On the optimal pole assignment for time-delay systems

Libor Pekař

Abstract— The well-known fact about linear time-invariant time-delay systems (LTI-TDS) is that these systems have an infinite spectrum. Not only plants themselves but also the whole control feedbacks then have this undesirable feature in most cases. The aim of this contribution is to present algebraic controller design in a special ring of proper and stable meromorphic functions followed by an optimal pole assignment minimizing the spectral abscissa. The main problem is how to place feedback poles to the prescribed positions exactly by a finite number of free (controller) parameters. Clearly, it is not possible to place all poles but the idea is to push the rightmost ones as left as possible, which gives rise to the task of the spectral abscissa minimization. The spectral abscissa is a nonsmooth nonconvex function of free controller parameters in general. Moreover, there is a problem of its sensitivity to infinitesimally small delay changes. Four advanced iterative algorithms; namely, Quasi-Continuous Shifting Algorithm, Nelder-Mead algorithm, Extended Gradient Sampling Algorithm and Self-Organizing Migration Algorithm, are described as a possible numerical tools when minimization. Only two of them have already been used for the spectral abscissa minimization and none of them with the combination with algebraic controller design.

Keywords— Time-Delay Systems, Optimization, Spectral Abscissa, Pole Placement, Iterative Algorithms, Artificial Intelligence.

I. Introduction

A number of processes and systems are affected by any form of delay which unambiguously deteriorates the quality of a feedback control performance, namely stability and periodicity. Modern system and control theory has been dealing with this problem for longer than five decades starting by the idea of delay compensation by the Smith predictor [1]. LTI-TDS are frequently comprehended with lumped (pointwise) delays in the input-output relation only, which results in shifted arguments on the right-hand side of a set of ordinary differential equations (ODEs); however, this conception is rather restrictive in real-world applications since inner feedbacks can often be of time-distributed or delayed nature.

LTI-TDS in its modern meaning as anisochronic or hereditary models, in the contrary, offer a more universal dynamics description applying both integrators and delay elements on the left-hand side of a differential equation, either in a lumped or distributed form, yielding functional differential equations (FDEs). These models 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, see e.g. [2]–[6]. The most significant attribute of LTI-TDS is that due to transcendental nature of the characteristic equation, the system spectrum is infinite. So called retarded systems can only have a finite number of unstable system poles whereas the neutral ones may have poles formed in vertical strips along the imaginary axis. One can understand that dealing with analysis and synthesis of such systems is a rather involved.

These systems and models can be found in many theoretical and practical applications covering various fields of human activity, such as technology, informatics, biology, economy etc. Already Volterra [7] formulated differential equations incorporating the past states when he was studying predator-pray models. Aftereffect phenomenon is included in many processes, e.g. in chemical processes [8], heat exchange networks [9], [10], in internal combustion engines with catalytic converter [11], in metallurgic processes [12], in plastic industry [13] etc. Plenty of references to examples of processes with internal delays are introduced in [5], [6], [14].

Using the Laplace transform, LTI-TDS in a single-input single-output (SISO) case can be represented by a ratio of so called quasipolynomials [15] in one complex variable *s*, formed as linear combinations of products of *s*-powers and exponential terms, rather than polynomials which are usual in system and control theory. Hence, the Laplace transform of LTI-TDS is no longer rational and so called meromorphic functions have to be introduced. The transfer function denominator decides about the systems stability as usual, except cases of input-output or internal distributed delays [16], [17].

However, a transfer function representation in the form of quasipolynomial fractions is not suitable for controller design. A possibility is to introduce so called pseudopolynomials [16], or a meromorphic function representation can be extended to any type of the fractional description [18]. In order to meet natural requirements of asymptotical stability and controller properness (realizability), one may introduce the ring of stable

The author kindly appreciates the financial support which was provided by the Ministry of Education, Youth and Sports of the Czech Republic by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

L. Pekař is with the Tomas Bata University in Zlín, Faculty of Applied Informatics, nám. T. G. Masaryka 5555, 76001 Zlín, Czech Republic (corresponding author to provide phone: +420576035161; e-mail: pekar@fai.utb.cz).

and proper quasipolynomial (RQ) meromorphic functions (R_{MS}) [19]–[21]. Originally, the ring was developed for retarded systems only; however, the conception can be easily extended to neutral ones [4] as presented in this paper. Algebraic control design in this ring then employs the Bézout identity to obtain stable and proper controllers along with the Youla-Kučera parameterization for reference tracking and load disturbance rejection. The approach usually yields a delayed (anisochronic) controller.

Pole placement is a very favourite controller tuning method throughout the control theory and applications. As for delayless systems, the closed loop asymptotic stability (and even strong one, see [22]) is specified by the closed-loop characteristic (quasi)polynomial – whether a polynomial or a quasipolynomial is obtained, it is given by the particular controlled plant stability and factorization and by the control system structure – except the case of distributed delays, as mentioned above, where roots of the characteristic (quasi)polynomial do not coincide with closed loop poles. Since a controller can have only a finite number of coefficients representing selectable parameters, only some zeros of a characteristic quasipolynomial can be placed exactly.

In this paper, we define the pole placement problem as the minimization of the spectral abscissa function which means the real part of the right-most pole, instead of usual quasi-continuous shifting algorithms (QCSA) [23]–[26]. The presented methodology extends results by Vanbiervliet et al. [27] where state feedback controller design and the Extended Gradient Sampling Algorithm (EGSA) [28] for the abscissa minimization have been used. Unlike the cited paper, we do indtroduce three advanced optimization algorithms, namely the Nelder-Mead algorithm (NM) [29], the EGSA and the Self-Organizing Migration Algorithm (SOMA), e.g. in [30]. The NM and SOMA are used here to solve the presented problem for the first time. The QCSA is included into the presentation since it is intended to be used as an “initial” algorithm for primary poles shifting. The novelty is the combination of these optimal tuning principles with the algebraic controller structure design in R_{MS}.

Dealing with neutral LTI-TDS, moreover, brings about the problem of sensitivity of a vertical strip of poles to small delay changes which leads to the concept of so called strong stability, as mentioned above. Michiels and Vyhlídal [24] introduced a notion of the save upper bound which expresses the guaranteed real part of the right-most strip even infinitesimal delays shifts happen. In this sense, a spectral abscissa function value ought to be sought to the right from this line.

The paper is organized as follows. Possible state and input-output LTI-TDS models are introduced in Section II. Basic preliminaries about asymptotic, formal and strong stability LTI-TDS are presented in Section III. In Section IV, the R_{MS} ring is defined. Section V contains main steps of controller design in the ring. The main part of the contribution, i.e. the definition of the spectral abscissa and objective function for optimal pole placement and descriptions of the QCSA, NM, EGSA and SOMA, are introduced in Section VI.

II. LTI-TDS Models

A state space description of LTI-TDS can be given by a set of state and output FDEs in the form

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

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 real matrices of appropriate dimensions, $0\le\eta_i\le L$ stand for lumped delays and convolution integrals express distributed delays. If $\mathbf{H}_i\neq\mathbf{0}$ for any i = 1,2,...N_H, model (1) is called neutral; contrariwise, if $\mathbf{H}_i=\mathbf{0}$ for every i = 1,2,...N_H, a 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 a time instant, but also by a segment of the last model history of state and input variables

$$\mathbf{x}(t+\tau),\ \mathbf{u}(t+\tau),\ \tau\in[-L,0] \qquad (2)$$

Integrals in (1) can be exactly reformulated as sums using the Laplace transform or by the addition of a state variable, for details see e.g. [31], [32], or approximated via standard numerical approximation methods [6].

Regarding input-output description of LTI-TDS, the following general multi-input multi-output (MIMO) system in the form of the transfer matrix from (1) and considering zero initial conditions using the Laplace transform is obtained

$$\begin{aligned}\mathbf{Y}(s)&=\mathbf{G}(s)\mathbf{U}(s)=\frac{\mathbf{C}\,\mathrm{adj}[s\mathbf{I}-\mathbf{A}(s)]\mathbf{B}(s)}{\det[s\mathbf{I}-\mathbf{A}(s)]}\mathbf{U}(s)\\\mathbf{A}(s)&=s\sum_{i=1}^{N_H}\mathbf{H}_i\exp(-s\eta_i)+\mathbf{A}_0+\sum_{i=1}^{N_A}\mathbf{A}_i\exp(-s\eta_i)\\&\quad+\int_0^L\tilde{\mathbf{A}}(\tau)\exp(-s\tau)\mathrm{d}\tau \qquad (3)\\\mathbf{B}(s)&=\mathbf{B}_0+\sum_{i=1}^{N_B}\mathbf{B}_i\exp(-s\eta_i)+\int_0^L\tilde{\mathbf{B}}(\tau)\exp(-s\tau)\mathrm{d}\tau\end{aligned}$$

All transfer functions in $\mathbf{G}(s)$, or a transfer function in a single-input single-output (SISO) case, have identical denominator in the form

$$\begin{aligned} m(s) &= \operatorname{num}\det[s\mathbf{I}-\mathbf{A}(s)] = \operatorname{num}M(s) \\ &= s^n + \sum_{i=0}^{n}\sum_{j=1}^{h_i} m_{ij} s^i \exp(-s\eta_{ij}), \eta_{ij} \geq 0 \end{aligned} \tag{4}$$

where prefix num means the numerator of the determinant, and $\sum_{j=1}^{h_n} m_{nj} \exp(-\eta_{nj} s) \neq \text{constant}$ holds for a neutral system; otherwise, the system is retarded. The expression on the right-hand side of (4) represents a so called quasipolynomial [15]. Indeed, $M(s)$ is a ratio of quasipolynomials (i.e. a meromorphic function) in general due to distributed state (internal) delays, and all roots of the polynomial denominator of $M(s)$ are those of the numerator in this case. As a consequence, a transfer function (in a SISO case) can be expressed as a meromorphic function as well.

Note that input-output distributed delays yield the transfer function with a denominator polynomial factor the roots of which are those of the (quasi)polynomial numerator. Vice versa, for internal distributed delays, there is a polynomial factor in the transfer function numerator which zeros are those of the denominator.

III. Stability of LTI-TDS

LTI-TDS is said to be asymptotically stable if all poles are located in the open left half plane, $\mathbb{C}_0^-$, i.e. there is no s satisfying

$$M(s) = 0, \operatorname{Re} s \geq 0 \tag{5}$$

In the case of neutral systems, one has to be more careful when deciding about the stability since there may exist infinite strips of system poles tending to the imaginary axis. Moreover, these strips can be sensitive to even infinitesimally small deviations in delays. Hence, strictly negative roots of the characteristic (quasi)polynomial $m(s)$ (or the meromorphic function $M(s)$) do not guarantee a satisfactory stable behavior of a system from the asymptotic (and robust) point of view. Let us now introduce the associated difference equation and two stability notions for neutral LTI-TDS which are close to each other in the meaning.

Given a SISO neutral LTI-TDS (1), an associated difference equation is defined as

$$\mathbf{x}(t) - \sum_{i=1}^{N_H} \mathbf{H}_i \mathbf{x}(t-\eta_i) = 0 \tag{6}$$

A neutral TDS is said to be formally stable if

$$\operatorname{rank}\left[I - \sum_{i=1}^{N_H} \mathbf{H}_i \exp(-s\eta_i)\right] = n, \forall s : \operatorname{Re} s \geq 0 \tag{7}$$

see e.g. [16], [33]. It also means that a neutral LTI-TDS has only a finite number of poles in the (closed) right-half complex plane ($\mathbb{C}^+$) [34]. Clearly from (6) and (7), a system is formally stable if the characteristic equation

$$m_D(s) = \det\left[I - \sum_{i=1}^{N_H} \mathbf{H}_i \exp(-s\eta_i)\right] = 0 \tag{8}$$

expressing the spectrum of the difference equation has all its solutions in $\mathbb{C}_0^-$. The feature of a neutral LTI-TDS that rightmost solution of (8) is not continuous in delays, see e.g. [22], gives rise to a germane stability notion.

The difference equation (6) is strongly stable if it remains asymptotically stable when subjected to small variations in delays (i.e. a LTI-TDS remains formally stable). A necessary and sufficient strong stability condition in the Laplace transform can be formulated as

$$\sum_{j=1}^{h_i} \left|m_{nj}\right| < 1 \tag{9}$$

according to e.g. [4], [35], where m_{nj} are real coefficients for the highest s-power in (4).

Clearly, strong stability implies formal stability; contrariwise, a formally stable LTI-TDS can be destabilized in the formal sense by an infinitesimal change in delays.

IV. R_{MS} Ring

Algebraic approaches for analysis and control of LTI-TDS can be performed either in the state space or in the realm of input-output models where fields, modules and rings as principal algebraic notions and tools are utilized. Usually, commensurate delays, i.e. those which can be expressed as integer multiples of the smallest one, are assumed; however, delays are naturally real-valued and thus this assumption is rather restrictive for real applications.

As we are focused on Laplace transform models, hence, they are subjected below. A model with commensurate delays or that under some rational approximation yields the transfer function representation in the form of a fraction of polynomials over real numbers which is the most frequently used ring even for LTI-TDS. Non-commensurate or rationally unapproximated delays results in a fraction of quasipolynomials as presented above. However, these transfer function representations are not suitable in order to satisfy some basic control requirements, e.g. controller feasibility, feedback strong and asymptotic stability.

Rather more general approaches [18], [36] 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, i.e. division is not generally allowed. A powerful algebraic tool ensuring requirements above is a ring of stable and proper RQ-meromorphic functions (R_{MS}). Since the original definition of R_{MS} in [19] does not constitute a ring, some minor changes in the definition were made in [20]. Namely, although the retarded structure of LTI-TDS is considered only, the minimal ring conditions require the use of neutral quasipolynomials at least in the numerator as well. The ring was completely redefined and some its properties were introduced in [21] comprising models of neutral type and those with distributed delays.

An element $T(s)$ of R_{MS} ring is represented by a ratio of two (quasi)polynomials $y(s)/x(s)$ where the denominator is a (quasi)polynomial of degree n and the numerator can be factorized as

$$y(s)=\tilde{y}(s)\exp(-\tau s) \tag{10}$$

where $\tilde{y}(s)$ is a (quasi)polynomial of degree l and $\tau \geq 0$. Note that the degree of a quasipolynomial means its highest s-power.

The element lies in the space $H_\infty(\mathbb{C}^+)$, providing the finite norm defined as

$$\|T\|_\infty := \sup\{|T(s)| : \operatorname{Re} s \geq 0\} \tag{11}$$

i.e. it is analytic and bounded in $\mathbb{C}^+$, particularly, there is no pole s_0 such that $\operatorname{Re} s_0 \geq 0$ for a retarded denominator or $\operatorname{Re} s_0 \geq -\varepsilon, \varepsilon > 0$ for a neutral one. If the term includes distributed delays, all roots of $x(s)$ in $\mathbb{C}^+$ are those of $y(s)$ (i.e. removable singularities). Moreover, $T(s)$ is formally stable. The strong stability condition (9) for (quasi)polynomial $x(s)$ is a sufficient but not necessary condition guaranteeing that.

In addition, the ratio is proper, i.e. $l \leq n$. More precisely, there exists a real number $R > 0$ for which holds that

$$\sup_{\operatorname{Re} s>0, |s|\geq R} |T(s)| < \infty \tag{12}$$

see [37].

Let the plant be initially described by the transfer function

$$G(s)=\frac{b(s)}{a(s)} \tag{13}$$

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

$$G(s)=\frac{B(s)}{A(s)} \tag{14}$$

where $A(s), B(s) \in R_{MS}$ are coprime, i.e. there does not exist a non-trivial (non-unit) common factor of both elements. Details about divisibility can be found in [21]. 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 [16]. More precisely, two coprime elements $A(s), B(s) \in R_{MS}$ form a Bézout factorization if and only if

$$\inf_{\operatorname{Re} s\geq 0} (|A(s)|, |B(s)|) > 0 \tag{15}$$

It was, moreover, proved there that a strictly proper system which is not formally stable cannot be bounded input bounded output (BIBO) and even H_∞ stabilizable.

V. Controller Design in R$_{MS}$

The aim of this section is to outline controller design based on the algebraic approach in the R_{MS} ring satisfying the closed loop stability in that sense that all transfer functions in the feedback are from R_{MS} (i.e. they lies in H_∞, are proper and strongly stable) controller feasibility, reference tracking and load disturbance rejection. As a control system, the simple feedback loop is chosen for the simplicity, see Fig. 1.

For algebraic controller design in R_{MS} it is initially supposed that not only the plant is expressed by the transfer function over R_{MS} but a controller and all system signals are over the ring. Let $W(s)$ be 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 affected by a load disturbance, 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.

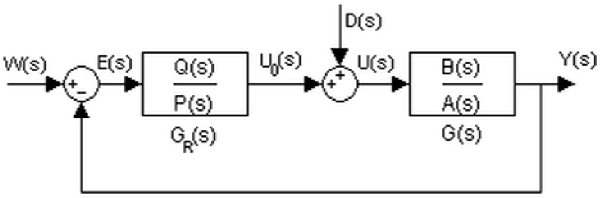

Fig. 1 Control feedback structure

External inputs, reference and load disturbance signals, respectively, have forms

$$W(s)=\frac{H_W(s)}{F_W(s)},\ D(s)=\frac{H_D(s)}{F_D(s)} \tag{16}$$

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

The following important feedback transfer functions can be derived

$$\begin{aligned} G_{WY}(s) &= \frac{Y(s)}{W(s)} = \frac{B(s)Q(s)}{M(s)},\ G_{DY}(s) = \frac{Y(s)}{D(s)} = \frac{B(s)P(s)}{M(s)} \\ G_{WE}(s) &= \frac{E(s)}{W(s)} = \frac{A(s)P(s)}{M(s)}, G_{DE}(s) = \frac{E(s)}{D(s)} = -\frac{B(s)P(s)}{M(s)} \end{aligned} \tag{17}$$

where the controller transfer function is factorized as follows

$$G_R(s) = \frac{Q(s)}{P(s)} \tag{18}$$

and the common denominator meromorphic function reads

$$M(s) = A(s)P(s) + B(s)Q(s) \tag{19}$$

Meromorphic functions $Q(s), P(s)$ are from R_{MS} and the fraction (18) is (Bézout) coprime. The numerator of $M(s) \in R_{MS}$ agrees with the characteristic quasipolynomial of the closed loop defined in (4).

A brief description of basic control design steps follows.

A. *Closed Loop Stabilization*

Given a Bézout coprime pair $A(s), B(s) \in R_{MS}$ the closed-loop system is stable if and only if there exists a (coprime) pair $P(s), Q(s) \in R_{MS}$ of the controller denominator and numerator, respectively, satisfying the Bézout identity

$$A(s)P(s) + B(s)Q(s) = 1 \tag{20}$$

A particular stabilizing solution of (20), say $P_0(s), Q_0(s)$, can be then parameterized as

$$\begin{aligned} P(s) &= P_0(s) \pm B(s)Z(s) \\ Q(s) &= Q_0(s) \mp A(s)Z(s) \end{aligned} \tag{21}$$

where $Z(s) \in R_{MS}$. Parameterization (21) is used to satisfy remaining control and performance requirements, such as reference tracking, disturbance rejection etc.

The proof of the statement above can be done analogously as in [19] where a three-step proof for a similar ring was presented. Condition (15) ensures i.a. that there can exist the ring inversion of $M(s)$ since it proves that there is no common zero of $A(s), B(s)$ in $\mathbb{C}^+$ (including infinity).

B. *Reference Tracking*

The task of this subsection is to find $Z(s) \in R_{MS}$ in (21) so that the reference signal is being tracked. The solution idea results the form of $G_{WE}(s)$ defined in (17). Consider the limit

$$\begin{aligned} &\lim_{t\to\infty} e_W(t) = \lim_{s\to 0} sE_W(s) = \lim_{s\to 0} sG_{WE}(s)W(s) \\ &= \lim_{s\to 0} sA(s)P(s)\frac{H_W(s)}{F_W(s)} \end{aligned} \tag{22}$$

where $\cdot_W$ expresses that the signal is a response to the reference not influenced by other external inputs. Limit (22) reaches zero if $\lim_{s\to 0} E_W(s) < \infty$ and $E_W(s)$ is analytic and bounded in the open right half-plane, i.e. $E_W(s) \in H_\infty(\mathbb{C}^+)$. Moreover, it must hold that $G_{WE}(s)$ is proper (or, equivalently, $E_W(s)$ is strictly proper) because of the feasibility (impulse free modes) of $e_W(t)$. If one want to prevent the closed loop stability from the sensitivity to small delays, the denominator of $E_W(s)$ must be a (quasi)polynomial satisfying (9). This implies, in other words, that the reference tracking is fulfilled if $E_W(s) \in R_{MS}$.

Alternatively, $F_W(s)$ must divide the product $A(s)P(s)$ in R_{MS}. Hence, all unstable zeros (including infinity) of $F_W(s)$ must be those of $A(s)P(s)$ and, moreover, the quasipolynomial numerator of $F_W(s)$ is strongly stable. It means that one has to set all unstable zeros of $F_W(s)$ (with corresponding multiplicities) as zeros of $P(s)$ - if there is no one already contained in $A(s)$. Recall that zeros mean zero points of a whole term in R_{MS}, not only those of a quasipolynomial numerator.

C. *Load Disturbance Attenuation*

The attenuation of the load disturbance signal entering a plant can be done analogously as for reference tracking. Thus, $Z(s)$ is chosen so that $Y_D(s) \in R_{MS}$ which is clear from

$$\begin{aligned} &\lim_{t\to\infty} y_D(t) = \lim_{s\to 0} sY_D(s) = \lim_{s\to 0} sG_{DY}(s)D(s) \\ &= \lim_{s\to 0} sB(s)P(s)\frac{H_D(s)}{F_D(s)} \end{aligned} \tag{23}$$

where $\cdot_D$ means that the output is influenced only by the disturbance. Or, $F_D(s)$ must divide the product $B(s)P(s)$ in R_{MS}.

One has to be careful when deciding about the form of $Z(s)$ since both divisibility conditions must be fulfilled simultaneously. A detailed procedure of reference tracking and disturbance rejection briefly described above was presented e.g. in [38].

VI. Spectral Abscissa and Minimization Techniques

Definition of the spectral abscissa, which gives rise to the objective function, and description of NM, EGSA and SOMA optimization techniques are topics of this section. First, let us give the motivation for the controller parameters optimization.

A. Motivation

The presented controller design in R_{MS} yields a controller in so called anisochronic form, which means that it contains internal (state) delays as well. Using a simple feedback loop for an unstable controlled plant results in a characteristic quasipolynomial instead of polynomial, i.e. the closed loop system is infinite-dimensional having the infinite spectrum (that is, all transfer functions have quasipolynomial denominator). For stable controlled plants, some feedback transfer functions are finite-dimensional, yet, some others do not; hence, the whole feedback is infinite-dimensional (with the characteristic quasipolynomial instead of a polynomial). Pole assignment philosophy, which places closed-loop poles to the prescribed positions, can not be adopted as for finite-dimensional systems since the controller owns only a finite number of free (selectable) parameters. Here we have the infinite number of poles which can not be place exactly. A possibility is to optimize the whole spectrum so that the right-most (i.e. “least stable”) pole is moved to the left as much as possible.

Analogously, one can not ensures that the numerator of the reference-to-output transfer function, $G_{WY}(s)$, is a polynomial. Thus, the closed-loop zeros can be optimized in the similar way as poles are, because of the fact that zeros placed too right in the complex plane cause undesirable high oscillations, see details e.g. in [26].

B. Spectral Abscissa

Let the controller obtained by the approach described in Section V has *r* selectable parameters $\mathbf{K} = \{k_1, k_2, ..., k_r\}$. The spectral abscissa function, $\alpha(\mathbf{K})$, is defined as

$$\alpha(\mathbf{K}) = \max \operatorname{Re}(s_i) \tag{24}$$

where s_i are system poles and $\alpha(\mathbf{K})$ is strictly negative, see e.g. [27], [39]. As presented in the motivation above, the objective is to solve the optimization problem

$$\min_{\mathbf{K}} \Phi(\mathbf{K}) = \min_{\mathbf{K}} \alpha(\mathbf{K}) \tag{25}$$

The question is why a complex optimization algorithm ought to be used instead of a standard one, say, the well known steepest descent algorithm. The reason lies in some spectral abscissa function properties. The first problem arises from the fact that $\alpha(\mathbf{K})$ is non-convex, i.e. it may have multiple local minima. It is clear that with such behavior the global minimum is hard to find, and many optimization algorithms will converge to a local minimum. The second difficulty is that $\alpha(\mathbf{K})$ is non-smooth w.r.t. parameter changes in points where are more the one real poles or conjugate pairs with the maximum real part [23], [27]. At this point the function $\alpha(\mathbf{K})$ is hence not differentiable. As third, the function is non-Lipschitz, for example, at points where the maximum real part has multiplicity greater than one [28]. However, it is assumed that the spectral abscissa is differentiable almost everywhere.

Analogously, if the feedback is finite-dimensional, zeros rather than poles can be considered in (24), (25).

C. Objective Function for Neutral LTI-TDS

However, the objective function $\Phi(\mathbf{K})$ defined in (25) is not suitable form for neutral LTI-TDS since one has to take into account the notion of the strong (or formal) stability. It means that condition (9) has to be included into objectives, either as a restriction for the objective function (25) or as a penalty subfunction as a part of the final objective function as presented in [39]. The latter idea yields the objective function e.g. in the form

$$\Phi(\mathbf{K}) = \alpha(\mathbf{K}) + \lambda\left(\sum_{j=1}^{h_i} \left|m_{nj}(\mathbf{K})\right|\right)^2 \tag{26}$$

where λ represents a weighting parameter. This conception, however, does not guarantee that the restriction (9) holds true. Alternatively, a barrier logarithmic function

$$\Phi(\mathbf{K}) = \alpha(\mathbf{K}) + \lambda \log^{-1}\left(\sum_{j=1}^{h_i} \left|m_{nj}(\mathbf{K})\right|\right) \tag{27}$$

might be used instead of (26), which can solve this problem. Obviously, an addition of condition (9) into the objective function makes sense only if some of $m_{nj}(\mathbf{K})$ can be adjusted by controller parameters, which is not quite frequent.

A neutral LTI-TDS brings other problem related to generic properties of such class of systems. Since there exist vertical strips of characteristic roots the position of which in the real axis is not continuous w.r.t. delays, the so called safe upper bound has been defined [24]. The notion expresses the real number that is definitely higher than the real part of the rightmost strip. If such number is strictly negative, the system is strongly stable and thus it can be stabilized safely.

More precisely, define $c_D(\boldsymbol{\eta})$ as

$$c_D(\boldsymbol{\eta}) := \sup\{\operatorname{Re} s : m_D(s) = 0\} \tag{28}$$

where $\boldsymbol{\eta}$ is the vector of N_H delays and $m_D(s)$ is the characteristic quasipolynomial (8) related to the associated difference equation (6). As mentioned above, $c_D(\boldsymbol{\eta})$ is not continuous w.r.t. $\boldsymbol{\eta}$ and it expresses the real part of the rightmost strip of poles of a neutral LTI-TDS. The safe upper bound $\overline{C}_D(\boldsymbol{\eta}) \in \mathbb{R}$ is defined as follows

$$\overline{C}_D(\boldsymbol{\eta}) := \lim_{\varepsilon \to 0^+} \sup\{c_D(\boldsymbol{\eta} + \delta\boldsymbol{\eta}) : \|\delta\boldsymbol{\eta}\| < \varepsilon\} \tag{29}$$

It holds that $\overline{C}_D(\boldsymbol{\eta}) \in \geq c_D(\boldsymbol{\eta})$ and $\overline{C}_D(\boldsymbol{\eta})$ is continuous in the delays. It has been proved in [24] that the quantity $\overline{C}_D(\boldsymbol{\eta})$ is the unique zero of the strictly decreasing function

$$c \in \mathbb{R} \to f(c, \boldsymbol{\eta}) - 1 \tag{30}$$

where $f(c, \boldsymbol{\eta})$ is defined as

$$f(c, \boldsymbol{\eta}) := \max_{\boldsymbol{\theta} \in [0, 2\pi]^{N_H}} \rho\left(\sum_{i=1}^{N_H} \mathbf{H}_i \exp(\mathrm{j}\theta_i - c\eta_i) \right) \tag{31}$$

where $\rho(\cdot)$ means the spectral radius. It is possible to estimate an upper bound on $\overline{C}_D(\boldsymbol{\eta})$ using the fact that

$$f(c, \boldsymbol{\eta}) \leq \sum_{i=1}^{N_H} \|\mathbf{H}_i\| \exp(-c\eta_i) \tag{32}$$

as the unique solution of the equation

$$\sum_{i=1}^{N_H} \|\mathbf{H}_i\| \exp(-c\eta_i) = 1 \tag{33}$$

If the control law can not change any of $\mathbf{H}_i$ (or, equivalently any of m_{nj}), one can concentrate on the characteristic roots (poles) with the real part larger than $\overline{C}_D(\boldsymbol{\eta})$, since the value of $\overline{C}_D(\boldsymbol{\eta})$ can not be adjusted in this case. It holds that all poles in the half plane $\operatorname{Re} s \geq \overline{C}_D + \varepsilon, \varepsilon > 0$, lie in a compact set and the number of these roots is finite [22], [24]. Hence when minimizing the objective function (or when pole placement, in general), only isolated poles right from the value of $\overline{C}_D(\boldsymbol{\eta})$ can be taken into account (shifted). In the contrary, if $\mathbf{H}_i$ can be changed, the value of $\overline{C}_D(\boldsymbol{\eta})$ varies and it must be recalculated in every iteration step; however, there is still no reason deal with the characteristic roots left from $\overline{C}_D(\boldsymbol{\eta})$. The knowledge of $\overline{C}_D(\boldsymbol{\eta})$ prevents to spend much control action to poles with smaller real part which are useless when stabilization and the minimizing of the spectral abscissa.

The case $\overline{C}_D(\boldsymbol{\eta}) > 0$ agrees with the strong instability, and if it not possible to improve $\overline{C}_D(\boldsymbol{\eta})$, one can give the controller tuning up.

D. Minimization Techniques

Description of some numerical optimization techniques we decided to utilize for the minimization of the objective function (spectral abscissa) follows. All these approaches enable to overcome all the difficulties with non-convexity and non-differentiability of the spectral abscissa function.

1) Nelder-Mead (NM) algorithm

The NM algorithm belonging to the class of comparative (direct search) algorithms, also called irregular simplex search algorithm, was originally published in [29]. This easy-to-use method does not require derivatives of the objective function and thus it is suitable for non-smooth functions. It is very popular and can be implemented in many different ways.

The method typically requires only one or two function evaluations per iteration, which is useful especially in applications where each function evaluation is time-consuming. On the other hand, the algorithm can fail since the convergence for non-smooth or discontinuous functions have not been proved yet [40]. It can also require an enormous amount of iterations to obtain a significant improvement in function value.

Consider a nonlinear objective function $\mathbf{K} \in \mathbb{R} \to \Phi(\mathbf{K}) \in \mathbb{R}^r$ to be minimized. The basic steps of the general algorithm can be done as follows.

Input: Objective function $\Phi(\mathbf{K})$.

Step 1: Construct the initial working simplex *S*, set transformation and termination parameters.

Step 2: Calculate the termination test information. If the test is satisfied, stop the algorithm.

Step 3: Order simplex vertices as the worst, second worst and the best one.

Step 4: Calculate the central point and reflex the worst vertex. If the reflection is successful, accept the reflected point in the new working simplex and go to Step 3.

Step 5: Try to use contraction or expansion. If this succeeds, the accepted point becomes the new vertex; otherwise, shrink the simplex towards the best vertex. Go to Step 3.

Output: The best vertex and its function value.

Let us describe each step of the algorithm in more details. Note that we keep the notation for the objective function (spectral abscissa).

The working simplex construction

A simplex S in $\mathbb{R}^r$ is a convex hull of $r+1$ vertices $\mathbf{K}_1, \mathbf{K}_2, ..., \mathbf{K}_{r+1} \in \mathbb{R}^r$

$$S = \operatorname{conv}\{\mathbf{K}_1, \mathbf{K}_2, ..., \mathbf{K}_{r+1}\} \tag{34}$$

The initial (non-degenerate) simplex can be constructed either as a regular or as a right-angled simplex. The latter is easier to handle as

$$\mathbf{K}_j = \mathbf{K}_1 + h_j \mathbf{e}_j,\ j = 2, ..., r+1 \tag{35}$$

where $\mathbf{K}_1$ is a “starting” point, h_j stands for a stepsize and $\mathbf{e}_j$ is a unit (Euclidean) vector in $\mathbb{R}^r$.

Termination parameters

During the minimization the simplex changes in its size and shape as well. The algorithm terminates when either the simplex is sufficiently small or the function values at the vertices are close to each other or the number of iterations reaches the prescribed number. Usually some of these three conditions are combined together and the procedure ends when at least one of the conditions becomes true. We use the limit number of iterations, say *ni*. Moreover, for discontinuous functions the termination test has to include the information of the simplex size [40] whereas the function values test is useless. Let ε_S is the limit simplex size defined by the user, and then the termination test related to the simplex size can be formulated as

$$\sum_{i=2}^{r+1} \left\| \mathbf{K}_i - \mathbf{K}_1 \right\| < \varepsilon_S \tag{36}$$

Ordering

Determine the best ($\mathbf{K}_{min}$), second worst ($\mathbf{K}_s$) and the worst vertex ($\mathbf{K}_{max}$) as

$$\Phi_{min} = \min_i \Phi(\mathbf{K}_i), \Phi_{max} = \max_i \Phi(\mathbf{K}_i), \Phi_s = \max_{i \neq max} \Phi(\mathbf{K}_i) \tag{37}$$

Reflection

The central point can be imagined as the “mean” coordinate of all vertices except the worst one, i.e.

$$\mathbf{K}_c = \frac{1}{r} \sum_{\substack{i=1 \\ i \neq max}}^{r} \mathbf{K}_i \tag{38}$$

The calculation of the new simplex then continues by reflecting $\mathbf{K}_{max}$ over $\mathbf{K}_c$ to a new position $\mathbf{K}_{ref}$ according to the formula

$$\mathbf{K}_{ref} = \mathbf{K}_c + \alpha(\mathbf{K}_c - \mathbf{K}_{max}) \tag{39}$$

where $\alpha > 0$ is a reflection control parameter, usually $\alpha = 1$.

If it holds that $\mathbf{K}_{min} \leq \mathbf{K}_{ref} < \mathbf{K}_s$, the iteration is finished and $\mathbf{K}_{ref}$ becomes a new simplex point instead of $\mathbf{K}_{max}$.

Contraction and expansion

If the reflection does not succeed, one has to perform expansion or contraction, depending on the value of $\Phi(\mathbf{K}_{ref})$ relation to $\Phi(\mathbf{K}_{min})$, $\Phi(\mathbf{K}_s)$ and $\Phi(\mathbf{K}_{max})$. Hence, if $\Phi(\mathbf{K}_{ref}) < \Phi(\mathbf{K}_{min})$, i.e. the reflected point is the best one, the expansion point is computed as follows

$$\mathbf{K}_{exp} = \mathbf{K}_c + \beta(\mathbf{K}_{ref} - \mathbf{K}_c) \tag{40}$$

where $\beta > 1$ is an expansion control parameter, usually $\beta = 2$. There are more ways how to construct the new working simplex; however, to avoid premature termination of iterations for non-smooth functions, see (Rowan 1990), $\mathbf{K}_{exp}$ becomes the new simplex vertex if $\Phi(\mathbf{K}_{exp}) < \Phi(\mathbf{K}_{min})$. Otherwise, $\mathbf{K}_{ref}$ is accepted.

There are two types of contractions; first, if $\Phi(\mathbf{K}_s) \leq \Phi(\mathbf{K}_{ref}) < \Phi(\mathbf{K}_{max})$, compute the contracted point as

$$\mathbf{K}_{con} = \mathbf{K}_c + \gamma(\mathbf{K}_{ref} - \mathbf{K}_c) \tag{41}$$

where $0 < \gamma < 1$ is a contraction control parameter mostly set as $\gamma = 0.5$. If $\Phi(\mathbf{K}_{con}) < \Phi(\mathbf{K}_{ref})$, $\mathbf{K}_{con}$ becomes a vertex in the new working simplex; otherwise, shrinkage has to be made. On the contrary, if $\Phi(\mathbf{K}_{ref}) \geq \Phi(\mathbf{K}_{max})$, i.e. $\mathbf{K}_{ref}$ is the worst point, one ought to perform contraction according to

$$\mathbf{K}_{con} = \mathbf{K}_c + \gamma(\mathbf{K}_{max} - \mathbf{K}_c) \tag{42}$$

If it holds that $\Phi(\mathbf{K}_{con}) < \Phi(\mathbf{K}_{max})$, accept $\mathbf{K}_{con}$; otherwise, perform shrinkage.

Shrinkage

In the case that expansion or contraction fails, one has to shrink the current simplex towards the best vertex $\mathbf{K}_{min}$. This operation is given by the formula

$$\mathbf{K}_i = \mathbf{K}_{min} + \delta(\mathbf{K}_i - \mathbf{K}_{min}), i = 1,2,\ldots,r+1 \neq min \tag{43}$$

Experiences with the algorithm show that shrink transformations almost never happen in practice [40]. A non-shrink iteration of the algorithm is fast, since only one or two function values are computed.

2) Extended Gradient Sampling Algorithm (EGSA)

The second optimization approach presented here is the Extended Gradient Sampling Algorithm (EGSA), see [27], [39] which is based on the gradient sampling algorithm developed by Burke et al. [28]. The original algorithm is essentially an extension of the well-known steepest descent method. The basic difference lies in the computation of the non-smooth search direction. Thus, EGSA requires, with comparison to NM algorithm, a numerical estimation of the gradient, even in points where the objective function is not differentiable. It is expected that $\Phi(\mathbf{K})$ is differentiable almost everywhere. The basic steps of EGSA can be given as follows.

Input: Objective function $\Phi(\mathbf{K})$.

Step 1: Initialize a starting point $\mathbf{K}_0$ arbitrarily. Set control and termination parameters.

Step 2: Choose $r+1$ points near by $\mathbf{K}_0$. Compute the Clarke subdifferential and the (non-smooth) steepest descent direction using the gradient sampling method. If the norm of the direction is very small, then terminate the algorithm.

Step 3: Calculate the step length along the direction from

Step 2. If it fails choose another (substitute) direction. If all possible directions fail, stop.

Step 4: Update the current position $\mathbf{K}_i$ to $\mathbf{K}_{i+1}$ and go to Step 2.

Output: The best position and its function value.

Starting point selection

The initial point $\mathbf{K}_0$ can be selected freely, or there is a possibility to utilize the quasi-continuous poles shifting algorithm (QCSA) to obtain a convenient position in the hyperspace of control tuning parameters. For details about the CQSA, the reader is referred to [23], [24].

Computing the non-smooth steepest descent direction

The advantage of EGSA compared to the classical steepest descent direction lies in the ability to compute (estimate) gradient even for non-smooth objective functions. The generalized gradient, also called Clarke subdifferential, is given by

$$\partial_c \Phi(\mathbf{K}_i) = \mathrm{conv}\left\{ \lim_{\mathbf{K}\to\mathbf{K}_i} \nabla\Phi(\mathbf{K}) : \mathbf{K} \in N \right\} \tag{44}$$

where ∇ denotes the gradient of Φ in any point from a subset of a neighborhood *N* around the current position $\mathbf{K}_i$, in *i*th iteration, where Φ is differentiable.

The non-smooth steepest descent direction can defined as the negative of the vector with the smallest norm in the Clarke subdifferential, i.e.

$$\mathbf{d}_i = -\underset{\mathbf{v}\in\partial_c\Phi(\mathbf{K}_i)}{\arg\min} \|\mathbf{v}\| \tag{45}$$

The Clarke subdifferential can be approximated as follows [28]

$$\partial_c \Phi(\mathbf{K}_i) \approx \left\{ \begin{matrix} \nabla\Phi(\mathbf{K}_{i,1}), \nabla\Phi(\mathbf{K}_{i,2}), \ldots, \nabla\Phi(\mathbf{K}_{i,r+1}) : \\ \mathbf{K}_{i,j} \in N,\ j = 1,2,\ldots r+1 \end{matrix} \right\} \tag{46}$$

It means that the subdifferential can be expressed as a bundle of $r+1$ gradients calculated at some points on the hypersphere around iterate $\mathbf{K}_i$, under the assumption that the objective function is differentiable almost everywhere. If one of the sampled points is not differentiable, it is discarded and re-sampled. In [27] it is suggested that taking $2r$ points from the neighborhood of $\mathbf{K}_i$ instead of $r+1$ is a save choice for the calculation of the gradient which is described in the next subsection.

The neighborhood is typically chosen to be a ball with the centre at $\mathbf{K}_i$ and radius ρ. The choice of ρ is a difficult task; one possibility is to decrease the radius during iterations.

As mentioned above, the non-smooth steepest descent direction agrees with the negative of vector from the bundle with the smallest norm. This can be calculated by a simple quadratic program

$$\begin{aligned} \mathbf{v}_{\min,i} &= \underset{\mathbf{v}}{\arg\min}\, \mathbf{v}^T \left(\partial_c \Phi(\mathbf{K}_i)\right)^T \left(\partial_c \Phi(\mathbf{K}_i)\right)\mathbf{v} \\ &= \underset{\mathbf{v}}{\arg\min} \left\| \left(\partial_c \Phi(\mathbf{K}_i)\right)\mathbf{v}_{\min,i} \right\|^2 = \underset{\mathbf{v}}{\arg\min} \|\mathbf{d}_i\|^2 \end{aligned} \tag{47}$$

i.e. $\mathbf{d}_i = -\left(\partial_c \Phi(\mathbf{K}_i)\right)\mathbf{v}_{\min,i}$. There is, however, the mishmash between sizes of $\mathbf{v}$ and $\left(\partial_c \Phi(\mathbf{K}_i)\right)$; therefore, we suggest to exclude a particular $\mathbf{v}$ from (47) when evaluation. Moreover, if $\left(\partial_c \Phi(\mathbf{K}_i)\right)$ is not positive definite, one can obtain an undesirable direction (towards the maximum). Hence, it is possible to take $\mathbf{d}_i = -\mathbf{v}_{\min,i}$.

Usually, the direction is normalized as

$$\mathbf{d}_{norm,i} = \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|} \tag{48}$$

to avoid large jumps in the space of $\mathbf{K}$.

Then $\mathbf{K}$ is updated as follows

$$\mathbf{K}_{i+1} = \mathbf{K}_i + \lambda \mathbf{d}_{norm,i} \tag{49}$$

where λ is the step length the searching of which is described in a subsection below.

The iterations are terminated if so called Clarke stationary is reached, which occurs when the norm of the non-smooth steepest descent is exactly zero (theoretically). We omit test presented in [39], see there for details, and perform the following simple test instead

$$\|\mathbf{d}_i\| < \varepsilon \tag{50}$$

where $\varepsilon > 0$ is a sufficiently small termination parameter and the gradient in not normed. Alternatively, one can terminate the algorithm if the prescribed number of iterations is reached.

Gradient estimation

Numerical approximation of the gradient at differentiable points in the vicinity of the current iteration can be performed e.g. as described in [39] using the simplex gradient method. Apparently, the subset *N* must constitute a simplex defined above. Define the matrix $\mathbf{V}(\mathbf{K}_{i,1}) \in \mathbb{R}^{r \times l}$ of simplex directions

$$\mathbf{V}(\mathbf{K}_{i,1}) := \left[\mathbf{K}_{i,2} - \mathbf{K}_{i,1}, \mathbf{K}_{i,3} - \mathbf{K}_{i,1}, \ldots, \mathbf{K}_{i,l} - \mathbf{K}_{i,1}\right] \tag{51}$$

where $\mathbf{K}_{i,1}$ is an arbitrary point from *N* in which the gradient is calculated.

The value of *l* equals the number of points around $\mathbf{K}_i$ minus one, i.e. it may be in most cases $l = r$ or $l = 2r - 1$, see the remark in the previous subsection.

A vector of objective function differences is defined as

$$\delta\left(\Phi\left(\mathbf{K}_{i,1}\right)\right):=\begin{bmatrix}\Phi\left(\mathbf{K}_{i,2}\right)-\Phi\left(\mathbf{K}_{i,1}\right),\Phi\left(\mathbf{K}_{i,3}\right)-\Phi\left(\mathbf{K}_{i,1}\right),\\ \ldots,\Phi\left(\mathbf{K}_{i,l}\right)-\Phi\left(\mathbf{K}_{i,1}\right)\end{bmatrix}^T \quad (52)$$

Finally, the simplex gradient $\mathbf{D}\Phi\left(\mathbf{K}_{i,1}\right)$ as a numerical approximation of the gradient $\nabla\Phi\left(\mathbf{K}_{i,1}\right)$ reads

$$\mathbf{D}\Phi\left(\mathbf{K}_{i,1}\right)=\left(\mathbf{V}\left(\mathbf{K}_{i,1}\right)^T\right)^+\delta\left(\Phi\left(\mathbf{K}_{i,1}\right)\right)\approx\nabla\Phi\left(\mathbf{K}_{i,1}\right) \quad (53)$$

where $(\cdot)^+$ denotes the so called Moore-Penrose pseudoinverse [41] which equals the standard matrix inverse for $l=r$.

Step length selection

The task is to set λ in (49) appropriately. There are many ways how to solve the problem, e.g. in (Vyhlídal 2010) a sufficient decrease in Φ is given by

$$\Phi\left(\mathbf{K}_i+\lambda\mathbf{d}_{norm,i}\right)-\Phi\left(\mathbf{K}_i\right)>-\beta^{m+1}\left\|\mathbf{v}_{\min,i}\right\| \quad (54)$$

where $\beta\in(0,1)$ and $m\geq 1$ is the smallest integer such that (54) holds. However, such solution can fail for convex functions.

We decided, in the contrary, to perform an easy test as follows. First, discretize λ into small steps, say $\lambda_k=k\Delta\lambda, k=0,1,2,\ldots$. Then find

$$\lambda_{\max}:=\max\lambda:\Phi\left(\mathbf{K}_i+\lambda_{k+1}\mathbf{d}_{norm,i}\right)-\Phi\left(\mathbf{K}_i+\lambda_k\mathbf{d}_{norm,i}\right)<0 \quad (55)$$

Naturally, it is possible to set an upper bound on $\lambda_{\max}$, say $\overline{\lambda_{\max}}$, to avoid large jumps.

If this method fails, i.e. $\lambda_{\max}=0$, the second best vector in (47) is accepted, etc. If all directions fail, try to do (55) with $\mathbf{d}_{norm,i}=-\mathbf{d}_{norm,i}$.

3) Self-Organizing Migration Algorithm (SOMA)

SOMA is ranked among evolution algorithms, more precisely genetic algorithms, dealing with populations similarly as differential evolution does, see e.g. [30]. The algorithm is based on vector operations over the space of feasible solutions (parameters) in which the population is defined. In SOMA, every single generation, in which a new population is generated, is called a migration round. Population specimens cooperate while 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.

The method converges very fast; however, the number of function evaluations in every iteration can be very high - depending on the number of specimens and step length when moving on the hyperspace.

The main steps of the basic algorithm strategy called “All to One” can be formulated as follows.

Input: Objective function $\Phi(\mathbf{K})$.

Step 1: Set control and termination parameters. Generate a population based on a selected prototypal specimen.

Step 2: Find the best specimen (leader), i.e. that with the minimal function value.

Step 3: Move all other specimens towards the leader and evaluate their function values in each step.

Step 4: Select the new population and test the minimal divergence of the population. If it succeeds, stop. Otherwise, go to Step 2.

Output: The leader and its function value.

Look at these steps in more details again.

Prototypal specimen

A population described below in a separate subsection must be generated based on a prototypal specimen. This specimen is a vector of (controller free) parameters $\mathbf{K}$ which can be found e.g. by the quasi-continuous poles shifting algorithm.

Control and termination parameters

Specific control and termination parameters, which have to be set before the rest of the algorithm starts, are explained in this subchapter.

Two parameters, the initial radius (*Rad*) and the size of the population (*PopSize*), control the construction of an initial population based on the prototypal specimen. *Rad* > 0 should be chosen high enough to cover the range of all acceptable parameters. *PopSize* > 0 means the number of specimens in the population – the higher value yields a higher chance to find a global minimum yet the computational time increases.

The moving of specimens on the hyperspace of searched parameters is given by four control parameters: *PathLength*, *Step*, *PRT* and $\mathbf{v}_{PRT}$. *PathLength* should be within the interval $[1.1,5]$ and it expresses the length of the path when approaching the leader. For instance, *PathLength* = 1 means that the specimen stops its moving exactly at the position of the leader. The value of $Step\in[0.11,PathLength]$ represents the sampling of the path. E.g. a pair of settings *PathLength* = 1 and *Step* = 0.2 agrees with that the specimen makes five steps until it reaches the leader. $PRT\in[0,1]$ enables to calculate the perturbation vector $\mathbf{v}_{PRT}$ which indicates whether the active specimen moves to the leader directly or not. $\mathbf{v}_{PRT}$ is defined as

$$\begin{aligned}&\mathbf{v}_{PRT}=\left[v_{PRT,1},v_{PRT,2},\ldots,v_{PRT,r}\right]^T\in\{0,1\}^r\\ &v_{PRT,j}=1 \qquad \text{if } rnd_j<PRT\\ &v_{PRT,j}=0 \qquad \text{else}\\ &j=1,2,\ldots,r\end{aligned} \quad (56)$$

where $rnd_j\in[0,1]$ is a randomly generated number for each coordinate of a specimen. The perturbation vector enters stochasticity to the specimens moving as it is apparent from a subsection below.

There are two termination parameters in the algorithm: *M*, *MinDiv*. The value of *M* means the maximal number of migration rounds defined by the user, and *MinDiv* expresses the selected minimal diversity, i.e. the algorithm running is terminated if

$$\max_{j} \Phi\left(\mathbf{K}_{i,j}\right) - \min_{j} \Phi\left(\mathbf{K}_{i,j}\right) < MinDiv \qquad (57)$$

where a subscript *i* means the current iteration (migration round) and *j* denotes the *j*th specimen in the current population.

Population construction

As mentioned above, population $P = \left\{\mathbf{K}_1, \mathbf{K}_2, ..., \mathbf{K}_{PopSize}\right\}$ has to be generated based on a prototypal specimen controlled by parameters *Rad* and *PopSize*. Let $\mathbf{K}_1$ be the prototypal specimen, then other specimens can generated as

$$\mathbf{K}_j = \mathbf{K}_1 + Rad\left[rnd_1, rnd_2, ..., rnd_r\right]^T, \; j = 2,3,..., PopSize \qquad (58)$$

where $rnd_i \in [-1,1], i = 1,2,...,r$, is a random number. Each specimen in the population is then evaluated by the cost function.

Movement of specimens on the hyperspace

Once the population is generated (or generally after every migration round in the *i*th iteration) the best valued specimen, so called leader, $\mathbf{K}_{i,L}$, which is determined as

$$\mathbf{K}_{i,L} = \arg\min_{j} \Phi\left(\mathbf{K}_{i,j}\right), \; j = 1,2,..., PopSize \qquad (59)$$

All other specimens are then moved towards the leader during the migration round. The movement randomness is given by $\mathbf{v}_{PRT}$. Although the authors of SOMA suggest to calculate $\mathbf{v}_{PRT}$ 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

$$\begin{gathered}\mathbf{K}_{i,j,k} = \mathbf{K}_{i,j,0} + Step\left[(j-1)\left(\mathbf{K}_{i,L} - \mathbf{K}_{i,j,0}\right) + \operatorname{diag}\left(\mathbf{v}_{PRT}\right)\left(\mathbf{K}_{i,L} - \mathbf{K}_{i,j,0}\right)\right] \\ j = 1,2,...PopSize \neq L; k = 1,2,..., \operatorname{round}\left(PathLength / Step\right) - 1\end{gathered} \qquad (60)$$

where $\operatorname{diag}\left(\mathbf{v}_{PRT}\right)$ means the diagonal square matrix with elements of $\mathbf{v}_{PRT}$ on the main diagonal and *k* is the *k*th step in the path of the *j*th specimen in the current population (in *i*th iteration).

The role of $\mathbf{v}_{PRT}$ is evident, for instance, if $\mathbf{v}_{PRT} = [1,1,...,1]^T$, 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.

$$\mathbf{K}_{i+1,j} = \arg\min_{k} \Phi\left(\mathbf{K}_{i,j,k}\right), k = 0,1,..., \operatorname{round}\left(PathLength / Step\right) - 1 \qquad (61)$$

These specimens then generate the new population for the next migration round (iteration).

E. Cost Function Evaluation

For the evaluation of the cost (objective) function, it is crucial to compute the rightmost poles of the feedback system. There are several numerical techniques solving this task. Control system characteristic roots can be found either as a solution of the characteristic equation, or pairs of a characteristic root and the corresponding eigenvector (for reasons of numerical stability).

Poles can be efficiently found e.g. by the quasipolynomial mapping rootfinder (QPMR) algorithm, see [42], which is the gridding method based on the searching of zero points of real and imaginary parts of the characteristic quasipolynomial in a prescribed area.

VII. Conclusion

The presented paper has introduced and presented the three iteration minimization algorithms, namely, Nelder-Mead method, Extended Gradient Sampling Algorithm and Self-Organizing Migration Algorithm as suitable optimization methods for tuning of controllers for LTI-TDS obtained by the algebraic approach in the R_{MS} ring.

In the first part, LTI-TDS systems and models have been presented, followed by various stability notions related to such systems. Then, the R_{MS} ring for a description of and controller design for LTI-TDS has been introduced. The algebraic controller design in this ring for the simple feedback loop satisfying closed-loop stability in R_{MS} sense, reference tracking and step disturbance rejection has followed. The spectral abscissa as the maximum of real parts of the spectrum has been then defined, which has given rise to the objective function.

A thorough description of the three minimization techniques has been the next part of the contribution.

A comparative study based on the principles described in this paper has been presented in a very brief and concise form in [43].

References

[1] O. J. M. Smith, “Closer control of Loops with Dead Time,” *Chemical Engineering Progress*, vol. 53, no.5, pp. 217-219, 1957.

[2] R. Bellman and K. L. Cooke, *Differential-Difference Equations*. New York: Academic Press, 1963.

[3] H. Górecki, S. Fuksa, P. Grabowski, and A Korytowski, *Analysis and Synthesis of Time Delay Systems*. John Wiley & Sons, New York, 1989.

[4] J. K. Hale and S. M. Verduyn Lunel, "Introduction to functional differential equations," *Applied Mathematical Sciences*, vol. 99. New York : Springer-Verlag, 1993.

[5] V. B. Kolmanovskii and A. Myshkis, *Introduction to the Theory and applications of Functional Differential Equations*. Dordrecht: Kluwer Academy, 1999.

[6] J.-P. Richard, "Time-delay systems: an overview of some recent advances and open problems," *Automatica*, vol. 39, no. 10, pp. 1667-1694, 2003.

[7] V. Volterra, "Sur le théorie mathématique des phénomenès héreditaires," *Journal des Mathématiques Pures et Appliquées*, vol. 7, pp. 249-298, 2003.

[8] P. Zítek and J. Hlava, "Anisochronic internal model control of time-delay systems," *Control Engineering Practice*, vol. 9, no. 5, pp. 501-516, 2003.

[9] P. Zítek, "Frequency domain synthesis of hereditary control systems via anisochronic state space," *International Journal of Control*, vol. 66, no. 4, pp. 539-556, 1997.

[10] L. Pekař, R. Prokop, and P. Dostálek, Circuit heating plant model with internal delays, *WSEAS Transaction on Systems*, vol. 8, no. 9, pp. 1093-1104, 2009.

[11] C. Roduner and H. P. Geering, "Model-based multivariable control of a gasoline engine under consideration of receiver dead times," *Automatisierungstechnik*, vol. 44, no. 7, pp. 314-321, 1996. (in German)

[12] J. Morávka and K. Michálek, "Anisochronous model of the metallurgical RH Process," *Transactions of the VŠB – Technical University of Ostrava, Mechanical Series*, vol. 14, no. 2, pp. 91–96, 2008.

[13] M. Ovsík, D. Maňas, M. Maňas, M. Staněk, K. Kyas, M. Bednařík, and A. Mizera, "Effect of beta irradiation on the microhardness of HDPE," in *Recent Researches in Circuits and Systems, Proc. of the 16th WSEAS Int. Conf. on Systems*, Kos, Greece, 2012, pp. 285-288.

[14] S.-I. Niculescu, "Delay effects on stability," in *Lecture Notes in Control and Information Sciences*, vol. 269. Berlin: Springer-Verlag, 2001.

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

[16] J. J. Loiseau, "Algebraic tools for the control and stabilization of time-delay systems," *Annual Reviews in Control*, vol. 24, pp. 135-149, 2000.

[17] L. Pekař, R. Prokop, and R. Matušů, "A stability test for control systems with delays based on the Nyquist criterion," *International Journal of Mathematical Models in Applied Sciences*, vol. 5, no. 6, pp. 1213-1224, 2011.

[18] V. Kučera, Diophantine equations in control - a survey. Automatica, vol. 29, no. 6, pp. 1361-1375, 1993.

[19] P. Zítek and V. Kučera, "Algebraic design of anisochronic controllers for time delay systems," *Int. J. Control*, vol. 76, no. 16, pp. 905-921, 2003.

[20] L. Pekař and R. Prokop, "Some observations about the RMS ring for delayed systems," in *Proc. of the 17th Int. Conf. on Process Control '09*, Štrbské Pleso, Slovakia, 2009, pp. 28-36.

[21] L. Pekař and R. Prokop, "A revised ring of stable and proper quasipolynomial meromorphic functions for LTI-TDS," in *Proc. of the 14th WSEAS Int. Conf. on Math. and Computational Methods in Science and Engineering*, Sliema, Malta, 2012, to be published.

[22] J. K. Hale and S. M. Verduyn Lunel, "Strong stabilization of neutral functional differential equations," *IMA J. Math. Control and Information*, vol. 19, pp. 5-23, 2002.

[23] W. Michiels, K. Engelborghs, P. Vansevenant, and D. Roose, "Continuous pole placement for delay equations," *Automatica*, vol. 38, no. 6, pp. 747-761, 2002.

[24] W. Michiels and T.Vyhlídal, "An eigenvalue based approach for the stabilization of linear time-delay systems of neutral type," *Automatica*, vol. 41, no. 6, pp. 991-998, 2005.

[25] W. Michiels, T. Vyhlídal, and P. Zítek, "Control design for time-delay systems based on quasi-direct pole placement," *Journal of Process Control*, vol. 20, no. 3, pp. 337-343, 2010.

[26] L. Pekař and R. Prokop, "Implementation of a new quasi-optimal controller tuning algorithm for time-delay systems," in *MATLAB for Engineers – Applications in Control, Electrical Engineering, IT and Robotics*, 1st ed., K. Perůtka, Ed. Rijeka, Croatia: InTech, 2011, pp 3-26.

[27] T. Vanbiervliet, K. Verheyden, W. Michiels, and S. Vandewalle, "A nonsmooth optimization approach for the stabilization of time-delay systems," *ESIAM. Control, Optimisation and Calculus of Variations*, vol. 14, no. 3, pp. 478-493, 2008.

[28] J. Burke, A. Lewis, and M. Overton, "A robust gradient sampling algorithm for nonsmooth, nonconvex optimization," *SIAM Journal of Optimization*, vol. 15, no. 3. pp. 751-779, 2005.

[29] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[30] I. Zelinka, "SOMA-self organizing migrating algorithm," in *New Optimization Techniques in Engineering*, G. C. Onwobolu and B. V. Babu, Eds. Berlin: Springer, 2004, pp. 167-217.

[31] P. Zítek and A. Víteček, *Control Design of Time-Delay and Nonlinear Subsystems*. Prague: ČVUT Publishing, 1999. (in Czech)

[32] D. Roose, T. Luzyanina, K. Engelborghs, and W. Michiels, "Software for stability and bifurcation analysis of delay differential equations and applications to stabilization," in *Advances in Time-Delay Systems*, S.-I. Niculescu and K. Gu, Eds., vol. 38, Springer Verlag, New York, 2004, pp. 167-182.

[33] C. I. Byrnes, M. W. Spong, and T. J. Tarn, "A several complex variables approach to feedback stabilization of neutral delay-differential systems," *Mathematical Systems Theory*, vol. 17, no. 1, pp. 97-133, 1984.

[34] L. S. Pontryagin, "On the zeros of some elementary transcendental functions," *Izvestiya Akademii Nauk SSSR*, vol. 6, pp. 115-131, 1942. (in Russian)

[35] P. Zítek and T. Vyhlídal, "Argument-increment based stability criterion for neutral time delay systems," in *Proceedings of the 16th Mediterranean Conference on Control and Automation*, Ajaccio, France, 2008, pp. 824-829.

[36] M. Vidyasagar, *Control System Synthesis: A Factorization Approach*. Cambridge, M. A.: MIT Press, 1985.

[37] J. R. Partington and C. Bonnet, "H_∞ and BIBO stabilization of delay systems of neutral type," *Systems & Control Letters*, vol. 52, no. 3-4, 2004, pp. 283-288.

[38] L. Pekař and R. Prokop, "Non-stepwise Reference Tracking for Time Delay Systems," in *Proceedings of the 12th International Carpathian Conference (ICCC)*, Velké Karlovice, Czech Republic, 2011, pp. 292-297.

[39] T. Vyhlídal, W. Michiels, and P. McGahan, "Synthesis of a strongly stable state-derivative controller for a time delay system using constrained nonsmooth optimization," *IMA Journal of Mathematical Control and Information*, vol. 27, no. 4, pp. 437-455, 2010.

[40] S. Singer and J. Nelder. (2009). Nelder-Mead algorithm. *Scholarpedia*, vol. 4, no. 7, pp. 2928. Available: http://www.scholarpedia.org/article/Nelder-Mead_algorithm.

[41] R. Penrose, "A generalized inverse for matrices," *Proc. of the Cambridge Philosophical Society*, vol. 51, pp. 406-413, 1955.

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

[43] L. Pekař and R. Prokop, "Optimal pole placement for LTI-TDS via some advanced iterative algorithms – part II: application," in *Recent Researches in Circuits and Systems, Proc. of the 16th WSEAS Int. Conf. on Systems*, Kos, Greece, 2012, pp. 202-206.