizon. The result of the optimization is a sequence of optimal control moves which drives system states towards a given reference point while respecting system constraints (such as upper and lower limits on the inputs and states) and optimizing a selected performance criterion. Traditionally, MPC is implemented in a so-called Receding Horizon fashion where the optimal control move is achieved by solving optimization problem in each time instance for a newly measured state. This approach induces lot of computation load at each sampling time, which might be prohibitive if not enough computation power is available or if sampling time is too short.

If less powerful control platforms are employed, additional care has to be taken to respect real-time constraints. One approach to decrease computational load involved in obtaining of optimal control action u^* for a particular value of x is to “pre-compute” the optimal solution to a given optimization problem for all possible initial conditions of x using parametric programming techniques [1]. The optimal control can be then found as an explicit function $u^*(x)$ mapping the states to the control inputs. The function is computed off-line and takes a form of lookup table. Implementation of such a table can be done very efficiently on-line, as the evaluation of the feedback law involves only matrix multiplications, additions and logic comparisons. As a consequence, real-time implementation of such an *explicit* MPC can be done much faster compared to traditional on-line MPC fashion.

In this work we aim at implementing explicit MPC on a Programmable Logic Controller (PLC) restricted to 1024 bytes of memory. Three factors determine whether the design procedure will be successful:

- whether it is possible to construct the explicit MPC controller off-line in an automated fashion;
- whether the controller is reasonably small as not to exceed the memory capabilities of the PLC;

- whether the controller can be implemented using programming instructions which the control device understands.

In the paper we illustrate how to synthesize the parametric solution to MPC optimization problem using the Multi-Parametric (MPT) Toolbox [3] and how to implement it on a PLC. First, we introduce the controlled plant and derive its mathematical model. Then we show which commands have to be used to set up the MPC optimization problem and how to synthesize the explicit MPC controller using MPT. Having a lookup table we introduce an algorithm capable to transform it into a binary search tree (BST), which can be downloaded directly to the PLC. At the end we show results of the laboratory plant control using explicit MPC controller provided on the PLC.

1. Physical Setup

1.1 Podnadpis

The laboratory Air-stream and Temperature Control Plant LTR 700 [8] is produced by the German company Amira. It consists of a fan, a heating coil, a differential pressure sensor, a temperature sensor, and an actuator box.

This plant is designed for heating the entering medium. Commonly, air is the medium which is intake to the plant thanks to the fan. The entering air is further heated by heating coil. In order to obtain hot air of desired amount and temperature, we can manipulate the fan speed or the amount of heat generated by heating coil. In this work, the heating coils output is set to constant value (50%). We aim at controlling only the air-flow rate (manipulated variable) by fan speed (control variable).

The plant is schematically illustrated in Fig. 1. Actual temperature and air-flow can be measured by sensors (TI, FI).

It is possible to implement several control configurations ranging from simple feedback loops, through cascade loops, up to multivariable control with two inputs and two outputs.

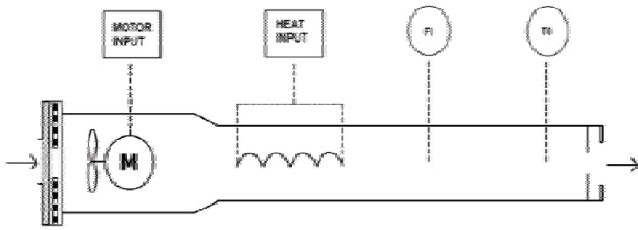


Fig.1 Sensors in the air heater.

1.1.1 Airflow

The plant converts air-flow value to a current signal in a range of 10.4-20mA. This output signal is connected to the input of analog I/O module which is shown in Fig. 2. For our control program, the connected signal is converted into an integer number with physical address $A/W4$ (A –analog, I –input of the PLC, W –memory size of 16 bits, 4– input to PLC). This integer is converted into a corresponding quantity expressed in mA units by the following relation:

$$x_{[mA]} = \frac{AIW4 - 0.596}{1583.26}, \quad (1)$$

Corresponding value of the state in a percentage range is achieved by

$$x_{[\%]} = \frac{x_{[mA]} - 10.4}{0.096}, \quad (2)$$

2.1.2 Fan speed

Formula which relates actual control action $u_{[\%]}$, expressed in percentage range, and a current signal output to the fan engine in mA units is

$$u_{[mA]} = 0.2u_{[\%]} \quad (3)$$

Therefore, corresponding integer representation of the output is

$$AQW0 = 1585.917u_{[mA]} + 51.23, \quad (4)$$

where $AQW0$ is a physical address (A –analog, Q –output of the PLC, W –memory size of 16 bits, 0– output from PLC). This input signal is connected to the output of analog I/O module which is shown in Fig. 2.

The mathematical model of the fan airflow can be captured by one differential equation of the following form

$$\frac{dV}{dt} = kD_M^3 f - V, \quad (5)$$

Here, V represents volume of air flow, k denotes a proportionality coefficient as a function of the Reynolds number of blender, D_M is blender diameter and f represents blender rotation frequency. By linearizing (5) around the steady state f^s and V^s , the following transfer function model can be derived

$$G = \frac{kD_M^3}{s+1} = \frac{Z}{T_v s + 1}, \quad (6)$$

where Z denotes gain of the system and T_v represents time constant.

Corresponding state-space representation of the fan is

$$\dot{x} = Ax + Bu \quad (7a)$$

$$y = Cx + Dy \quad (7b)$$

where $x = V - V^s$ is the state and $u = f - f^s$ is the input. Based on the steady state values of the variables mentioned above, the matrices are defined by

$$A = -\frac{1}{T_v}, B = \frac{1}{Z}, C = 1, D = 0.$$

This linear state-space representation can then be used to find a closed-form representation of the MPC feedback law by using techniques of parametric programming as described in Section 2.

1.2 PLC

A programmable logic controller (PLC) is a special digital computer often used in process automation such as for control of machinery on factory assembly lines, amusement rides, or lighting fixtures. Unlike general-purpose computers, PLCs are designed for multiple input and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are performed in constant length cycles [6].

In this work, we have used the SIMATIC S7-200 micro PLC, which is exceptionally compact, remarkably capable, fast and comes with easy to operate hardware and software. It has a modular design, still open-ended enough to be expanded. The main components of the selected PLC are briefly described next.

1.2.1 CPU

The S7-222 CPU can be seen on the left side in Fig. 2. Important to notice is that the CPU only provides 1024 bytes of memory for program data. This limit is both restrictive and challenging from the control synthesis point of view. Another limitation is that control algorithms have to be developed using so-called *ladder logic*, a visual programming language which only requires the algorithm to be composed of most basic operations (e.g. sums, products, comparisons, etc.).

Real-time data measurements can be stored on a memory cartridge [6], marked by “MC” in Fig.2. In our case it provides a 256kB storage for measured signals, which could be captured at a 0.04 second sampling rate. The captured data can subsequently be opened in MS-Excel.

1.2.2 Power Source

Power source (LOGOPower 6EP1332-1SH42) [6, 7] is a standard transformer used to supply PLC from public power network (see Fig. 2), where the “Power Network” is 1-phase AC line supply with voltage rate of 100-240 V (50/60Hz) to isolated output voltage 24 V DC.

1.2.3 Analog I/O Module

Module EM235 is necessary for the PLC to communicate with the controlled plant by means of analog signals. The module is shown in Fig. 2, where it is situated in between the central processor unit and the power source. The module allows 4 analog inputs and one analog output to be connected. Inputs have to be connected and configured corresponding way. The outputs in addition have to have correct HW configuration. For setup the outputs to the 0-20 mA we have to set a set of 6 switches. The set is placed right bottom corner of the EM235 in Fig. 2. For our purpose we have set them as in Fig. 3. The PLC communicates with the master PC by a 9 pin RS 486 port, located under the memory cartridge.

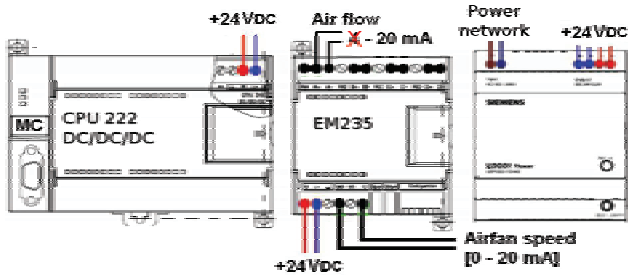
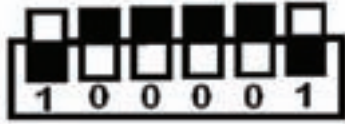


Fig.2 PLC connections to the plant.



Fi.3 Setting of the EM 235 inputs.

2 Explicit Model Predictive Control

In model predictive control (MPC), the optimal control actions are found by optimizing for plant behavior while taking process constraints into account. This is always achieved by formulating and solving an optimization problem where a given objective function is minimized subject to the constraints. A model of the plant is employed as an additional constraint to capture the predicted evolution of the plant:

We consider the discrete-time, stabilizable linear time-invariant model given by the state-space representation

$$x(k+1) = Ax(k) + Bu(k), \quad (8a)$$

$$y(k) = Cx(k), \quad (8b)$$

Here, $x(k)$ denotes state at time instance k , $x(k+1)$ is a successor state, $u(k)$ is control input, and $y(k)$ is a system output. It is assumed that variables are constrained by upper and lower limits

$$\underline{x} \leq x(k) \leq \bar{x}, \quad \underline{u} \leq u(k) \leq \bar{u}, \quad \underline{y} \leq y(k) \leq \bar{y} \quad (9)$$

For the system (10) consider now the constrained finite-time optimal control problem

$$\min_{\Delta u_0, \dots, \Delta u_{N-1}} \sum_{k=0}^{N-1} \|R\Delta u_k\|_p + \|Q(y - y_{ref})\|_p \quad (10a)$$

$$\text{s.t. } x_0 = x(t), \quad (10b)$$

$$x_{k+1} = Ax_k + Bu_k, \quad (10c)$$

$$y_k = Cx_k, \quad (10d)$$

$$x_k \in X, \quad (10e)$$

$$y_k \in Y, \quad (10f)$$

$$u_k \in U, \quad (10g)$$

Here p denotes a matrix norm (either $p=1$, $p=2$ or $p=\infty$), the integer N represents the finite prediction horizon, and R , Q are weighting matrices used to tune performance of the MPC controller. The linear model in (10c) serves to predict the future states based on the knowledge of the initial state $x(t)$, which is assumed to be available at each time instance. The optimization is performed over the increments to provide offset-free tracking of the reference trajectory. The

state, output and input constraints represented, respectively, by the polyhedral sets X , Y , and U .

MPC is usually implemented in so-called receding horizon fashion. Here the optimal solution to the problem (10) is found for particular value of $x(t)$, which results into the optimal sequence. Here is the control action increment, used to introduce integral action. Out of this sequence, only the first element (i.e.) is actually implemented to the plant and the rest is discarded. At the next time instance, a new initial state measurement $x(t)$ is obtained and the whole procedure is repeated. This repetitive optimization is performed in order to introduce feedback control into the procedure and to deal with possible disturbances and plant model mismatches.

One can write new control action as $u_k = u_{k-1} + \Delta u_k$ which function of previous known control action and control action increment. The optimization problem (10) than can be rewritten into the state-space formulation with extended state feedback $[x_k, \bar{u}_{k-1}]^T$. Therefore, state-space representation reduces number of optimization variables from two (u_k and Δu_k) to one (Δu_k). The time-varying reference is constrained to be $y_{ref,k+1} = y_{ref,k}$, hence, state vector can be extended into three states $x_{k+1} = [x_k, u_{k-1}, y_{ref,k}]^T$ and state-space representation is defined in the form:

$$x_{k+1} = \begin{bmatrix} A & B & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \tilde{x}_k + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} \Delta u_k = \tilde{A}\tilde{x}_k + \tilde{B}\Delta u_k \quad (11a)$$

$$y_k = [C \quad D \quad 0] \tilde{x}_k + D\Delta u_k = \tilde{C}\tilde{x}_k + \tilde{D}\Delta u_k \quad (11b)$$

with all the matrices of appropriate dimensions.

If the initial state $x(t)$ and the value of the reference signal y_{ref} are known, the problem (10) can be solved as a quadratic problem (QP) for $p=2$ and for $p=1$ or $p=\infty$ as a linear program (LP). Even though efficient polynomial-time algorithms exist to solve both type of problems, the time needed to perform the optimization can be prohibitive if the sampling time is too short, or if the implementation hardware is very simple and thus less capable. To address this issue, in their seminal work (Bemporad et al. (2002)) have shown (for a quadratic type of performance induces) how to solve the problem (10) parametrically for all admissible initial conditions $x(t)$ by employing techniques of parametric programming. In this approach the optimal solution to (10) is found as an explicit state feedback law parametrized in the initial condition $x(t)$. The advantage of the parametric solutions is that the optimal control input can be obtained in real-time by simply evaluating a look-up table. The main result of the parametric approach is summarized by the following theorem.

Theorem 3.1. (Bemporad et al. (2002)). The optimal solution to the problem (10) is a piecewise affine function of the initial state x_0

$$\Delta u_0^* = F_r x_0 + g_r \quad \text{if } x_0 \in R_r \quad (12)$$

where $R_r = \{x_0 \mid H_r x_0 \leq K_r\}$ are polytopic regions of the state space, and F_r and g_r are the matrices of state-feedback law active in the r -th region.

Theorem 3.1 shows that the optimal solution for the problem (10) can be found as a look-up table consisting of r components. Therefore, once the table is calculated, MPC can be implemented in a real time by simply evaluating the table for the actual measurements of $x_0 := x(t)$. The table can be calculated efficiently using, e.g. the Multi-Parametric Toolbox [3]. Performance of the MPC scheme can be tuned by appropriately adjusting the weighting matrices Q and R , and by a suitable choice of prediction horizon N .

3 Implementation on PLC

As already mentioned, typical PLCs have severe memory limitations. Our PLC, in particular, only allows 1024 bytes of memory storage. A special care has thus to be taken when evaluating the explicit MPC feedback (encoded as a lookup table composed of feedbacks F_r , g_r , and regions R_r) on such a device. To perform this task efficiently, we employ the *binary search tree* (BST) algorithm.

The basic idea of BST algorithm is to hierarchically organize the controller regions into a tree structure where, at level of the tree, the number of regions to consider is decreased by a factor of two. Therefore the table traversal can be performed in time logarithmic in the number of regions. The tree is constructed in an iterative fashion. At each iteration an optimal separating hyperplane $h_i x(t) \leq k_i$ is selected such that the set of all regions processed at the i -th iteration is divided into two smaller subsets: regions R_i^+ residing on one side of the hyperplane and R_i^- on the other side. A new node in the tree is then created which contains information about the hyperplane and two pointers to child nodes. The left child is created by recursively calling the algorithm for regions R_i^+ and the right child for the regions R_i^- . The exploration of a given tree branch stops when no further subdivision can be achieved. In such a case a leaf is created which points to the region which contains $x(t)$. The resulting tree is then composed of the set of separating hyperplanes linked to the actual regions through a set of pointers.

To be able to use a BST-encoded tree on a PLC, the tree is transformed into a so-called "data-block". In this data table, first M entries represent one hyperplane and pointers to next line which should be explored. Obtaining the optimal control action for a particular value of x then reduces to traversing the binary search tree using Algorithm 1.

Algorithm 1 Table traversal via binary search tree

INPUT: BST tree composed of separating hyperplanes $h_i x \leq k_i, i=1, \dots, M$ and linked nodes, state measurements $x(t)$

OUTPUT: Optimal control input $u^*(x)$

- 1: $r \leftarrow 1$
- 2: repeat
- 3: if $h_r x \leq k_r$ then
- 4: $r \leftarrow$ index of the left child node (negative index)
- 5: else
- 6: $r \leftarrow$ index of the right child node (positive index)
- 7: end if
- 8: until r is a leaf node (positive index).
- 9: $u^*(x(t)) = F_r x(t) + g_r$

The PLC version of Algorithm 1, implemented using the Ladder Logic (LAD) programming language, is universal and can process any kind of lookup tables described by binary search trees. The LAD diagram consists of several routines and subroutines, a short excerpt of which is shown in Fig. 4. The program allocates 74 bytes of global memory in main routine and at most 34 bytes of temporary memory in subroutines. The total amount of memory allocated for controller is 874 bytes, the rest (150 bytes) remains to user.

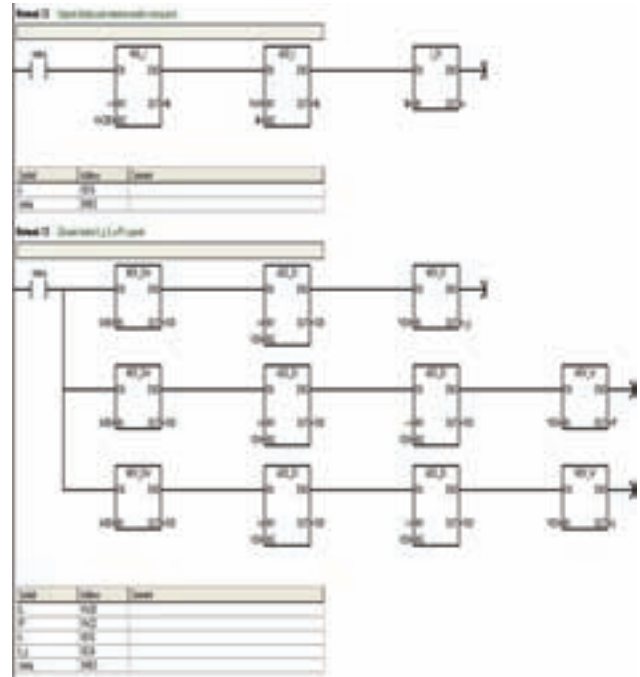


Fig.4 A short excerpt of the LAD implementation of Algorithm 1

4 Experimental Results

In this section we show how MPC could be used for control of the fan heater described in the Section 1 using PLC. The control objective is to drive the volume of air flow to a time varying reference while respecting motor capacity $0\% \leq f \leq 100\%$ and volume of the air flow $0\% \leq V \leq 100\%$. The following mathematical model of the fan heater was obtained using identification methods:

$$G = \frac{5.12}{0.4726s + 1} e^{-0.3s} \quad (13)$$

MPC synthesis using the Multi-Parametric Toolbox begins with a definition of the prediction model:

```
>> A=-0.1953; B=0.0923; C=1; D=0;
>> fan=ss(A,B,C,D)
>> Ts=0.25
>> model=mpt_sys(fan,Ts)
```

where the model is converted into the discrete-time domain using sampling time T_s . Time delay of the model can be omitted as it is less than the time constant and the real time verification proved such model to be satisfactory. Next, constraints on state, input and output are defined:

```
>> model.umax=100-fs; model.umin=0-fs;
>> model.xmax=100-Vs; model.xmin=0-Vs;
>> model.ymax=100-Vs; model.ymin=0-Vs;
```

Notice that the constraints are imposed on the deviation variables with linearization points $f^s = 15\%$ and $V_s = 30\%$.

Once the model is complete, parameters of the MPC problem to be solved could be defined by

```
>> problem.R=1; %penalty on u_k
>> problem.Q=1; %penalty on x_k
>> problem.Qy=1000; %penalty on (y_k-y_ref)
>> problem.N=5; %prediction horizon
>> problem.norm=2; %use quadratic cost
>> problem.tracking=1; %use time-varying reference
```

Values of the penalty matrices R and Q_y were chosen with respect to allowed number of regions (which reflect the

memory footprint of the controller). The number of regions can be reduced by lowering R and increasing the value of Q_y . The upper bound on the number of regions is 26 for a controller with 1 state and 1 input, otherwise the controller footprint would exceed 1 kB.

Finally, the parametric solution to problem (10) can be calculated as a lookup table using the command

```
>> ctrl=mpt_control(model,problem)
```

Result of the composition is, in this case, a lookup table consisting of 25 regions in a 3D state-space. State-space representation used in the controller consist of the following matrices

$$A = \begin{bmatrix} 0.8226 & 0.3199 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0.3199 \\ 0 \\ 0 \end{bmatrix} \quad (14a)$$

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (14b)$$

BST tree is constructed from the F_i lookup table using the MPT command

```
>> tree=mpt_searchTree(ctrl)
```

In our case, tree consists of 25 nodes in 7 levels, which corresponds to 724 bytes of memory. Selected parts of the data-block are depicted in Fig. 5. The number of values in one line corresponds with number of the state variables and constant.

The data are subsequently downloaded to the PLC, which then executes the table traversal at each sampling instance based on the measurements of the states. When the region for the actually measured state is found, Algorithm 1 is executed and the corresponding control input to the system is calculated as

$$u_k = u_{k-1} + \Delta u^* \quad (15)$$

with state-feedback law $\Delta u^* = F_i \tilde{x}_k + g_i$ where \tilde{x}_k is state variable of the system (11).

```
// h_i, k_i, I_positive (h*x >= k), I_negative (h*x < k)
WD2316 0.014993905868, 0.000000000000, 0.999887594072, 70.303400479304
WM232 -2, -3
WD236 -0.000172245543, -0.000023729104, 0.999999994884, 69.987020388607
WM252 -4, -5
WD256 -0.031289541095, 0.009555382228, -0.999510298050, 27.860553742954
WM272 -6, -7
WD276 -0.007172535481, 0.000094602731, 0.999974272562, 69.499794876778
WM292 -8, -9
WD296 -0.634115037629, 0.002361650321, 0.773235114089, 21.510222553989
WM312 -10, -11
WD316 0.077584962071, 0.008448289942, -0.996940948647, 24.734743662862
WM332 -12, -13
WD336 0.023194184544, 0.004509499517, -0.99972080135, 26.339590462389
WM352 -14, -15
WD356 -0.643222433078, 0.002393804802, 0.765675630593, 20.524844140814
WM372 13, 10
WD376 -0.000382795506, -0.000020170019, 0.999999992630, 69.972416000882
WM392 -16, 7
.
.
.
// F_i, G_i
WD716 -2.571702741900, -1.000000000000, -0.000000000000, 218.847880947400
WD732 -2.551297916900, -0.9960302813100, 3.877595031500, 20.925234301400
WD748 -2.548395788500, -0.995468575500, 3.630187328100, 14.321173817900
WD764 -2.542212686100, -0.994266433800, 3.963681899900, 7.300173971700
WD780 -2.504735063800, -0.990672097000, 3.075256534800, -1.470205824300
WD796 -2.504735062400, -0.990672097500, 3.075086170100, -1.521293174400
WD812 -2.504727020800, -0.990671583200, 3.054245316400, 0.000000000000
```

Fig. 5 A short excerpt of the data-block provided on the PLC.

The data-block representing the controller was downloaded to the PLC to perform real-time experiments. First, ability of the controller to follow a time-varying reference, where user

can change a setpoint at any time, is documented in Figs. 6-7.

System response near the and upper bound is without oscillations within a reasonable settling time, while response around the middle and near the lower bound has longer settling time. Such response can be caused by different behavior of the plant throughout the state ranges. That means, several models are necessary to describe plant behavior sufficiently. Therefore, possibility how to eliminate oscillations, is to control the plant as a hybrid system, which is not possible due to restricted amount of the memory.

To reduce long settling time, one can approximate time delay in model (13) by Taylor series or Padé approximation. This approach results in the better but more complex model with more regions, thus impossible to apply on PLC. The maximum amount of regions is function of the number of state variables. Therefore, if one wants more regions, either a simpler model is required or region reduction techniques have to be employed [4,2].

Real data presented in Figs. 6-7 show that the MPC controller utilizes the predictions to change the value of the input signal in the same period as the reference was changed, such that output signal is steered towards this reference. Experiment also proved that MPC controller with time-varying reference can be implemented on the PLC in a real time.

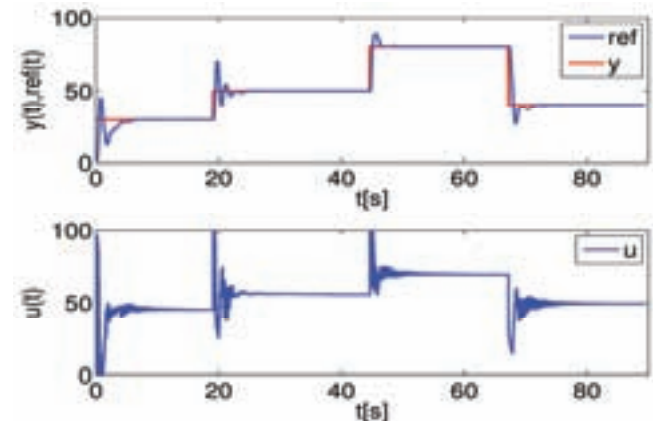


Fig.6 Control of the fan heater tracking time-varying reference.

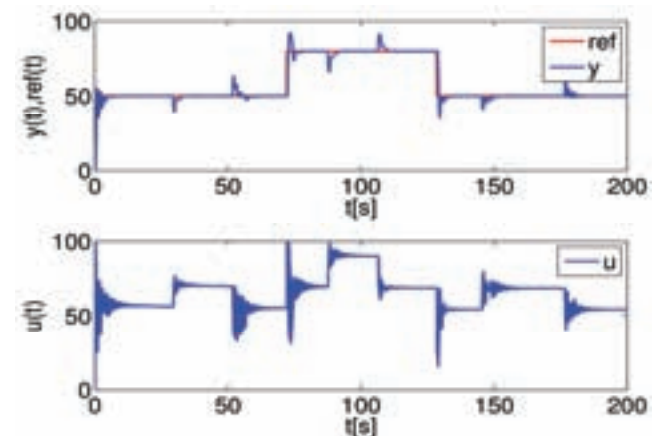


Fig.7 Control of the fan heater with disturbance during the tracking of time-varying reference.

5 Conclusions

In this paper we have shown how MPC can be implemented on a programmable logic controller with severe limitations on allowed memory storage. The approach was based on the pre-calculating the solution to the MPC optimization

problem just once, for all possible initial conditions. The result is then given in the form of a lookup table. Such a table was subsequently encoded as a binary search tree for its efficient evaluation in real time. Experimental results confirm that the controller provides satisfactory performance while respecting design constraints.

Acknowledgment

Authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0095/11, and the Slovak Research and Development Agency under the project APVV-0029-07. This work was also financed by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

References

- [1] Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.
- [2] Kvasnica, M. and Fikar, M. (2010). Performance-lossless complexity reduction in explicit MPC. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 5270–5275. Atlanta, USA.
- [3] Kvasnica, M., Grieder, P., and Baotic, M. (2004). Multi-Parametric Toolbox (MPT). Available from <http://control.ee.ethz.ch/mpt/>.

[4] Kvasnica, M., Löfberg, J., and Fikar, M. (2011). Stabilizing polynomial approximation of explicit MPC. *Automatica*. (accepted).

[5] Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice Hall.

[6] Siemens (2008). *S7-200 Programmable Controller System Manual*.

[7] Siemens (2010). *Betriebsanleitung Nr.:C98130-A7560-A2-6419*.

[8] Svetíková, M., Annus, J., Čírka, L., and Fikar, M. (2003). Real time control of a laboratory fan heater using dspace tools. In *Proceedings of the 14th International Conference Process Control '03*. Štrbské Pleso, High Tatras (Slovakia).

Ing. Ivana Rauová, Ing. Richard Valo, Ing. Michal Kvasnica PhD, and Prof. Ing. Miroslav Fikar, DrSc.

Slovak University of Technology in Bratislava,
Institute of Automation, Information Engineering and Mathematics,
Department of Information Engineering and Process Control
Radlinského 9
812 37 Bratislava, Slovakia,
ivana.rauova@stuba.sk,
richard.valo@stuba.sk
michal.kvasnica@stuba.sk
miroslav.fikar@stuba.sk
<http://www.kirp.chtf.stuba.sk>

PI controller design for laboratory process with uncertainties

Jana Závacká, Monika Bakošová

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 controller-parameter (k_p , k_i)-plane and the sixteen plant theorem. The stability boundaries obtained for sixteen Kharitonov plants split the controller-parameter plane in stable and unstable regions. The designed robust PI controllers which parameters are chosen from the stable region common for all sixteen plants 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: robust control, PI controller, interval uncertainty, process control

Introduction

Chemical reactors are ones of the most important plants in chemical industry, see e.g. [8]. 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], [7].

In this paper, a simple method for design of robust PI controllers is [9]. 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 [5].

The approach is used for design of a robust PI controller for a laboratory continuous stirred tank reactor, which can be modelled 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], [10]) 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 represent a chemical reactor with a stirrer and a cooling/heating coil.

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
Minimum operation volume	1 l
Maximum operation volume	2 l
Minimum vessel depth	0.054 m
Maximum vessel depth	0.108 m

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 peristaltic 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 $0.00175 \text{ dm}^3/\text{s}$, which represents the pump performance 40%.

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 0-30% 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 $0.00175 \text{ dm}^3/\text{s}$ of NaCl solution dosed into the reactor was assured by the peristaltic pump with performance 40% in all experiments. Fourteen various step changes of water flow rate were realized between $0.0032 \text{ dm}^3/\text{s} - 0.01145 \text{ dm}^3/\text{s}$, which represented the PSV opening 50 - 100%. The step responses were measured repeatedly. The resultant transfer function of the laboratory reactor was identified in the form of a transfer function (1) with the parametric interval uncertainty. The software LDDIF [6] was used for identification, which is based on the least squares algorithm. The values of the uncertain parameters are shown in Table 2.

$$G(s) = \frac{b_1s + b_0}{a_2s^2 + a_1s + a_0} \quad (1)$$

Parameter	Minimal value	Maximal value
b_1	0.0028	0.0428
b_0	-0.2776	-0.0156
a_2	1	1
a_1	0.6349	5.5024
a_0	0.2084	3.1351

Tab. 2 Uncertain parameters

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, [9], [11], [5]. 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 $N_i(s)$, $i=1, 2, 3, 4$ 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).

$$\begin{aligned} N_1(s) &= b_1^-s + b_0^- \\ N_2(s) &= b_1^+s + b_0^+ \\ N_3(s) &= b_1^+s + b_0^- \\ N_4(s) &= b_1^-s + b_0^+ \end{aligned} \quad (2)$$

$$\begin{aligned} D_1(s) &= a_2^- + a_1^- + a_0^+ \\ D_2(s) &= a_2^+ + a_1^+ + a_0^- \\ D_3(s) &= a_2^+ + a_1^- + a_0^- \\ D_4(s) &= a_2^- + a_1^+ + a_0^+ \end{aligned} \quad (3)$$

where b_k^- and b_k^+ , $k=0, 1$ are lower and upper bounds of the intervals of the numerator a_l^- and a_l^+ , $l=0, 1, 2$, are lower and upper bounds of intervals of the denominator parameters. 16 Kharitonov systems (4) can be obtained using polynomials (2), (3)

$$G_{ij}(s) = \frac{N_i(s)}{D_j(s)} \quad (4)$$

Substituting $s=j\omega$ into (4) and decomposing the numerator and the denominator polynomials of (4) into their even and odd parts one obtains

$$G_{ij}(j\omega) = \frac{N_{ie}(-\omega^2) + j\omega N_{io}(-\omega^2)}{D_{je}(-\omega^2) + j\omega D_{jo}(-\omega^2)} \quad (5)$$

The closed loop characteristic polynomial is as follows

$$\begin{aligned} \Delta(j\omega) &= [k_i N_{ie}(-\omega^2) - k_p \omega^2 N_{io}(-\omega^2) - \omega^2 D_{jo}(-\omega^2)] + \\ &+ j [k_p \omega N_{ie}(-\omega^2) + k_i \omega N_{io}(-\omega^2) + \omega D_{je}(-\omega^2)] \end{aligned} \quad (6)$$

Then, equating the real and imaginary parts of $\Delta(j\omega)$ to zero, one obtains

$$k_p(-\omega^2 N_{io}(-\omega^2)) + k_i(N_{ie}(-\omega^2)) = \omega^2 D_{jo}(-\omega^2) \quad (7)$$

and

$$k_p(N_{ie}(-\omega^2)) + k_i(N_{io}(-\omega^2)) = -D_{je}(-\omega^2) \quad (8)$$

After denoting

$$\begin{aligned} F_i(\omega) &= -\omega^2 N_{io}(-\omega^2) \\ G_i(\omega) &= N_{ie}(-\omega^2) \\ H_i(\omega) &= N_{ie}(-\omega^2) \\ I_i(\omega) &= N_{io}(-\omega^2) \\ J_j(\omega) &= \omega^2 D_{jo}(-\omega^2) \\ K_j(\omega) &= -D_{je}(-\omega^2) \end{aligned} \quad (9)$$

(7), (8) and (9) can be written as

$$\begin{aligned} k_p F_i(\omega) + k_i G_i(\omega) &= J_j(\omega) \\ k_p H_i(\omega) + k_i I_i(\omega) &= K_j(\omega) \end{aligned} \quad (10)$$

From these equations, parameters of the PI controller are expressed in the form

$$k_p = \frac{J_j(\omega)I_i(\omega) - K_j(\omega)G_i(\omega)}{F_i(\omega)I_i(\omega) - G_i(\omega)H_i(\omega)} \quad (11)$$

and

$$k_i = \frac{K_i(\omega)F_i(\omega) - J_j(\omega)H_i(\omega)}{F_i(\omega)I_i(\omega) - G_i(\omega)H_i(\omega)} \quad (12)$$

Consider one of the systems (4), where $i=2$ and $j=3$

$$G_{23}(s) = \frac{0.0428s - 0.156}{s^2 + 0.6349s + 0.2084} \quad (13)$$

Then

$$k_p = \frac{a_2^+ b_0^+ \omega^2 - a_0^- b_0^+ - a_1^- b_1^+ \omega^2}{(b_1^+)^2 \omega^2 + (b_0^+)^2} \quad (14)$$

$$k_i = \frac{a_1^- \omega^2 + k_p b_1^+ \omega^2}{b_0^+}$$

The stability boundary of the closed loop system (13) in the (k_p, k_i) -plane for $\omega=[0, 0.6267]$ is plot in the Figure 1. Then parameters k_p and k_i of the stabilizing controller are chosen from the stable region.

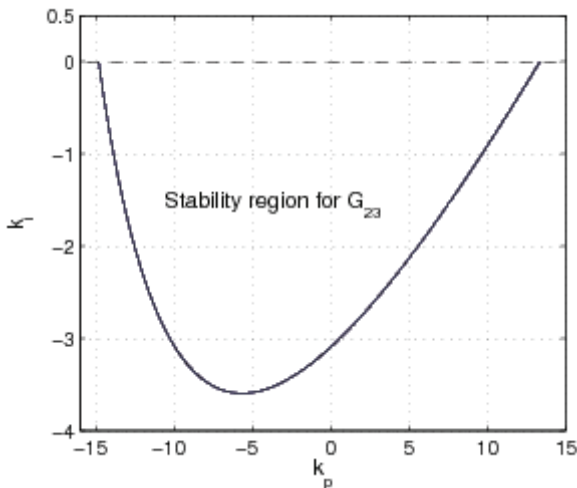


Fig.1 Stability region of parameters k_p, k_i for the system G_{23}

Stable regions for all 16 Kharitonov systems are obtained alike. In the Figure 2 are shown stable regions for 16 Kharitonov systems (4). The controller which stabilizes all 16 Kharitonov systems has to be found in the intersection of all stable regions (this intersection lies in the red rectangle), which is in detail displayed in the Figure 3.

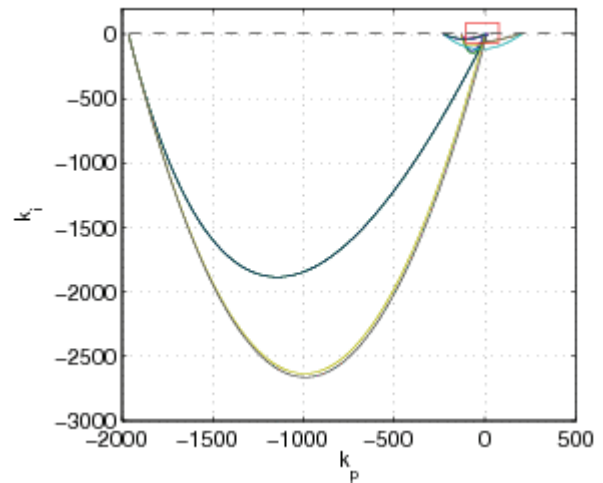


Fig.2 Stability regions for 16 Kharitonov plants

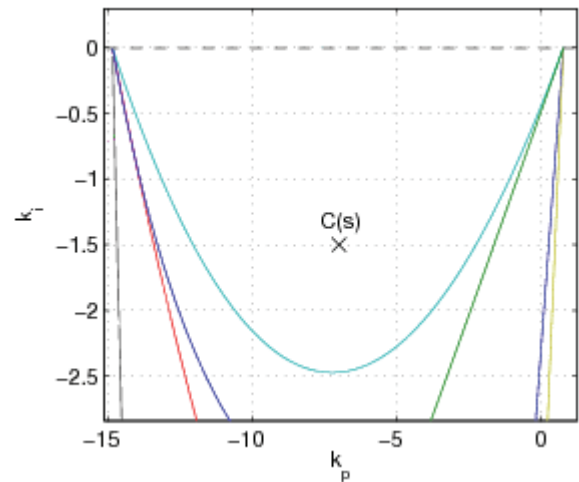


Fig.3 Detail of the stability region for 16 Kharitonov plants

The parameters of the robust PI controller for control of the laboratory reactor (15) were chosen from the stable region of parameters k_p, k_i according to simulation results obtained for various choices of PI controllers.

$$C(s) = \frac{k_p s + k_i}{s} = \frac{-7s - 1.5}{s} \quad (15)$$

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 [dm³/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 4 and 5. Robustness of the designed PI controller (15) was tested by setting the reference value in a wider area. Control responses of the reactor are shown in Figure 4 are for $\omega \in [12; 32]$ mS and in the Figure 5 for $\omega \in [18; 30]$ mS.

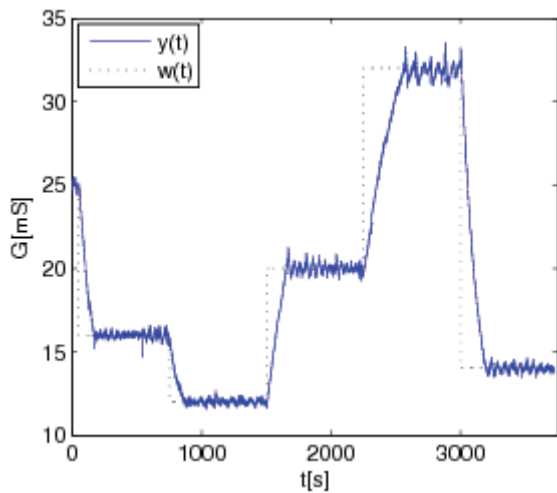


Fig.4 Control of the reactor with robust PI controller

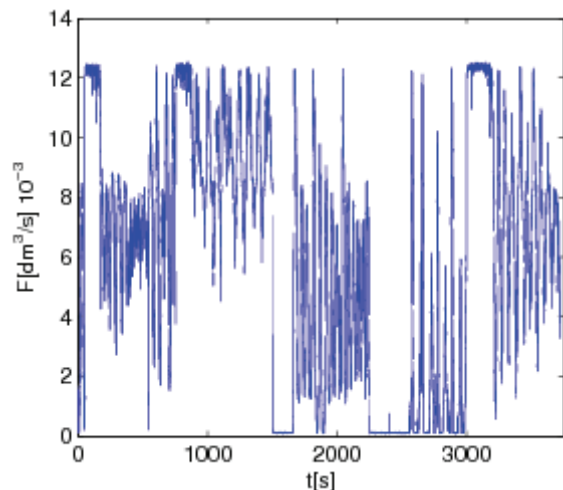


Fig.5 Control of the reactor with robust PI controller

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 and sixteen plant theorem. The robust 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 where controlled variable conductivity G [mS] of NaCl, was controlled by water flow rate F [dm³/s]. The varying reference was always reached. The control responses were without overshoots and fast enough.

Acknowledgements

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Republic under the grants 1/0537/10 and 1/0095/11.

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] Barmish, B: *New Tools for Robustness of Linear Systems*. Macmillan Publishing Company, 1994.
- [6] ČIRKA, L., FIKAR, M.: *A Dynamical System Identification Toolbox, Selected Topics in Modelling and Control*. STU Press, 58-62, Bratislava, 2007.
- [7] 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, Massachusetts, USA, July 5-7. CD-ROM 92, 2004.
- [8] MIKLEŠ, J., FIKAR, M: *Process modelling, identification and control II*. STU Press, Bratislava, 2004.

[9] TAN, N., KAYA, I.: Computation of stabilizing PI controllers for interval systems. In Mediterranean Conference on Control and Automation, Rhodes, Greece, June 2003.

[10] VOJTEŠEK, J., DOSTAL, J., MATUŠŮ: Multifunctional process control teaching system pct40. In 16th Int. Conferences Process Control, page 15f.pdf, Štrbské Pleso, Slovakia, 2007.

[11] 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, PhD.

Doc. Ing. Monika Bakošová, CSc.

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}@stuba.sk

- 2001** AT&P journal PLUS 1: Adaptívne a nelineárne riadenie systémov (tlačená verzia)
Adaptive and nonlinear control systems (printed version)
- AT&P journal PLUS 2: Robotika, mechatronika, diskrétné výrobné systémy (tlačená verzia)
Robotics, mechatronics, discrete manufacturing systems (printed version)
- 2002** AT&P journal PLUS 3: Robustné systémy riadenia (tlačená verzia)
Robust control systems (printed version)
- 2003** AT&P journal PLUS 4: Samonastavujúce sa systémy v riadení procesov (tlačená verzia)
Selftuning systems in process control (printed version)
- 2004** AT&P journal PLUS 5: Robotické systémy (elektronická – CD verzia)
Robotics systems (electronic – CD version)
- 2005** AT&P journal PLUS 6: Mechatronika (elektronická – CD verzia)
Mechatronics (electronic – CD version)
- AT&P journal PLUS 7: Umelá inteligencia v praxi (elektronická – CD verzia)
Artificial intelligence in Practise (electronic – CD version)
- 2006** AT&P journal PLUS 1: Mechatronické systémy (elektronická – CD verzia)
Mechatronic systems (electronic – CD version)
- AT&P journal PLUS 2: Inteligentné meracie systémy (elektronická – CD verzia)
Intelligent measurement systems (electronic – CD version)
- 2007** AT&P journal PLUS 1: MAMS'2007 (elektronická – CD verzia)
MAMS'2007 (electronic – CD version)
- AT&P journal PLUS 2: Riadenie procesov (elektronická – CD verzia)
Process Control (electronic – CD version)
- 2008** AT&P journal PLUS 1: Mobilné robotické systémy (elektronická – CD verzia)
Mobile robotic systems (electronic – CD version)
- AT&P journal PLUS 2: Riadenie v energetike (elektronická – CD verzia)
Control of Power Systems (electronic – CD version)
- 2009** AT&P journal PLUS 1: Inteligentné pohybové systémy (elektronická – CD verzia)
Intelligent motion control systems (electronic – CD version)
- AT&P journal PLUS 2: Riadenie procesov (elektronická – CD verzia)
Process control (electronic – CD version)
- 2010** AT&P journal PLUS 1: Systémy automatického riadenia (elektronická – CD verzia)
Systems of automatic control (electronic – CD version)
- AT&P journal PLUS 2: Robotika vo vzdelávaní (elektronická – CD verzia)
Robotics in education (electronic – CD version)
- 2011** ATP Journal PLUS 1: Systémy automatického riadenia (elektronická – CD verzia)
Systems of automatic control (electronic – CD version)