We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



1

Implementation of a New Quasi-Optimal Controller Tuning Algorithm for Time-Delay Systems

Libor Pekař and Roman Prokop
Tomas Bata University in Zlín
Czech Republic

1. Introduction

Systems and models with dead time or aftereffect, also called hereditary, anisochronic or time-delay systems (TDS), belonging to the class of infinite dimensional systems have been largely studied during last decades due to their interesting and important theoretical and practical features. A wide spectrum of systems in natural sciences, economics, pure informatics etc., both real-life and theoretical, is affected by delays which can have various forms; to name just a few the reader is referred e.g. to (Górecki et al., 1989; Marshall et al., 1992; Kolmanovskii & Myshkis, 1999; Richard, 2003; Michiels & Niculescu, 2008; Pekař et al., 2009) and references herein. Linear time-invariant dynamic systems with distributed or lumped delays (LTI-TDS) in a single-input single-output (SISO) case can be represented by a set of functional differential equations (Hale & Verduyn Lunel, 1993) or by the Laplace transfer function as a ratio of so-called quasipolynomials (El'sgol'ts & Norkin, 1973) in one complex variable s, rather than polynomials which are usual in system and control theory. Quasipolynomials are formed as linear combinations of products of s-powers and exponential terms. Hence, the Laplace transform of LTI-TDS is no longer rational and so-called meromorphic functions have to be introduced. A significant feature of LTI-TDS is (in contrast to undelayed systems) its infinite spectrum and transfer function poles decide - except some cases of distributed delays, see e.g. (Loiseau, 2000) - about the asymptotic stability as in the case of polynomials.

It is a well-known fact that delay can significantly deteriorate the quality of feedback control performance, namely stability and periodicity. Therefore, design a suitable control law for such systems is a challenging task solved by various techniques and approaches; a plentiful enumeration of them can be found e.g. in (Richard, 2003). Every controller design naturally requires and presumes a controlled plant model in an appropriate form. A huge set of approaches uses the Laplace transfer function; however, it is inconvenient to utilize a ratio of quasipolynomials especially while natural requirements of internal (impulse-free modes) and asymptotic stability of the feedback loop and the feasibility and causality of the controller are to be fulfilled.

The meromorphic description can be extended to the fractional description, to satisfy requirements above, so that quasipolynomials are factorized into proper and stable meromorphic functions. The ring of stable and proper quasipolynomial (RQ)

meromorphic functions ($\mathbf{R}_{MS}$) is hence introduced (Zítek & Kučera, 2003; Pekař & Prokop, 2010). Although the ring can be used for a description of even neutral systems (Hale & Verduyn Lunel, 1993), only systems with so-called retarded structure are considered as the admissible class of systems in this contribution. In contrast to many other algebraic approaches, the ring enables to handle systems with non-commensurate delays, i.e. it is not necessary that all system delays can be expressed as integer multiples of the smallest one. Algebraic control philosophy in this ring then exploits the Bézout identity, to obtain stable and proper controllers, along with the Youla-Kučera parameterization for reference tracking and disturbance rejection.

The closed-loop stability is given, as for delayless systems, by the solutions of the characteristic equation which contains a quasipolynomial instead of a polynomial. These infinite many solutions represent closed-loop system poles deciding about the control system stability. Since a controller can have a finite number of coefficients representing selectable parameters, these have to be set to distribute the infinite spectrum so that the closed-loop system is stable and that other control requirements are satisfied.

The aim of this chapter is to describe, demonstrate and implement a new quasi-optimal pole placement algorithm for SISO LTI-TDS based on the quasi-continuous pole shifting – the main idea of which was presented in (Michiels et al., 2002) - to the prescribed positions. The desired positions are obtained by overshoot analysis of the step response for a dominant pair of complex conjugate poles. A controller structure is initially calculated by algebraic controller design in $\mathbf{R}_{MS}$. Note that the maximum number of prescribed poles (including their multiplicities) equals the number of unknown parameters. If the prescribed roots locations can not be reached, the optimizing of an objective function involving the distance of shifting poles to the prescribed ones and the roots dominancy is utilized. The optimization is made via Self-Organizing Migration Algorithm (SOMA), see e.g. (Zelinka, 2004). Matlab m-file environment is utilized for the algorithm implementation and, consequently, results are tested in Simulink on an attractive example of unstable SISO LTI-TDS.

The chapter is organized as follows. In Section 2 a brief general description of LTI-TDS is introduced together with the coprime factorization for the $\mathbf{R}_{MS}$ ring representation. Basic ideas of algebraic controller design in $\mathbf{R}_{MS}$ with a simple control feedback are presented in Section 3. The main and original part of the chapter – pole-placement shifting based tuning algorithm – is described in Section 4. Section 5 focuses SOMA and its utilization when solving the tuning problem. An illustrative benchmark example is presented in Section 6.

2. Description of LTI-TDS

The aim of this section is to present possible models of LTI-TDS; first, that in time domain using functional differential equations, second, the transfer function (matrix) via the Laplace transform. Then, the latter concept is extended so that an algebraic description in a special ring is introduced. Note that for the further purpose of this chapter the state-space functional description is useless.

2.1 State-space model

A LTI-TDS system with both input-output and internal (state) delays, which can have point (lumped) or distributed form, can be expressed by a set of functional differential equations

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} &= \sum_{i=1}^{N_H}\mathbf{H}_i\frac{\mathrm{d}\mathbf{x}(t-\eta_i)}{\mathrm{d}t}+\mathbf{A}_0\mathbf{x}(t)+\sum_{i=1}^{N_A}\mathbf{A}_i\mathbf{x}(t-\eta_i)+\mathbf{B}_0\mathbf{u}(t)\\ &\quad+\sum_{i=1}^{N_B}\mathbf{B}_i\mathbf{u}(t-\eta_i)+\int_0^L\mathbf{A}(\tau)\mathbf{x}(t-\tau)\,d\tau+\int_0^L\mathbf{B}(\tau)\mathbf{u}(t-\tau)\,d\tau\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t)\end{aligned} \tag{1}$$

where $\mathbf{x}\in\mathbb{R}^n$ is a vector of state variables, $\mathbf{u}\in\mathbb{R}^m$ stands for a vector of inputs, $\mathbf{y}\in\mathbb{R}^l$ represents a vector of outputs, $\mathbf{A}_i$, $\mathbf{A}(\tau)$, $\mathbf{B}_i$, $\mathbf{B}(\tau)$, $\mathbf{C}$, $\mathbf{H}_i$ are matrices of appropriate dimensions, $0\le\eta_i\le L$ are lumped (point) delays and convolution integrals express distributed delays (Hale & Verduyn Lunel, 1993; Richard, 2003; Vyhlídal, 2003). If $\mathbf{H}_i\neq\mathbf{0}$ for any i = 1,2,...N_H, model (1) is called neutral; on the other hand, if $\mathbf{H}_i=\mathbf{0}$ for every i = 1,2,...N_H, so-called retarded model is obtained. It should be noted that the state of model (1) is given not only by a vector of state variables in the current time instant, but also by a segment of the last model history (in functional Banach space) of state and input variables

$$\mathbf{x}(t+\tau),\ \mathbf{u}(t+\tau),\ \tau\in\langle -L,0\rangle \tag{2}$$

Convolution integrals in (1) can be numerically approximate by summations for digital implementation; however, this can destabilize even a stable system. Alternatively, one can integrate (1) and add a new state variable to obtain derivations on the right-hand side only. In the contrary, the model can also be expressed in more consistent functional form using Riemann-Stieltjes integrals so that both lumped and distributed delays are under one convolution. For further details and other state-space TDS models the reader is referred to (Richard, 2003).

2.2 Input-output model

This contribution is concerned with retarded delayed systems in the input-output formulation governed by the Laplace transfer function matrix (considering zero initial conditions) as in (3). Hence, in the SISO case (we are concerning about here), the transfer function is no longer rational, as for conventional delayless systems, and a meromorphic function as a ratio of retarded quasipolynomials (RQ) is obtained instead.

$$\begin{aligned}\mathbf{Y}(s) &= \mathbf{G}(s)\mathbf{U}(s)=\frac{\mathbf{\Gamma}_1\mathbf{\Gamma}_2}{\Delta}\mathbf{U}(s)\\ \Delta &= \det\left[s\mathbf{I}-\mathbf{A}_0-\sum_{i=1}^{N_A+1}\mathbf{A}_i\exp(-s\eta_i)-\int_0^L\mathbf{A}(\tau)\exp(-s\tau)\,\mathrm{d}\tau\right]\\ \mathbf{\Gamma}_1 &= \mathbf{C}\mathrm{adj}\left[s\mathbf{I}-\mathbf{A}_0-\sum_{i=1}^{N_A+1}\mathbf{A}_i\exp(-s\eta_i)-\int_0^L\mathbf{A}(\tau)\exp(-s\tau)\,\mathrm{d}\tau\right]\\ \mathbf{\Gamma}_2 &= \mathbf{B}_0+\sum_{i=1}^{N_B+1}\mathbf{B}_i\exp(-s\eta_i)+\int_0^L\mathbf{B}(\tau)\exp(-s\tau)\,\mathrm{d}\tau\end{aligned} \tag{3}$$

A (retarded or neutral) quasipolynomial of degree n has the generic form

$$q(s) = s^n + \sum_{i=0}^{n}\sum_{j=1}^{h} x_{ij} s^i \exp\left(-\vartheta_{ij} s\right), \vartheta_{ij} \geq 0 \tag{4}$$

where $x_{nj} \neq 0$ in the neutral case for some j, whereas a RQ owns $x_{nj} = 0$ for all j.

However, the transfer function representation in the form of a ratio of two quasipolynomials is not suitable in order to satisfy controller feasibility, causality and closed-loop (Hurwitz) stability (Loiseau 2000; Zítek & Kučera, 2003). Rather more general approaches utilize a field of fractions where a transfer function is expressed as a ratio of two coprime elements of a suitable ring. A ring is a set closed for addition and multiplication, with a unit element for addition and multiplication and an inverse element for addition. This implies that division is not generally allowed.

2.3 Plant description in $\mathbf{R}_{MS}$ ring

A powerful algebraic tool ensuring requirements above is a ring of stable and proper RQ-meromorphic functions ($\mathbf{R}_{MS}$). Since the original definition of $\mathbf{R}_{MS}$ in (Zítek & Kučera, 2003) does not constitute a ring, some minor changes in the definition was made in (Pekař & Prokop, 2009). Namely, although the retarded structure of TDS is considered only, the minimal ring conditions require the use of neutral quasipolynomials at least in the numerator as well.

An element $T(s) \in \mathbf{R}_{MS}$ is represented by a proper ratio of two quasipolynomials

$$T(s) = \frac{y(s)}{x(s)} \tag{5}$$

where a denominator $x(s)$ is a quasipolynomial of degree n and a numerator can be factorized as

$$y(s) = \tilde{y}(s)\exp(-\tau s) \tag{6}$$

where $\tilde{y}(s)$ is a quasipolynomial of degree l and $\tau \geq 0$. $T(s)$ is stable, which means that there is no pole s_0 such that $\mathrm{Re}\{s_0\} \geq 0$; in other words, all roots of $x(s)$ with $\mathrm{Re}\{s_0\} \geq 0$ are those of $y(s)$. Moreover, the ratio is proper, i.e. $l \leq n$.

Thus, $T(s)$ is analytic and bounded in the open right half-plane, i.e.

$$\sup_{\mathrm{Re}\{s\}\geq 0} \left|T(s)\right| < \infty \tag{7}$$

As mentioned above, in this contribution only retarded systems are considered, i.e. $x(s)$, $y(s)$ are RQs. Let the plant be initially described as

$$G(s) = \frac{b(s)}{a(s)} \tag{8}$$

where $a(s)$, $b(s)$ are RQs. Hence, using a coprime factorization, a plant model has the form

$$G(s) = \frac{B(s)}{A(s)} \tag{10}$$

where $A(s), B(s) \in \mathbf{R}_{MS}$ are coprime, i.e. there does not exist a non-trivial (non-unit) common factor of both elements. Note that a system of neutral type can induce problem since there can exist a coprime pair $A(s), B(s)$ which is not, however, Bézout coprime – which implies that the system can not be stabilized by any feedback controller admitting the Laplace transform, see details in (Loiseau et al., 2002).

3. Controller design in RMS

This section outlines controller design based on the algebraic approach in the $\mathbf{R}_{MS}$ ring satisfying the inner Hurwitz (Bounded Input Bounded Output - BIBO) stability of the closed loop, controller feasibility, reference tracking and disturbance rejection.

For algebraic controller design in $\mathbf{R}_{MS}$ it is initially supposed that not only the plant is expressed by the transfer function over $\mathbf{R}_{MS}$ but a controller and all system signals are over the ring. As a control system, the common negative feedback loop as in Fig. 1 is chosen for the simplicity, where $W(s)$ is the Laplace transform of the reference signal, $D(s)$ stands for that of the load disturbance, $E(s)$ is transformed control error, $U_0(s)$ expresses the controller output (control action), $U(s)$ represents the plant input, and $Y(s)$ is the plant output controlled signal in the Laplace transform. The plant transfer function is depicted as $G(s)$, and $G_R(s)$ stands for a controller in the scheme.

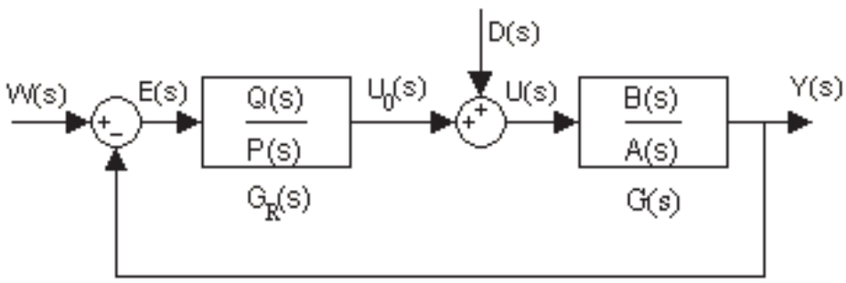


Fig. 1. Simple control feedback loop

Control system external inputs have forms

$$W(s) = \frac{H_W(s)}{F_W(s)},\ D(s) = \frac{H_D(s)}{F_D(s)} \tag{11}$$

where $H_W(s)$, $H_D(s)$, $F_W(s)$, $F_D(s) \in \mathbf{R}_{MS}$.

The following basic transfer functions can be derived in the control system in general

$$\begin{aligned} G_{WY}(s) &= \frac{Y(s)}{W(s)} = \frac{B(s)Q(s)}{M(s)},\ G_{DY}(s) = \frac{Y(s)}{D(s)} = \frac{B(s)P(s)}{M(s)} \\ G_{WE}(s) &= \frac{E(s)}{W(s)} = \frac{A(s)P(s)}{M(s)},\ G_{DE}(s) = \frac{E(s)}{D(s)} = -\frac{B(s)P(s)}{M(s)} \end{aligned} \tag{12}$$

where

$$G_R(s) = \frac{Q(s)}{P(s)} \tag{13}$$

$$M(s) = A(s)P(s) + B(s)Q(s) \tag{14}$$

and $Q(s)$, $P(s)$ are from $\mathbf{R}_{MS}$ and the fraction (13) is (Bézout) coprime (or relatively prime). The numerator of $M(s) \in \mathbf{R}_{MS}$ agrees to the characteristic quasipolynomial of the closed loop.
Following subsections describes briefly how to provide the basic control requirements.

3.1 Stabilization

According to e.g. (Kučera, 1993; Zítek & Kučera, 2003), the closed-loop system is stable if and only if there exists a pair $P(s), Q(s) \in \mathbf{R}_{MS}$ satisfying the Bézout identity

$$A(s)P(s) + B(s)Q(s) = 1 \tag{15}$$

a particular stabilizing solution of which, $P_0(s), Q_0(s)$, can be then parameterized as

$$\begin{aligned} P(s) &= P_0(s) \pm B(s)T(s) \\ Q(s) &= Q_0(s) \mp A(s)T(s), \; T(s) \in \mathbf{R}_{MS} \end{aligned} \tag{16}$$

Parameterization (16) is used to satisfy remaining control and performance requirements.

3.2 Reference tracking and disturbance rejection

The question is how to select $T(s) \in \mathbf{R}_{MS}$ in (16) so that tasks of reference tracking and disturbance rejection are accomplished. The key lies in the form of $G_{WE}(s)$ and $G_{DY}(s)$ in (12). Consider the limits

$$\begin{aligned} \lim_{t\to\infty} y_D(t) &= \lim_{s\to 0} sY_D(s) = \lim_{s\to 0} sG_{DY}(s)D(s) \\ &= \lim_{s\to 0} sB(s)P(s)\frac{H_D(s)}{F_D(s)} \end{aligned} \tag{17}$$

$$\begin{aligned} \lim_{t\to\infty} e_W(t) &= \lim_{s\to 0} sE_W(s) = \lim_{s\to 0} sG_{WE}(s)W(s) \\ &= \lim_{s\to 0} sA(s)P(s)\frac{H_W(s)}{F_W(s)} \end{aligned} \tag{18}$$

where $\cdot_D$ means that the output is influenced only by the disturbance, and symbol $\cdot_W$ expresses that the signal is a response to the reference. Limit (17) is zero if $\lim_{s\to 0} Y_D(s) < \infty$ and $Y_D(s)$ is analytic in the open right half-plane. Moreover, for the feasibility of $y_D(t)$, $Y_D(s)$ must be proper. This implies that the disturbance is asymptotically rejected if $Y_D(s) \in \mathbf{R}_{MS}$. Similarly, the reference is tracked if $E_W(s) \in \mathbf{R}_{MS}$.
In other words, $F_D(s)$ must divide the product $B(s)P(s)$ in $\mathbf{R}_{MS}$, and $A(s)P(s)$ must be divisible by $F_W(s)$ in $\mathbf{R}_{MS}$. Details about divisibility in $\mathbf{R}_{MS}$ can be found e.g. in (Pekař & Prokop, 2009). Thus, if neither $B(s)$ has any common unstable zero with $F_D(s)$ nor $A(s)$

has any common unstable zero with $F_W(s)$, one has to set all unstable zeros of $F_D(s)$ and $F_W(s)$ (with corresponding multiplicities) as zeros of $P(s)$. Note that zeros mean zero points of a whole term in $\mathbf{R}_{MS}$, not only of a quasipolynomial numerator. Unstable zeros agrees with those with $\mathrm{Re}\{s\} \geq 0$.

4. Pole-placement shifting based controller tuning algorithm

In this crucial section, the idea of a new pole-placement shifting based controller tuning algorithm (PPSA) is presented. Although some steps of PPSA are taken over some existing pole-shifting algorithms, the idea of connection with pole placement and the SOMA optimization is original.

4.1 Overview of PSSA

We first give an overview of all steps of PPSA and, consequently, describe each in more details. The procedure starts with controller design in $\mathbf{R}_{MS}$ introduced in the previous section. The next steps are as follows:

1. Calculate the closed-loop reference-to-output transfer function $G_{WY}(s)$. Let l_{num} and l_{den}, respectively, be numbers of unknown (free, selectable) real parameters of the numerator and denominator, respectively. Sign $l = l_{num} + l_{den}$.
2. Choose a simple model of a stable LTI system in the form of the transfer function $G_{WY,m}(s)$ with a numerator of degree n_{num} and the denominator of degree n_{den}. Calculate step response maximum overshoots of the model for a suitable range of its n_{num} zeros and n_{den} poles (including their multiplicities). If $\overline{n}_{num} \leq n_{num}$ and $\overline{n}_{den} \leq n_{den}$, respectively, are numbers of all real zeros (poles) and pairs of complex conjugate zeros (poles) of the model, it must hold that $\overline{n}_{num} \leq l_{num}$ and $\overline{n}_{den} \leq l_{den}$, respectively.
3. Prescribe all poles and zeros of the model with respect to calculated maximum overshoots (and maximal overshoot times). If the poles and zeros are dominant (i.e. the rightmost), the procedure is finished. Otherwise do following steps.
4. Shift the rightmost (or the nearest) zeros and poles to the prescribed locations successively. If the number of currently shifted poles and conjugate pairs $\overline{n}_{den} \leq n_{sp} \leq l_{den}$ is higher then $\overline{n}_{den}$, try to move the rest of dominant (rightmost) poles to the left. The same rule holds for shifted zeros, analogously.
5. If all prescribed poles and zeros are dominant, the procedure is finished. Otherwise, select a suitable cost function reflecting the distance of dominant poles (zeros) from prescribed positions and distances of spectral abscissas of both, prescribed and dominant poles (zeros).
6. Minimize the cost function, e.g. via SOMA.

Now look at these steps of the algorithm at great length.

4.2 Characteristic quasipolynomial and characteristic entire function

Algebraic controller design in the $\mathbf{R}_{MS}$ ring introduced in Section 3 results in a controller owning the transfer function $G_R(s)$ containing a finite number of unknown (free, selectable) parameters. The task of PPSA is to set these parameters so that the possibly infinite spectrum of the closed loop has dominant (rightmost) poles located in (or near by) the prescribed positions. If possibly, one can prescribe and place dominant zeros as well. Note

that controller design in $\mathbf{R}_{MS}$ using the feedback system as in Fig. 1 results in infinite spectrum of the feedback if the controlled plant is unstable.
If the (quasi)polynomial numerator and denominator of $G(s)$ have no common roots in the open right-half plane, the closed-loop spectrum is given entirely by roots of the numerator $m(s)$ of $M(s)$, the so called characteristic quasipolynomial. In the case of distributed delays, $G(s)$ has some common roots with $\mathrm{Re}\{s\} \geq 0$ in both, the numerator and denominator, and these roots do not affect the system dynamics since they cancel each other. In this case, the spectrum is given by zeros of the entire function $m(s)/m_U(s)$, i.e. the characteristic entire function, where $m_U(s)$ is a (quasi)polynomial the only roots of which are the common unstable roots.
The (quasi)polynomial denominator of $G_{WY}(s)$ agrees with $m(s)$. Its role is much more important than the role of the numerator of $G_{WY}(s)$ since the closed-loop zeros does not influence the stability. In the light of this fact, the setting of closed-loop poles has the priority. Therefore, one has to set l_{den} free denominator parameters first. Free (selectable) parameters in the numerator of $G_{WY}(s)$ are to be set only if there exist those which are not contained in the denominator. The number of such "additional" parameters is l_{num}.

4.3 Closed-loop model and step response overshoots

The task now is how to prescribe the closed-loop poles appropriately. We choose a simple finite-dimensional model of the reference-to-output transfer function and find its maximum overshoots and overshoot times for a suitable range of the model poles.
Let the prescribed (desired) closed-loop model be of the transfer function

$$G_{WY,m}(s) = k_1 \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} = k_2 \frac{s - z_1}{(s - s_1)(s - \bar{s}_1)} \tag{19}$$

where $k_1, k_2, b_1, b_0, a_1, a_0 \in \mathbb{R}$ are model parameters $z_1 \in \mathbb{R}^-$ stands for a model zero and $s_1 \in \mathbb{C}^-$ is a model stable pole where $\bar{s}_1$ expresses its complex conjugate. To obtain the unit static gain of $G_{WY,m}(s)$ it must hold true

$$k_1 = \frac{a_0}{b_0}, k_2 = -\frac{|s_1|^2}{z_1} \tag{20}$$

Sign $s_1 = \alpha + \omega \mathrm{j}, \alpha < 0, \omega \geq 0$ and calculate the impulse function $g_{WY,m}(t)$ of $G_{WY,m}(s)$ using the Matlab function *ilaplace* as

$$g_{WY,m}(t) = k_2 \exp(\alpha t)\left[\cos(\omega t) - \frac{z_1 - \alpha}{\omega}\sin(\omega t)\right] \tag{21}$$

Since $i_{WY,m}(t) = h'_{WY,m}(t)$, where $h_{WY,m}(t)$ is the step response function, the necessary condition for the existence of a step response overshoot at time t_O is

$$i_{WY,m}(t_O) = 0, t_O > 0 \tag{22}$$

The condition (22) yields these two solutions: either $t_O \to -\infty$ (which is trivial) or

$$t_O = \frac{1}{\omega}\arccos\left(\pm\frac{|\alpha - z_1|}{\sqrt{(\alpha - z_1)^2 + \omega^2}}\right) \tag{23}$$

when considering $\arccos(x) \in [0, \pi]$. Obviously, (23) has infinitely many solutions. If $\alpha < 0, z_1 < 0$, the maximum overshoot occurs at time

$$t_{\max} = \min(t_O) \tag{24}$$

One can further calculate the step response function $h_{WY,m}(t)$ as

$$h_{WY,m}(t) = \frac{k_2}{|s_1|^2}\left[\exp(\alpha t)\left(z_1\cos(\omega t) - \frac{z_1\alpha - |s_1|^2}{\omega}\sin(\omega t)\right) - z_1\right] \tag{25}$$

Define now the maximum relative overshoot as

$$\Delta h_{WY,m,\max} := \frac{h_{WY,m}(t_{\max}) - h_{WY,m}(\infty)}{h_{WY,m}(\infty)} \tag{26}$$

see Fig. 2.

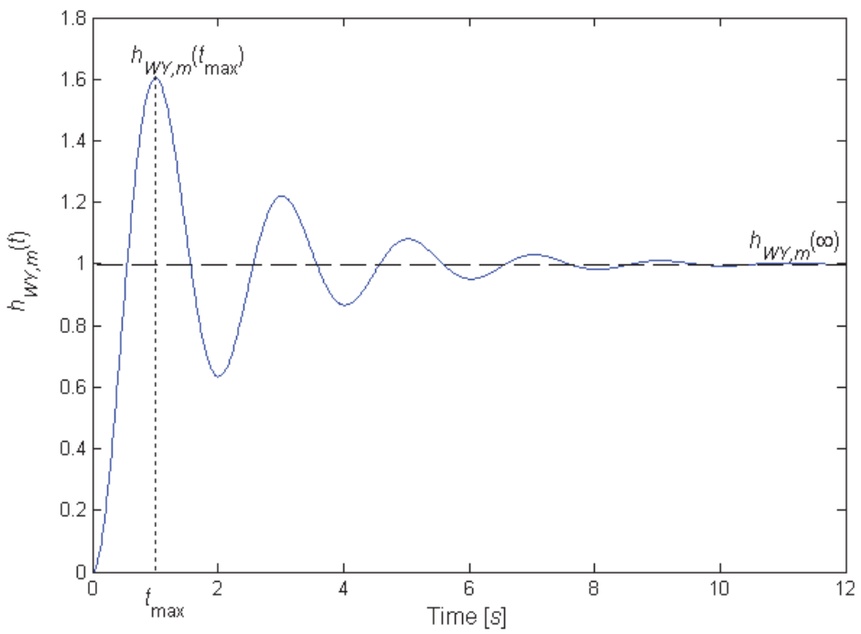


Fig. 2. Reference-to-output step response characteristics and the maximum overshoot

Using definition (26) one can obtain

$$\Delta h_{WY,m,\max} = \exp(\alpha t)\left(\frac{-z_1\omega\cos(\omega t) + \left(z_1\alpha - |s_1|^2\right)\sin(\omega t)}{z_1\omega}\right)\Bigg|_{t=t_{\max}} \tag{27}$$

Obviously, $\Delta h_{WY,m,\max}$ is a function of three parameters, i.e. n_1, α, ω, which is not suitable for a general formulation of the maximal overshoot. Hence, let us introduce new parameters ξ_α, ξ_z as

$$\xi_\alpha = -\frac{\alpha}{\omega},\ \xi_z = -\frac{z_1}{\omega} \tag{28}$$

which give rise from (23), (24) and (27) to

$$\begin{aligned} \Delta h_{WY,m,\max} &= \frac{1}{\xi_z}\exp\left(-\xi_\alpha t_{\max,norm}\right)\left(-\xi_z\cos\left(t_{\max,norm}\right)+\left(\xi_\alpha^2+1-\xi_\alpha\xi_z\right)\sin\left(t_{\max,norm}\right)\right) \\ t_{\max,norm} &= \omega t_{\max} = \min\left(\arccos\left(\pm\frac{\left|\xi_\alpha-\xi_\omega\right|}{\sqrt{\left(\xi_\alpha-\xi_\omega\right)^2+1}}\right)\right) \end{aligned} \tag{29}$$

where $t_{\max,norm}$ represents the normalized maximal overshoot time.
We can successfully use Matlab to display function $\Delta h_{WY,m,\max}(\xi_\alpha,\xi_z)$ and $t_{\max,norm}(\xi_\alpha,\xi_z)$ graphically, for suitable ranges of ξ_α, ξ_z as can be seen from Fig. 3 – Fig. 7.
Recall that model (19) gives rise to $n_{num}=1, n_{den}=2, \overline{n}_{num}=1, \overline{n}_{den}=1$.

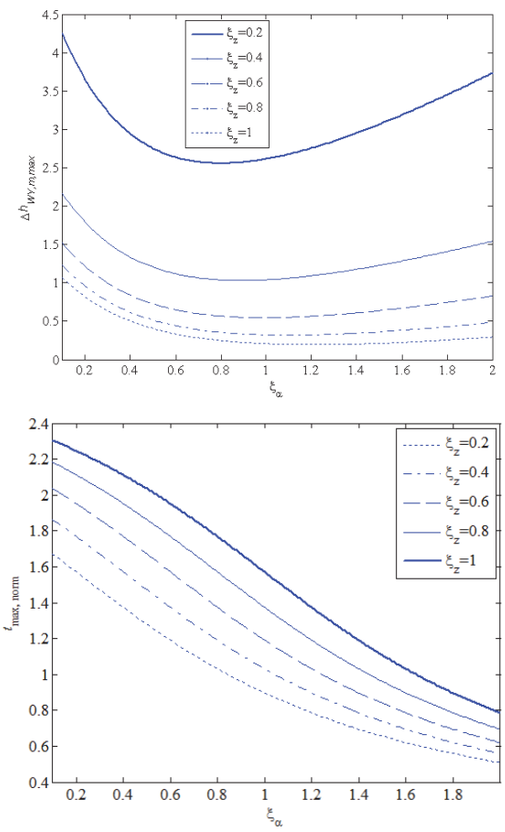


Fig. 3. Maximum overshoots $\Delta h_{WY,m,\max}(\xi_\alpha,\xi_z)$ (a) and normalized maximal overshoot times $t_{\max,norm}(\xi_\alpha,\xi_z)$ (b) for $\xi_\alpha=[0.1,2]$, $\xi_z=\{0.2,0.4,0.6,0.8,1\}$.

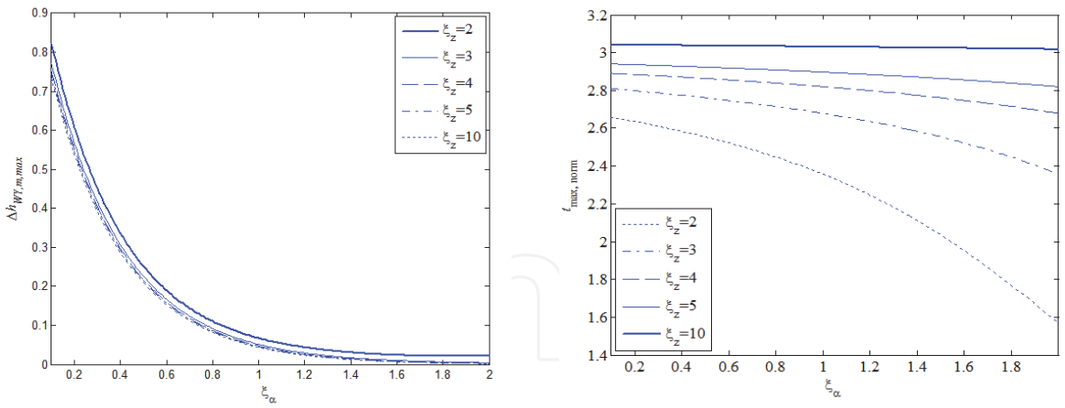


Fig. 4. Maximum overshoots $\Delta h_{WY,m,\max}(\xi_\alpha,\xi_z)$ (a) and normalized maximal overshoot times $t_{\max,norm}(\xi_\alpha,\xi_z)$ (b) for $\xi_\alpha=[0.1,2]$, $\xi_z=\{2,3,4,5,10\}$.

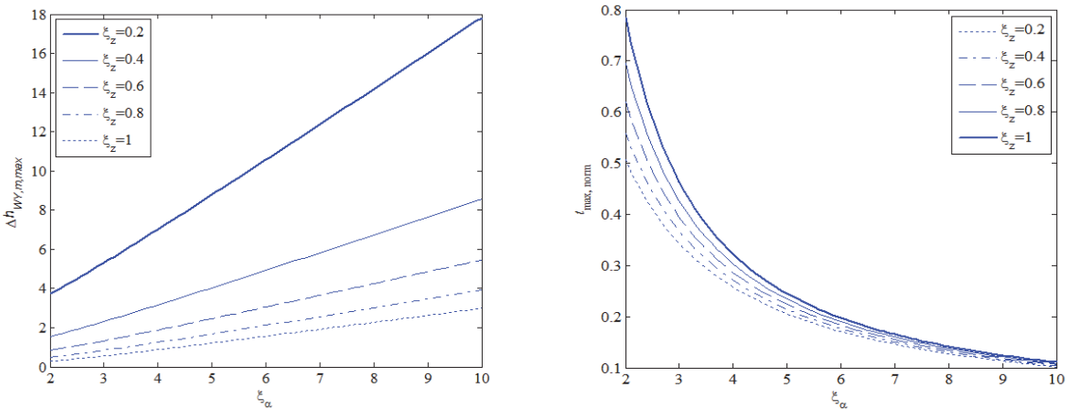


Fig. 5. Maximum overshoots $\Delta h_{WY,m,\max}(\xi_\alpha,\xi_z)$ (a) and normalized maximal overshoot times $t_{\max,norm}(\xi_\alpha,\xi_z)$ (b) for $\xi_\alpha=[2,10]$, $\xi_z=\{0.2,0.4,0.6,0.8,1\}$.

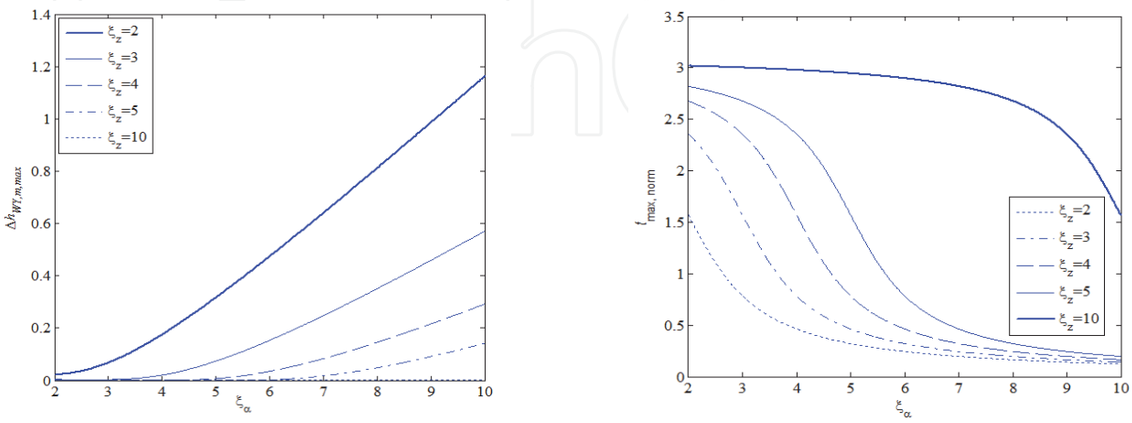


Fig. 6. Maximum overshoots $\Delta h_{WY,m,\max}(\xi_\alpha,\xi_z)$ (a) and normalized maximal overshoot times $t_{\max,norm}(\xi_\alpha,\xi_z)$ (b) for $\xi_\alpha=[2,10]$, $\xi_z=\{2,3,4,5,10\}$.

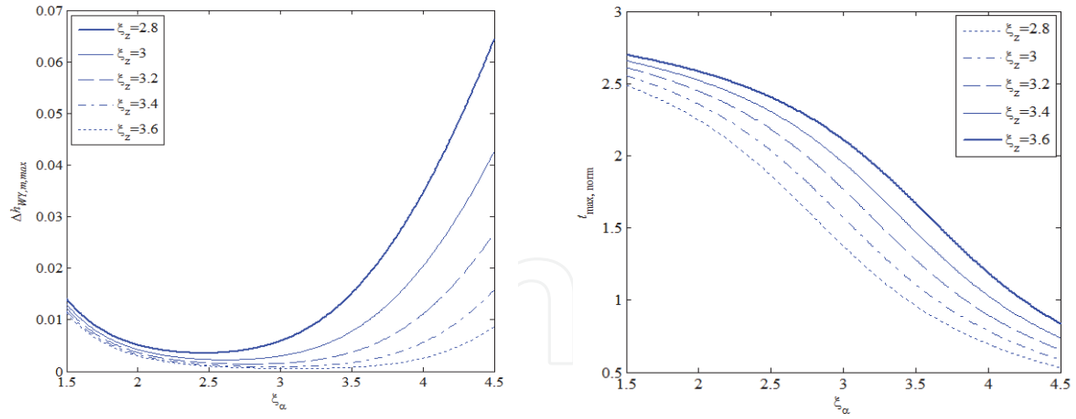


Fig. 7. Maximum overshoots $\Delta h_{WY,m,\max}(\xi_\alpha,\xi_z)$ (a) and normalized maximal overshoot times $t_{\max,norm}(\xi_\alpha,\xi_z)$ (b) for $\xi_\alpha=[1,5,4.5]$, $\xi_z=\{2.8,3,3.2,3.4,3.6\}$ - A detailed view on "small" overshoots.

The procedure of searching suitable prescribed poles can be done e.g. as in the following way. A user requires $\Delta h_{WY,m,\max}=0.03$ (i.e. the maximal overshoot equals 3 %), $\xi_\alpha=4$ (i.e. "the quarter dumping") and $t_{\max}=5\,\mathrm{s}$. Fig. 7 gives approximately $\xi_z=2.9$ which yields $t_{\max,norm}\approx 1.2$. These two values together with (28) and (29) result in $s_1=-0.96+0.24\mathrm{j}$, $z_1=-0.7$.

4.4 Direct pole placement

This subsection extends step 3 of PPSA from Subsection 4.1. The goal is to prescribe poles and zeros of the closed-loop "at once". The drawback here is that the prescribed poles (zeros) might not be dominant (i.e. the rightmost). The procedure was utilized to LTI-TDS e.g. in (Zítek & Hlava, 2001).

Given quasipolynomial $m(s)$ with a vector $\mathbf{v}=[v_1,v_2,...,v_l]^T\in\mathbb{R}^l$ of l free parameters, the assignment of n prescribed single roots σ_i, i = 1...n, can be done via the solution of the set of algebraic equations in the form

$$[m(\sigma_i,\mathbf{v})]_{s=\sigma_i}=0,\ i=1...n \tag{30}$$

In the case of complex conjugate poles, one has to take the real and imaginary part separately as

$$\mathrm{Re}\left\{[m(\sigma_i,\mathbf{v})]_{s=\sigma_i}\right\}=0,\mathrm{Im}\left\{[m(\sigma_i,\mathbf{v})]_{s=\sigma_i}\right\}=0 \tag{31}$$

for every pair of roots.

If a root σ_i has the multiplicity p, it must be calculated

$$\left[\frac{\mathrm{d}^j}{\mathrm{d}s^j}m(\sigma_i,\mathbf{v})\right]_{s=\sigma_i}=0,\ j=0...p-1 \tag{32}$$

or

$$\mathrm{Re}\left[\frac{\mathrm{d}^j}{\mathrm{d}s^j}m(\sigma_i,\mathbf{v})\right]_{s=\sigma_i}=0,\ \mathrm{Im}\left[\frac{\mathrm{d}^j}{\mathrm{d}s^j}m(\sigma_i,\mathbf{v})\right]_{s=\sigma_i}=0,\ j=0...p-1 \tag{33}$$

Note that if $m(s)$ is nonlinear with respect to $\mathbf{v}$, one can solve a set on non-linear algebraic equations directly, or to use an expansion

$$m(\sigma_i,\mathbf{v})\approx m(\sigma_i,\mathbf{v}_0)+\sum_{j=1}^{k}\Delta v_j\left[\frac{\partial m(s,\mathbf{v}_0)}{\partial v_j}\right]_{\substack{s=\sigma_i\\ \mathbf{v}=\mathbf{v}_0}} \tag{34}$$

where $\mathbf{v}_0$ means a point in which the expansion is made or an initial estimation of the solution and $\Delta\mathbf{v}=\left[\Delta v_1,\Delta v_2,...,\Delta v_l\right]^T$ is a vector of parameters increments. Equations (34) should be solved iteratively, e.g. via the well-known Newton method. Note, furthermore, that the algebraic controller design in $\mathbf{R}_{MS}$ for LTI-TDS results in the linear set (30)-(34) with respect to selectable parameters – both, in the numerator and denominator of $G_{WY}(s)$.
It is clear that a unique solution is obtained only if the set of $n=l$ independent equations is given. If $n<l$, equations (30)-(34) can be solved using the Moore-Penrose (pseudo)inverse minimizing the norm $\|\mathbf{v}\|_2=\sum_{i=1}^{k}v_i^2$, see (Ben Israel & Greville, 1966). Contrariwise, whenever $n>l$, it is not possible to place roots exactly and the pseudoinverse provides the minimization of squares of the left-hand sides of (30)-(34).
The methodology described in this subsection is utilized on both, the numerator and denominator.

4.5 Continuous poles (zeros) shifting

Once the poles (zeros) are prescribed, it ought to be checked whether these roots are the rightmost. If yes, the PPSA algorithm stops; if not, one may try to shift poles so that the prescribed ones become dominant. There are two possibilities. First, the dominant roots move to the prescribed ones; second, roots nearest to the prescribed ones are shifted – while the rest of the spectrum (or zeros) is simultaneously pushed to the left. The following describes it in more details.
We describe the procedure for the closed-loop denominator and its roots (poles); the numerator is served analogously for all its free parameters which are not included in the denominator. Recall that l_{den} is the number of unknown (selectable) parameters, n_{den} stands for the number of model (prescribed) poles (including their multiplicities), $\overline{n}_{den}$ represents the number of real poles and conjugate pairs of prescribed poles and n_{sp} is the number of currently shifted real poles and conjugate pairs. Generally, it holds that

$$\overline{n}_{den}\le n_{sp}\le l_{den} \tag{35}$$

The idea of continuous poles shifting described below was introduced in (Michiels et al., 2002). Similar procedure which, however, enables to shift less number of poles since $n_{sp}\le l_{den}$ includes every single complex pole instead of a conjugate pair, was investigated in (Vyhlídal, 2003). Roughly speaking, the latter is based on solution of (30) - (34) where $\mathbf{v}_0$ represents the vector of actual controller parameters, $\mathbf{v}=\mathbf{v}_0+\Delta\mathbf{v}$ are new controller

parameters and σ_i means prescribed poles (in the vicinity of the actual ones) here. Now look at the former methodology in more details.
The approach (Michiels et al., 2002) is based on the extrapolation

$$m(\sigma_i,\mathbf{v}) \approx \underbrace{m(\sigma_i,\mathbf{v}_0)}_{=0} + \Delta\sigma_i \left[\frac{\partial m(s,\mathbf{v})}{\partial s}\right]_{\substack{s=\sigma_i \\ \mathbf{v}=\mathbf{v}_0}} + \Delta v_j \left[\frac{\partial m(s,\mathbf{v})}{\partial v_j}\right]_{\substack{s=\sigma_i \\ \mathbf{v}=\mathbf{v}_0}} = 0,\ i=1...n_{sp},\ j=1...l_{den} \quad (36)$$

yielding

$$\frac{\Delta\sigma_i}{\Delta v_j} \approx -\left[\frac{\partial m(s,\mathbf{v})}{\partial s}\right]^{-1}_{\substack{s=\sigma_i \\ \mathbf{v}=\mathbf{v}_0}} \left[\frac{\partial m(s,\mathbf{v})}{\partial v_j}\right]_{\substack{s=\sigma_i \\ \mathbf{v}=\mathbf{v}_0}} \quad (37)$$

where $\mathbf{v}_0$ represents the vector of actual controller parameters, σ_i means actual poles and $\Delta\sigma_i$ and Δv_j are increments of poles and controller parameters, respectively. In case of a *p*-multiple pole, the following term is inserted in (36) and (37) instead of $m(s)$

$$\frac{\mathrm{d}^p}{\mathrm{d}s^p} m(s) \quad (38)$$

However, (38) can be used only if the pole including all multiplicities is moved. If, on the other hand, the intention is to shift a part of poles within the multiplicity to the one location and the rest of the multiplicity to another (or other) location(s), it is better to consider a multiple pole as a "nest" of close single poles.
Then a matrix

$$\mathbf{S} = \left[\frac{\Delta\sigma_i}{\Delta v_j}\right] \in \mathbb{R}\ n_{sp} x l_{den} \quad (39)$$

is called the sensitivity matrix satisfying

$$\Delta\mathbf{v} = \mathbf{S}^{+}\Delta\boldsymbol{\sigma} \quad (40)$$

where $\Delta\boldsymbol{\sigma} = \left[\Delta\sigma_1, \Delta\sigma_2, ..., \Delta\sigma_{n_{sp}}\right]^T$ and $\mathbf{S}^{+}$ means the pseudoinverse.
It holds that

$$\frac{\Delta \mathrm{Re}\{\sigma_i\}}{\Delta v_j} = \mathrm{Re}\left\{\frac{\Delta\sigma_i}{\Delta v_j}\right\} \quad (41)$$

thus, if poles are shifted in a real axis only, it can be calculated

$$\Delta\mathbf{v} = \mathrm{Re}\{\mathbf{S}\}^{+}\,\mathrm{Re}\{\Delta\boldsymbol{\sigma}\} \quad (42)$$

Otherwise, the following approximation ought to be used

$$\Delta\mathbf{v} \approx \mathrm{Re}\left\{\mathbf{S}^{+}\Delta\boldsymbol{\sigma}\right\} \quad (43)$$

The continuous shifting starts with $n_{sp} = \overline{n}_{den}$. Then, one can take the number of $\overline{n}_{den}$ rightmost poles and move them to the prescribed ones. The rightmost closed-loop pole moves to the rightmost prescribed pole etc. Alternatively, the same number of dominant poles (or conjugated pairs) can be considered; however, the nearest poles can be shifted to the prescribed ones. If two or more prescribed poles own the same dominant pole, it is assigned to the rightmost prescribed pole and removed from the list of moved poles. The number $n_{sp} \in \{\overline{n}_{den}, l_{den}\}$ is incremented whenever the approaching starts to fail for any pole. If $n_{sp} > \overline{n}_{den}$, the rest of dominant poles is pushed to the left. More precisely, shifting to the prescribed poles is described by the following formula

$$\Delta\sigma = \frac{\sigma_p - \sigma_s}{\left|\sigma_p - \sigma_s\right|}\delta \tag{44}$$

and pushing to the left agrees with

$$\Delta\sigma = -\delta \tag{45}$$

where δ is a discretization step in the space of poles, e.g. $\delta = 0.001$, σ_p is a prescribed pole and σ_s means a pole moved to the prescribed one.
If $n_{sp} = l_{den}$ and all prescribed poles become the rightmost (dominant) ones, PPSA is finished. Otherwise, do the last step of PPSA introduced in the following subsection.

5. Minimization of a cost function via SOMA

This step is implemented whenever the exact pole assignment even via shifting fails. In the first part of this subsection we arrange the cost function to be minimized. Then, SOMA algorithm (Zelinka, 2004) belonging to the wide family of evolution algorithms is introduced and briefly described. Again, the procedure is given for the pole-optimization; the zero-optimization dealing with the closed-loop numerator is done analogously.

5.1 Cost function

The goal now is to rearrange feedback poles (zeros) so that they are "sufficiently close" to the prescribed ones and, concurrently, they are "as the most dominant as possible". This requirement can be satisfied by the minimizing of the following cost function

$$F(\mathbf{v}) = d_\sigma(\mathbf{v}) + \lambda d_R(\mathbf{v}) = \sum_{i=1}^{\overline{n}_{den}} \left|\sigma_{s,i} - \sigma_{p,i}\right| + \lambda \,\mathrm{Re}\left\{\sigma_{d,i} - \sigma_{p,i}\right\} \tag{46}$$

where $d_\sigma(\mathbf{v})$ is the distance of prescribed poles $\sigma_{p,i}$ from the nearest ones $\sigma_{s,i}$, $d_R(\mathbf{v})$ expresses the sum of distances of dominant poles from the prescribed ones and $\lambda > 0$ represents a real weighting parameter. The higher λ is, the pole dominancy of is more important in $F(\mathbf{v})$. Recall that (when the dominant poles were moved)

$$\sigma_{s,1} \geq \sigma_{s,2} \geq \ldots \geq \sigma_{s,\overline{n}_{den}}, \sigma_{p,1} \geq \sigma_{p,2} \geq \ldots \geq \sigma_{p,\overline{n}_{den}}, \sigma_{d,1} \geq \sigma_{d,2} \geq \ldots \geq \sigma_{d,\overline{n}_{den}} \tag{47}$$

Alternatively, one can include both, the zeros and poles, in (46), not separately.

Poles can be found e.g. by the quasipolynomial mapping root finder (QPMR) implemented in Matlab, see (Vyhlídal & Zítek, 2003).
Hence, the aim is to solve the problem

$$\mathbf{v}_{opt} = \arg\min F(\mathbf{v}) \tag{48}$$

We use SOMA algorithm based on genetic operations with a population of found solutions and moving of population specimens to each other. A brief description of the algorithm follows.

5.2 SOMA

SOMA is ranked among evolution algorithms, more precisely genetic algorithms, dealing with populations similarly as differential evolution does. The algorithm is based on vector operations over the space of feasible solutions (parameters) in which the population is defined. Population specimens cooperate when searching the best solution (the minimum of the cost function) and, simultaneously, each of them tries to be a leader. They move to each other and the searching is finished when all specimens are localized on a small area.
In SOMA, every single generation, in which a new population is generated, is called a migration round. The notion of specific control and termination parameters, which have to be set before the algorithm starts, will be explained in every step of a migration round below.
First, population $P = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{PopSize}\}$ must be generated based on a prototypal specimen. For PPSA, this specimen is a vector of controller free parameters, $\mathbf{v}$, of dimension $D = l_{den}$. The prototypal specimen equals the best solution from Subsection 4.5. One can choose an initial radius (*Rad*) of the population in which other specimens are generated. The size of population (*PopSize*), i.e. the number of specimens in the population, is chosen by the user. Each specimen is then evaluated by the cost function (46).
The simplest strategy called "All to One" implemented here then selects the best specimen - leader, i.e. that with the minimal value of the cost function

$$\mathbf{v}_L^{mr} = \arg\min_i F\left(\mathbf{v}_i^{mr}\right) \tag{49}$$

where L denotes the leader, i is i-th of specimen in the population and mr means the current migration round. Then all other specimen are moved towards the leader during the migration round. The moving is given by three control parameters: *PathLength*, *Step*, *PRT*. *PathLength* should be within the interval $[1.1, 5]$ and it expresses the length of the path when approaching the leader. *PathLength* = 1 means that the specimen stops its moving exactly at the position of the leader. *Step* represents the sampling of the path and ought to be valued $[0.11, PathLength]$. E.g. a pair *PathLength* = 1 and *Step* = 0.2 agrees with that the specimen makes 5 steps until it reaches the leader. $PRT \in [0,1]$ enables to calculate the perturbation vector *PRTVector* which indicates whether the active specimen moves to the leader directly or not. *PRTVector* is defined as

$$\begin{aligned} &PRTVector = \left[p_1, p_2, ..., p_{l_{den}}\right]^T \in \{0,1\}^{l_{den}} \\ &p_i = 1 \quad \text{if } rnd_i < PRT \\ &p_i = 0 \quad \text{else} \end{aligned} \tag{50}$$

where $rnd_i \in [0,1]$ is a randomly generated number for each dimension of a specimen. Although authors of SOMA suggest to calculate *PRTVector* only once in migration round for every specimen, we try to do this in every step of the moving to the leader. Hence, the path is given by

$$\begin{aligned} &\mathbf{v}_{i,k+1}^{mr} = \mathbf{v}_{i,0}^{mr} + (i-1)Step\left(\mathbf{v}_L^{mr} - \mathbf{v}_{i,0}^{mr}\right) + \mathrm{diag}\left(PRTVector\right)Step\left(\mathbf{v}_L^{mr} - \mathbf{v}_{i,0}^{mr}\right) \\ &i = 1,2,...PopSize; k = 0,1,...round\left(PathLength\,/\,Step\right) - 1 \end{aligned} \tag{51}$$

where $\mathrm{diag}\left(PRTVector\right)$ means the diagonal square matrix with elements of *PRTVector* on the main diagonal and *k* is *k*-th step in the path.
If $PRTVector = [1,1,...1]^T$, the active specimen goes to the leader directly without "zig-zag" moves.
For every specimen of the population in a migration round, the cost function (i.e. value of the specimen) is calculated in every single step during the moving towards the leader. If the current position is better then the actual best, it becomes the best now. Hence, the new position of an active specimen for the next migration round is given by the best position of the specimen from all steps of moving towards the leader within the current migration round, i.e.

$$\mathbf{v}_i^{mr+1} = \arg\min_k F\left(\mathbf{v}_{i,k}^{mr}\right) \tag{52}$$

These specimens then generate the new population.
The number of migration round are given by user at the beginning of SOMA by parameter *Migration*, or the algorithm is terminated if

$$\max_i F\left(\mathbf{v}_i\right) - \min_i F\left(\mathbf{v}_i\right) < MinDiv \tag{53}$$

where *MinDiv* is the selected minimal diversity.
The final value $\mathbf{v}_{opt}$ is equal to $\mathbf{v}_L$ from the last migration round. We implemented the whole PPSA with SOMA in two Matlab m-files.

6. Illustrative example

In this closing session, we demonstrate the utilization of the PPSA and the methodology described above in Matlab on an attractive example.
Consider an unstable system describing roller skater on a swaying bow (Zítek et al., 2008) given by the transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b\exp\left(-(\tau+\vartheta)s\right)}{s^2\left(s^2 - a\exp\left(-\vartheta s\right)\right)} \tag{54}$$

see Fig. 8, where $y(t)$ is the skater's deviation from the desired position, $u(t)$ expresses the slope angle of a bow caused by force *P*, delays τ, ϑ mean the skater's and servo latencies and *b*, *a* are real parameters. Skater controls the servo driving by remote signals into servo electronics.
Let *b* = 0.2, *a* = 1, $\tau = 0.3$ s, $\vartheta = 0.1$ s, as in the literature, and design the controller structure according to the approach described in Section 3. Consider the reference and load disturbance in the form of a step-wise function.

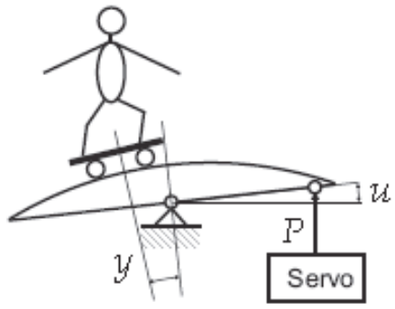


Fig. 8. The roller skater on a swaying bow

Hence, coprime factorization over $\mathbf{R}_{MS}$ can be done e.g. as

$$G(s)=\frac{B(s)}{A(s)}=\frac{\dfrac{b\exp\left(-(\tau+\vartheta)s\right)}{(s+m_0)^4}}{\dfrac{s^2\left(s^2-a\exp(-\vartheta s)\right)}{(s+m_0)^4}},W(s)=\frac{H_W(s)}{F_W(s)}=\frac{\dfrac{k_W}{s+m}}{\dfrac{s}{s+m}},\ D(s)=\frac{H_D(s)}{F_D(s)}=\frac{\dfrac{k_D}{s+m}}{\dfrac{s}{s+m}} \tag{55}$$

where $m_0>0$, k_W, $k_D \in \mathbb{R}$. Stabilization via the Bézout identity (15) results e.g. in the following particular solution

$$\begin{aligned} Q_0(s)&=\frac{\left(q_3s^3+q_2s^2+q_1s+q_0\right)(s+m_0)^4}{s^2\left(s^2-a\exp(-\vartheta s)\right)\left(s^3+p_2s^2+p_1s+p_0\right)+b\exp\left(-(\tau+\vartheta)s\right)\left(q_3s^3+q_2s^2+q_1s+q_0\right)} \\ P_0(s)&=\frac{\left(s^3+p_2s^2+p_1s+p_0\right)(s+m_0)^4}{s^2\left(s^2-a\exp(-\vartheta s)\right)\left(s^3+p_2s^2+p_1s+p_0\right)+b\exp\left(-(\tau+\vartheta)s\right)\left(q_3s^3+q_2s^2+q_1s+q_0\right)} \end{aligned} \tag{56}$$

using the generalized Euclidean algorithm, see (Pekař & Prokop, 2009), where p_2, p_1, p_0, q_3, q_2, q_1, $q_0 \in \circ$ are free parameters. In order to provide reference tracking and load disturbance rejection, use parameterization (16) while both, $F_W(s)$ and $F_D(s)$, divide $P(s)$; in other words, the numerator of $P(s)$ must satisfy $P(0)=0$. If we take

$$T=\frac{t_0(s+m_0)^4}{s^2\left(s^2-a\exp(-\vartheta s)\right)\left(s^3+p_2s^2+p_1s+p_0\right)+b\exp\left(-(\tau+\vartheta)s\right)\left(q_3s^3+q_2s^2+q_1s+q_0\right)} \tag{57}$$

$P(s)$ is obtained in a quite simple form with a real parameter t_0 which must be set as

$$t_0=\frac{-p_0m_0^4}{b} \tag{58}$$

Finally, the controller numerator and denominator in $\mathbf{R}_{MS}$, respectively, have forms

$$Q(s)=\frac{b\left(q_3s^3+q_2s^2+q_1s+q_0\right)(s+m_0)^4+p_0m_0^4s^2\left(s^2-a\exp(-\vartheta s)\right)}{b\left[s^2\left(s^2-a\exp(-\vartheta s)\right)\left(s^3+p_2s^2+p_1s+p_0\right)+b\exp\left(-(\tau+\vartheta)s\right)\left(q_3s^3+q_2s^2+q_1s+q_0\right)\right]}$$
$$P(s)=\frac{\left(s^3+p_2s^2+p_1s+p_0\right)(s+m_0)^4-p_0m_0^4\exp\left(-(\tau+\vartheta)s\right)}{s^2\left(s^2-a\exp(-\vartheta s)\right)\left(s^3+p_2s^2+p_1s+p_0\right)+b\exp\left(-(\tau+\vartheta)s\right)\left(q_3s^3+q_2s^2+q_1s+q_0\right)} \quad (59)$$

Hence, the controller has the transfer function

$$G_R(s)=\frac{b\left(q_3s^3+q_2s^2+q_1s+q_0\right)(s+m_0)^4+p_0m_0^4s^2\left(s^2-a\exp(-\vartheta s)\right)}{b\left[\left(s^3+p_2s^2+p_1s+p_0\right)(s+m_0)^4-p_0m_0^4\exp\left(-(\tau+\vartheta)s\right)\right]} \quad (60)$$

and the reference-to-output function reads

$$G_{WY}(s)$$
$$=\frac{b\left[b\left(q_3s^3+q_2s^2+q_1s+q_0\right)(s+m_0)^4+p_0m_0^4s^2\left(s^2-a\exp(-\vartheta s)\right)\right]\exp\left(-(\tau+\vartheta)s\right)}{(s+m_0)^4\left[s^2\left(s^2-a\exp(-\vartheta s)\right)\left(s^3+p_2s^2+p_1s+p_0\right)+b\exp\left(-(\tau+\vartheta)s\right)\left(q_3s^3+q_2s^2+q_1s+q_0\right)\right]} \quad (61)$$

which gives rise to the characteristic quasipolynomial

$$m(s)$$
$$=(s+m_0)^4\left[s^2\left(s^2-a\exp(-\vartheta s)\right)\left(s^3+p_2s^2+p_1s+p_0\right)+b\exp\left(-(\tau+\vartheta)s\right)\left(q_3s^3+q_2s^2+q_1s+q_0\right)\right] \quad (62)$$

Obviously, the numerator of $G_{WY}(s)$ does not have any free parameter not included in the denominator, i.e. l_{num} = 0. Moreover, the factor $(s+m_0)^4$ has a quadruple real pole; to cancel it, it must hold that $m_0 >> -\mathrm{Re}\{s_1\}=-\alpha$. Hence l_{den} = 7. Now, there are two possibilities – either set zero exactly to obtain constrained controller parameter (then l_{den} = 6) or to deal with the numerator and denominator of (61) together in (46) – we decided to utilize the former one. Generally, one can obtain e.g.

$$p_0=-\frac{b(z_1+m_0)^4\left(q_3z_1^3+q_2z_1^2+q_1z_1+q_0\right)}{m_0^4z_1^2\left(z_1^2-a\exp(-\vartheta z_1)\right)} \quad (63)$$

from (61).

Choose $\Delta h_{WY,m,\max}=0.5$, $\xi_\alpha=0.5$ and $t_{\max}=10$ s. From Fig. 3 we have $\xi_z=0.9$, $t_{\max,norm}\approx 2$ which gives $\omega=0.2, z_1=-0.18, \alpha=-0.1$. Then take e.g. $m_0=5$. Inserting plant parameters in (63) yields

$$p_0=5.4078\left(q_0-0.18q_1+0.0324q_2-0.005832q_3\right) \quad (64)$$

The concrete quasipolynomial which roots are being set, thus, reads

$$m_1(s)=s^2\left(s^2-\exp(-0.1s)\right)\left(s^3+p_2s^2+p_1s+5.4078\left(q_0-0.18q_1+0.0324q_2-0.005832q_3\right)\right)$$
$$+0.2\exp(-0.4s)\left(q_3s^3+q_2s^2+q_1s+q_0\right) \quad (65)$$

Initial direct pole placement results in controller parameters as

$$q_3 = 1.0051,\ q_2 = 0.9506,\ q_1 = 1.2582, q_0 = 0.2127, p_2 = 1.1179, p_1 = 0.4418, p_0 = 0.0603 \quad (66)$$

and poles locations in the vicinity of the origin are displayed in Fig. 9.

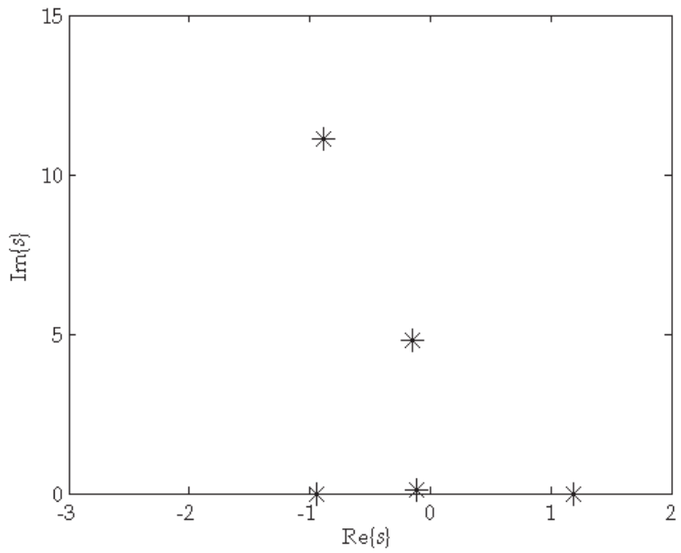


Fig. 9. Initial poles locations

The process of continuous roots shifting is described by the evolution of controller parameters, the spectral abscissa (i.e. the real part of the rightmost pole $\sigma_{d,1}$) and the distance of the dominant pole from the prescribed one $\left|\sigma_{d,1} - \sigma_{p,1}\right|$, as can be seen in Fig. 10 – Fig. 12, respectively. Note that p_0 is related to shifted parameters according to (64).

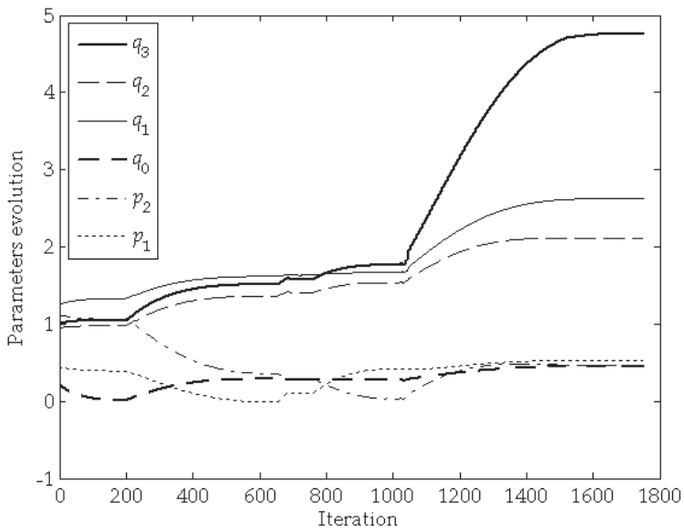


Fig. 10. Shifted parameters evolution

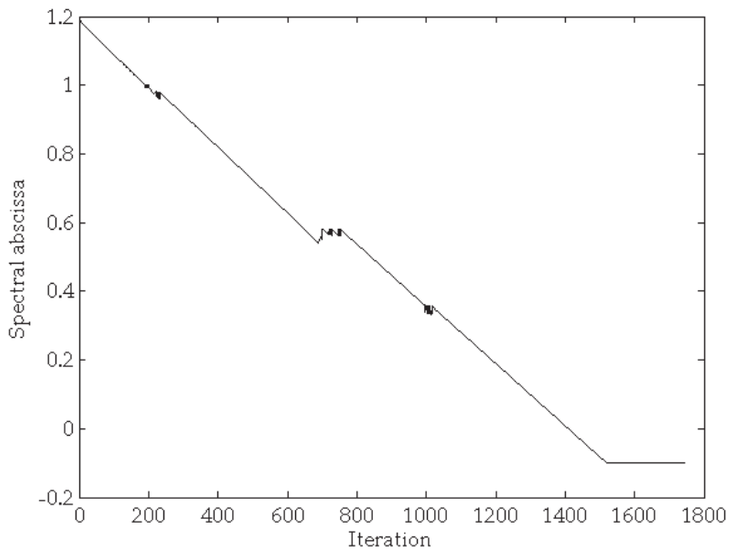


Fig. 11. Spectral abscissa evolution

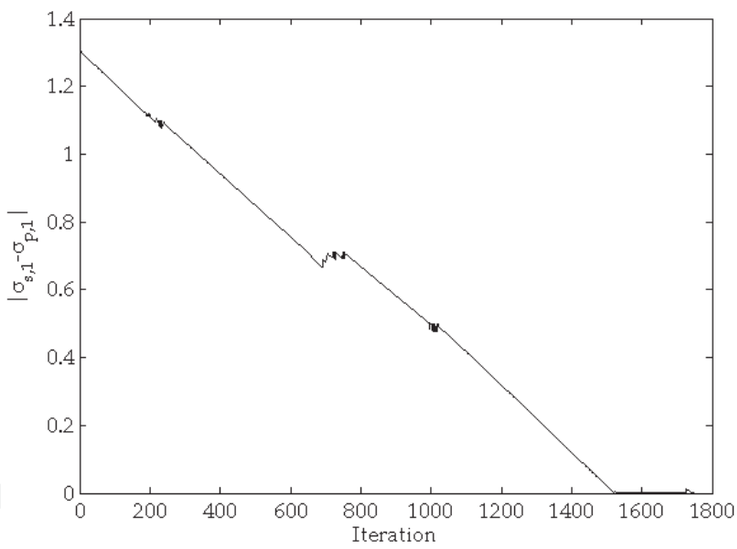


Fig. 12. Distance of the rightmost pole from the prescribed one

When shifting, it is suggested to continue in doing this even if the desired poles locations are reached since one can obtain a better poles distribution – i.e. non-dominant poles are placed more left in the complex space. Moreover, one can decrease the number of shifted poles during the algorithm whenever the real part of a shifted pole becomes "too different" from a group of currently moved poles.
The final controller parameters from the continuous shifting are

$$q_3 = 4.7587,\ q_2 = 2.1164,\ q_1 = 2.6252, q_0 = 0.4482, p_2 = 0.4636, p_1 = 0.529, p_0 = 4.6164 \quad (67)$$

and the poles location is pictured in Fig. 13.

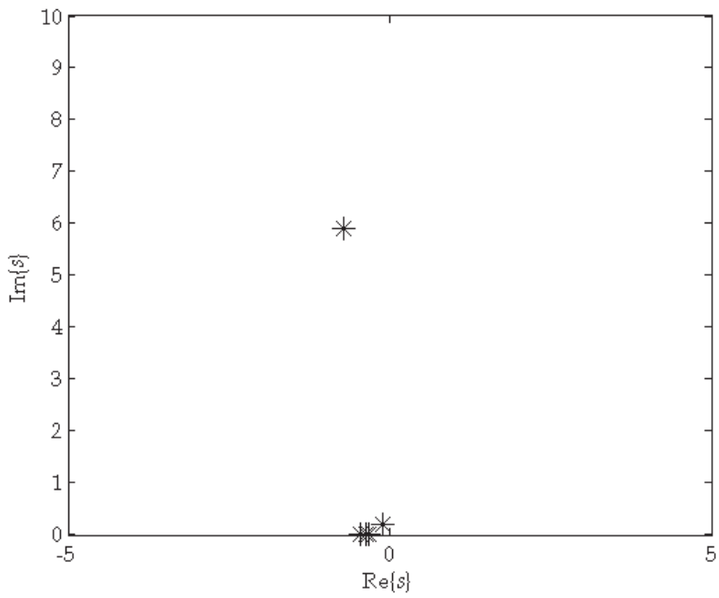


Fig. 13. Final poles locations

As can be seen, the desired prescribed pole is reached and it is also the dominant one. Thus, optimization can be omitted. However, try to perform SOMA to find the minimal cost function (16) with this setting: *Rad* = 2, *PopSize* = 10, *D* = 6, *PathLength* = 3, *Step* = 0.21, *PRT* = 0.6, *Migration* = 10, *MinDiv* = 10^{-6}, . Yet, the minimum of the cost function remains in the best solution from continuous shifting, i.e. according to (67), with the value of the cost function as $F(\mathbf{v}) = 2.93 \cdot 10^{-4}$.

7. Conclusion

This chapter has introduced a novel controller design approach for SISO LTI-TDS based on algebraic approach followed by pole-placement-like controller tuning and an optimization procedure. The methodology has been implemented in Matlab-Simulink environment to verify the results.

The initial controller structure design has been made over the ring of stable and proper meromorphic functions, $\mathbf{R}_{MS}$, which offers to satisfy properness of the controller, reference tracking and rejection of the load disturbance (of a nominal model). The obtained controller has owned some free (unset) parameters which must have been set properly.

In the crucial part of the work, we have chosen a simple finite-dimension model, calculated its step-response maximum overshoots and times to the overshoots. Then, using a static pole placement followed by continuous pole shifting dominant poles have been attempted to be placed to the desired prescribed positions.

Finally, optimization of distances of dominant (the rightmost) poles from the prescribed ones has been utilized via SOMA algorithm. The whole methodology has been tested on an attractive example of a skater on a swaying bow described by an unstable LTI TDS model.

The procedure is similar to the algorithm introduced in (Michiels et al., 2010); however, there are some substantial differences between them. Firstly, the presented approach is made in input-output space of meromorphic Laplace transfer functions, whereas the one in (Michiels et al., 2010) deals purely with state space. Second, in the cited literature, a number

poles less then a number of free controller parameters is set exactly and the rest of the spectrum is pushed to the left as much as possible. If it is possible it is necessary to choose other prescribed poles. We initially place the poles exactly; however, they can leave their positions during the shifting. Anyway, our algorithm does not require reset of selection assigned poles. Moreover, we suggest unambiguously how the prescribed poles (and zeros) positions are to be chosen – based on model overshoots. Last but not least, in (Michiels et al., 2010), the gradient sampling algorithm (Burke et al., 2005) on the spectral abscissa was used while SOMA together with more complex cost function is considered in this chapter.

The presented approach is limited to retarded SISO LTI-TDS without distributed delays only. Its extension to neutral systems requires some additional conditions on stability and existence of a stabilizing controller. Systems with distributed delays can be served in similar way as it is done here, yet with the characteristic meromorphic function instead of quasipolynomial. Multivariable systems would require deeper theoretic analysis of the controller structure design. The methodology is also time-comsupting and thus useless for online controller design (e.g. for selftuners).

In the future research, one can solve the problems specified above, choose other reference-to-output models and control system structures. There is a space to improve and modify the optimization algorithm.

8. Acknowledgment

The authors kindly appreciate the financial support which was provided by the Ministry of Education, Youth and Sports of the Czech Republic, in the grant No. MSM 708 835 2102 and by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

9. References

Ben-Israel, A. & Greville, T.N.E (1966). *Generalized Inverses: Theory and Applications*. Wiley, New York

Burke, J.; Lewis, M & Overton, M. (2005). A Robust Gradient Sampling Algorithm for Nonsmooth Nonconvex Optimization. *SIAM Journal of Optimization*, Vol. 15, Issue 3, pp. 751-779, ISSN 1052-6234

El'sgol'ts, L.E. & Norkin, S.B. (1973). *Introduction to the Theory and Application of Differential Equations with Deviated Arguments*. Academic Press, New York

Górecki, H.; Fuksa, S.; Grabowski, P. & Korytowski, A. (1989). *Analysis and Synthesis of Time Delay Systems*, John Wiley & Sons, ISBN 978-047-1276-22-7, New York

Hale, J.K., & Verduyn Lunel, S.M. (1993). Introduction to Functional Differential Equations. *Applied Mathematical Sciences*, Vol. 99, Springer-Verlag, ISBN 978-038-7940-76-2, New York

Kolmanovskii, V.B. & Myshkis, A. (1999). *Introduction to the Theory and Applications of Functional Differential Equations*, Cluwer Academy, ISBN 978-0792355045, Dordrecht, Netherlands

Kučera, V. (1993). Diophantine equations in control - a survey. Automatica, Vol. 29, No. 6, pp. 1361-1375, ISSN 0005-1098

Marshall, J.E.; Górecki, H.; Korytowski, A. & Walton, K. (1992). *Time Delay Systems, Stability and Performance Criteria with Applications*. Ellis Horwood, ISBN 0-13-465923-6, New York

Michiels, W.; Engelborghs, K; Vansevenant, P. & Roose, D. (2002). Continuous Pole Placement for Delay Equations. *Automatica*, Vol. 38, No. 6, pp. 747-761, ISSN 0005-1098

Michiels, W.; Vyhlídal, T. & Zítek, P. (2010). Control Design for Time-Delay Systems Based on Quasi-Direct Pole Placement. *Journal of Process Control*, Vol. 20, No. 3, pp. 337-343, 2010.

Loiseau, J.-J. (2000). Algebraic Tools for the Control and Stabilization of Time-Delay Systems. *Annual Reviews in Control*, Vol. 24, pp. 135-149, ISSN 1367-5788

Loiseau, J.-J. (2002). Neutral-Type Time-Delay Systems that are not Formally Stable are not BIBO Stabilizable. *IMA Journal of Mathematical Control and Information*, Vol. 19, No. 1-2, pp. 217-227, ISSN 0265-0754

Michiels, W. & Niculescu, S. (2008). Stability and Stabilization of Time Delay Systems: An eigenvalue based approach. *Advances in Design and Control*, SIAM, ISBN 978-089-8716-32-0, Philadelphia

Pekař, L. & Prokop, R. (2009). Some observations about the RMS ring for delayed systems, Proceedings of the 17th International Conference on Process Control '09, pp. 28-36, ISBN 978-80-227-3081-5, Štrbské Pleso, Slovakia, June 9-12, 2009.

Pekař, L.; Prokop, R. & Dostálek, P. (2009). Circuit Heating Plant Model with Internal Delays. *WSEAS Transaction on Systems*, Vol. 8, Issue 9, pp. 1093-1104, ISSN 1109-2777

Pekař, L. & Prokop, R. (2010). Control design for stable systems with both input-output and internal delays by algebraic means, *Proceedings of the 29th IASTED International Conference Modelling, Identification and Control (MIC 2010)*, pp. 400-407, ISSN 1025-8973, Innsbruck, Austria, February 15-17, 2010

Richard, J.P. (2003). Time-delay Systems: an Overview of Some Recent Advances and Open Problems. *Automatica*, Vol. 39, Issue 10, pp. 1667-1694, ISSN 0005-1098

Vyhlídal, T. (2003). *Analysis and Synthesis of Time Delay System Spectrum*, Ph.D. Thesis, Faculty of Mechanical Engineering, Czech Technical University in Prague, Prague

Vyhlídal, T. & Zítek, P. (2003). Quasipolynomial mapping based rootfinder for analysis of time delay systems, *Proceedings IFAC Workshop on Time-Delay systems, TDS'03*, pp. 227-232, Rocquencourt, France 2003.

Zelinka, I. (2004). SOMA-Self Organizing Migrating Algorithm. In: *New Optimization Techniques in Engineering*, G. Onwubolu, B.V.Babu (Eds.), 51 p., Springer-Verlag, ISBN 3-540-20167X, Berlin

Zítek, P. & Kučera, V. (2003). Algebraic Design of Anisochronic Controllers for Time Delay Systems. *International Journal of Control*, Vol. 76, Issue 16, pp. 905-921, ISSN 0020-7179

Zítek, P. & Hlava, J. (2001). Anisochronic Internal Model Control of Time-Delay Systems. *Control Engineering Practice*, Vol. 9, No. 5, pp. 501-516, ISSN 0967-0661

Zítek, P.; Kučera, V. & Vyhlídal, T. (2008). Meromorphic Observer-based Pole Assignment in Time Delay Systems. *Kybernetika*, Vol. 44, No. 5, pp. 633-648, ISSN 0023-5954



MATLAB for Engineers - Applications in Control, Electrical Engineering, IT and Robotics

Edited by Dr. Karel Perutka

ISBN 978-953-307-914-1
Hard cover, 512 pages
Publisher InTech
Published online 13, October, 2011
Published in print edition October, 2011

The book presents several approaches in the key areas of practice for which the MATLAB software package was used. Topics covered include applications for: -Motors -Power systems -Robots -Vehicles The rapid development of technology impacts all areas. Authors of the book chapters, who are experts in their field, present interesting solutions of their work. The book will familiarize the readers with the solutions and enable the readers to enlarge them by their own research. It will be of great interest to control and electrical engineers and students in the fields of research the book covers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Libor Pekař and Roman Prokop (2011). Implementation of a New Quasi-Optimal Controller Tuning Algorithm for Time-Delay Systems, MATLAB for Engineers - Applications in Control, Electrical Engineering, IT and Robotics, Dr. Karel Perutka (Ed.), ISBN: 978-953-307-914-1, InTech, Available from: http://www.intechopen.com/books/matlab-for-engineers-applications-in-control-electrical-engineering-it-and-robotics/implementation-of-a-new-quasi-optimal-controller-tuning-algorithm-for-time-delay-systems

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen