A SIMPLE DDS ALGORITHM FOR TDS: AN EXAMPLE

Libor Pekař
Tomas Bata University in Zlín
Faculty of Applied Informatic, Department of Automation and Control Engineering
Nad Stráněmi 4511, 76005 Zlín, Czech Republic
E-mail: pekar@fai.utb.cz

KEYWORDS
Delay Dependent Stability, MATLAB®, Model Rationalization, Multiple Time Delay System, Taylor Series Expansion.

ABSTRACT

This paper is aimed at the presentation and simulation verification of a novel simple and fast delay dependent stability (DDS) testing algorithm for linear time-invariant time delay systems (LTI TDS). The algorithm can be used for systems with multiple delays and/or those with many controller parameters. Value ranges of delays and tunable parameters decide about the (exponential) stability of LTI TDS and the goal is to determine such ranges that might not be convex. Our numerical gridding iterative DDS algorithm is based on the finite dimensional approximation of the characteristic quasipolynomial by means of the Taylor series expansion or a Padé-like approach followed by the Regula-Falsi result enhancement. Stability regions are sought through the determination of crossing delays (or parameters) which cause the system stability switching. Numerical simulation experiments performed in MATLAB® environment prove a sufficient accuracy and the practical usability of the proposed DDS algorithm.

INTRODUCTION

Delays in system dynamics constitute one of decisive factors on system stability. They can appear as a natural consequence of plant internal retarded feedbacks (Pekař et al. 2009; Zítek 1983), or more frequently, due to delayed control feedback systems. During recent decades, many approaches and methods on the decision about stability of linear time-invariant time delay systems (LTI TDS) with fixed parameters and delays and have been published, see e.g. (Gu et al. 2003; Michiels and Niculescu 2007; Richard 2003). However, delays and/or controller parameters may vary or can be undetermined, therefore there it emerges the task of the decision about the possible parameters or delays ranges which keeps a LTI TDS stable. We decide between delay-independent stability (DIS) (Delice and Sipahi 2010; Ergenc 2010) which is satisfied if the system is stable for any delay values vector, and delay-dependent stability (DDS) investigating admissible ranges of delays inside which the system remains stable (Olgac et al. 2007; Shao et al. 2013).

Both the problems are usually studied via heavy mathematical tools, such as Ljaponov-Krasovskii matrices. A powerful idea is the searching of crossing delays (or verifying their existence) that make the system switching to stability/instability due to root continuity property via, for instance, the so-called Rekasius transformation (Rekasius 1980) which represents one of practically applicable means for DDS.

The goal of this paper is to investigate and present a simple and computationally fast (by using an advanced polynomial-root finding computer program function) algorithm determining the crossing delays by using the iterative polynomial approximation of the characteristic quasipolynomial. The well-know Taylor series expansion constitutes the main tool for this task; furthermore, the Regula-Falsi principle, utilized consequently, makes the estimation more accurate. The methodology can also be applied for the determination of controller parameters which keeps the feedback system stable with fixed (nominal) delays.

A rather detailed simulation example performed in MATLAB® and Simulink® environment provides the reader with the demonstration and verification of the algorithm and it proves a very good accuracy.

LTI TDS AND ITS STABILITY

The aim of this section is to briefly introduce a LTI TDS model, its exponential stability and some spectral properties.

LTI TDS Model

Let the system be governed by transfer function $G(s)=b(s,\boldsymbol{\tau})/a(s,\boldsymbol{\tau})$, where $a(s,\boldsymbol{\tau})$, $b(s,\boldsymbol{\tau})$ are (retarded) quasipolynomials in $s \in \mathbb{C}$ of the form

$$q(s,\boldsymbol{\tau}) = s^n + \sum_{i=0}^{n-1}\sum_{j=1}^{h_i} q_{ij} s^i \exp\left(-s\sum_{k=1}^{L}\lambda_{ij,k}\tau_k\right) \quad (1)$$

where $q_{ij} \in \mathbb{R}$, $\lambda_{ij,k} \in \mathbb{N}_0$ and $\boldsymbol{\tau} = [\tau_1,..\tau_L]$ stands for independent delays.

Proceedings 29th European Conference on Modelling and Simulation ©ECMS Valeri M. Mladenov, Petia Georgieva, Grisha Spasov, Galidiya Petrova (Editors)
ISBN: 978-0-9932440-0-1 / ISBN: 978-0-9932440-1-8 (CD)

Spectral Properties and Stability

The spectrum of a such system is infinite, and if there are no common zeros of the numerator and denominator, roots $\sigma : a(\sigma, \boldsymbol{\tau}) = 0$ agree with system poles (characteristic values).

Definition 1. The spectral abscissa is

$$\alpha(\boldsymbol{\tau}) := \boldsymbol{\tau} \mapsto \max \operatorname{Re}(\sigma : m(\sigma, \boldsymbol{\tau}) = 0)$$

Property 1. Isolated poles behave continuously and smoothly with respect to $\boldsymbol{\tau}$ on $\mathbb{C}$; however, the function $\alpha(\boldsymbol{\tau})$ may be nonsmooth or even non-Lipschitz at a finite number of points (Vanbiervliet et al. 2008).

Definition 2. The system (1) is exponentially stable if $\alpha(\boldsymbol{\tau}) < 0$ with a fixed $\boldsymbol{\tau}$.

In the light of Property 1 and Definition 2, the stability can switch if the rightmost pole crosses the imaginary axis at crossing frequencies $\boldsymbol{\Omega} := \{\omega \in \mathbb{R}_{0+} : a(\mathrm{j}\omega, \overline{\boldsymbol{\tau}}) = 0\}$ for some corresponding crossing delays $\overline{\boldsymbol{\tau}}$.

Definition 3. The system (1) is DIS if and only if $a(s, \boldsymbol{\tau}) \neq 0$ for any $s \in \mathbb{C}_{0+}$ and $\boldsymbol{\tau} \in \mathbb{R}_{0+}^{L}$. It is DDS if and only if $a(s, \boldsymbol{\tau}) \neq 0$ for all $s \in \mathbb{C}_{0+}$ and some open and bounded sets $\boldsymbol{\tau}_i \in \mathbb{R}_{0+}^{L} \setminus \infty$, $i = 1,2,...$

Hence, if the system is DIS, then $\alpha(0) < 0$.

Lemma 1. A quasipolynomial $q(s, \overline{\boldsymbol{\tau}})$ has roots $s = \mathrm{j}\omega$ if and only if the polynomial

$$\hat{q}(s, \overline{\boldsymbol{\tau}})\Big|_{\exp(-\tau_i s) \mapsto \frac{1 - T_i s}{1 + T_i s}} \tag{2}$$

has the same roots for some $\mathbf{T} := [T_1, T_2, ..., T_L]$, $T_i \in \mathbb{R}$.

Transformation (2) is called the exact Rekasius transformation (Rekasius 1980) and it is widely used to determinate the set $\{\boldsymbol{\Omega}_l, \mathbf{T}_l\}, l \in \mathbb{N}$, $\boldsymbol{\Omega} = \bigcup_l \boldsymbol{\Omega}_l$, and, consequently, $\boldsymbol{\Omega}, \{\overline{\boldsymbol{\tau}}_l\}$ for DDS; however, such algorithms are mostly computationally heavy (Ergenc 2010).

DDS ALGORITHM

Now we intent to present our novel simple iterative gridding algorithm that estimates crossing delays by means of a rational approximation yet without the Rekasius transformation. The successive calculation of the well-known Taylor series expansion (in the neighborhood of the current leading root) followed by the (linear) Regula-Falsi zero point estimation is used to determine sets $\boldsymbol{\Omega}, \{\overline{\boldsymbol{\tau}}_l\}$. Polynomial roots then can be easily and quickly computed by means of a standard software tool, e.g. in MATLAB®, for fixed delay values. The gridding procedure ensures a sufficiently fast yet accurate crossing delays approximation even if the number of polynomial zeros seeking calculations is high.

Preliminaries

For a given characteristic quasipolynomial $a(s, \boldsymbol{\tau})$ with fixed $\boldsymbol{\tau}$ let us introduce the approximate polynomial

$$\hat{a}(s, \boldsymbol{\tau}) = \sum_{i=0}^{\hat{n}} \hat{a}_i(\boldsymbol{\tau}) s^i \tag{3}$$

of the appropriate order, for which it holds that

$$\left.\frac{\partial^k a(s, \boldsymbol{\tau})}{\partial s^k}\right|_{s=s_0} = \left.\frac{\partial^k \hat{a}(s, \boldsymbol{\tau})}{\partial s^k}\right|_{s=s_0}, k = 0,1,..,\mu \tag{4}$$

$s_0 \in \mathbb{C}$, where $\mu = \hat{n} + 1$ holds for the Taylor series expansion, and $\mu = \hat{n}$ with $\hat{a}_{\hat{n}} = 1$ is called the Padé-like approach for objectives of this paper. Note that the set (4) can also be expressed in a linear matrix form to find the solution directly by matrix operations. Since $a(s, \boldsymbol{\tau})$ is the analytic function, $\hat{a}(\cdot) \to a(\cdot)$ for $\hat{n} \to \infty$. With respect to (2), it can be set $\hat{n} = n + L$.

The second leading idea stems from the successive computation of $\hat{a}(s, \boldsymbol{\tau})$ in its current leading zero s_0 to obtain a descending sequence $\left\{a_{(k)}(\hat{s}_k, \boldsymbol{\tau})\right\}_{k=1}^{k_{\max}}$, where $\hat{s}_k := \{s : \hat{a}(s, \boldsymbol{\tau}) = 0, |s - \hat{s}_{k-1}| \to \min\}$, ideally uniformly converging to the zero. The value of $\hat{s}_{k_{\max}}$ then represents the estimation of the leading system pole.

Once the imaginary axis is skipped, the value of the crossing delay is approximated via the Regula-Falsi linear interpolation form the last two eventual values of $\hat{s}_{k_{\max}}$.

The Algorithm

Input. For the given $a(s, \boldsymbol{\tau})$, define a $L \times N$ grid by the discretization $\tau_{k,j+1} = \tau_{k,j} + \Delta\tau_{k,j}$, $\tau_{k,0} = 0$, $k = 1...L$, $j = 0...N-1$, introduce $\hat{a}(s, \boldsymbol{\tau})$ as in (3) of the order $\hat{n} = n + L$ with unknown $\hat{a}_i(\boldsymbol{\tau})$, initialize the counter $i = 1$ and choose the admissible error $\varepsilon > 0$.

Step 1. Compute the initial leading pole

$$\hat{s}_{0,...,0} = s_{0,...,0} := \{s \to \min(|\operatorname{Re} s_k|) : a(s_k, \mathbf{0}) = 0\}$$

of the delay-free system exactly.

Step 2. For ($j_1 = 0...N$, for ($j_2 = 0...N$, for... (for $j_L = 0...N$, do Steps 3-9 of this algorithm))). Hence, the number of L inner (nested) for-loops is performed.

Step 3. If not $j_1 = j_2 = ... = j_L = 0$, do the following:
Set $\boldsymbol{\tau} = \left[\tau_{1,j_1}, \tau_{2,j_2} ..., \tau_{L,j_L}\right]$, $M = \max k : j_k \neq 0$. Consequently set

$$\begin{aligned}\hat{s}_{old} &= \hat{s}_{j_1,...,j_{M-1},j_M-1,0...0} \\ \boldsymbol{\tau}_{old} &= \left[\tau_{1,j_1}, ..., \tau_{M-1,j_{M-1}}, \tau_{M,j_M-1}, \tau_{M+1,0}, ..., \tau_{L-1,0}, \tau_{L,0}\right]\end{aligned}$$

and $\hat{s}_0 := \hat{s}_{old}$.
Step 4. Compute $\hat{a}(s, \boldsymbol{\tau})$ according to (4) in $\hat{s}_0$. Define $\hat{s}_1 := \left\{s \to \min|s - \hat{s}_0| : \hat{a}(s, \boldsymbol{\tau}) = 0\right\}$.

Step 5. While $|\hat{s}_1 - \hat{s}_0| > \varepsilon$, set $\hat{s}_0 = \hat{s}_1$ and execute Step 4.

Step 6. Set $\hat{s}_{new} := \hat{s}_1$. If $\operatorname{sgn}(\operatorname{Re}\hat{s}_{new}) = \operatorname{sgn}(\operatorname{Re}\hat{s}_{old})$, set

$$\hat{s}_{j_1,...,j_L} = \hat{s}_{j_1,...,j_{M-1},j_M,0...0} := \hat{s}_{new}$$

and the inner loop is finished; else, $i = i + 1$ and go to Step 7.

Step 7. Compute the Regula-Falsi approximation: Set

$$\overline{\tau}_M = \tau_{M,j_M-1} - \operatorname{Re}\hat{s}_{old}\left(\tau_{M,j_M} - \tau_{M,j_M-1}\right) / \left(\operatorname{Re}\hat{s}_{new} - \operatorname{Re}\hat{s}_{old}\right)$$

and perform Step 8.

Step 8. For $k = M - 1,...,1$ do:

If $j_k \neq 0$, set $\hat{s}_0 := \hat{s}_{new}$

$$\begin{aligned}\boldsymbol{\tau}_{old} &= \left[\tau_{1,j_1}, ..., \tau_{k,j_k-1}, \overline{\tau}_{k+1}, ..., \overline{\tau}_M, 0, ..., 0\right] \\ \boldsymbol{\tau} &= \left[\tau_{1,j_1}, ..., \tau_{k,j_k}, \overline{\tau}_{k+1}, ..., \overline{\tau}_M, 0, ..., 0\right]\end{aligned}$$

and compute the leading root $\hat{s}_1$ of $\hat{a}(s, \boldsymbol{\tau}_{old})$ in the neighborhood of $\hat{s}_0$ as in Steps 4-5. Update values $\hat{s}_{old} := \hat{s}_1$ and $\hat{s}_0 := \hat{s}_1$. Similarly, calculate $\hat{a}(s, \boldsymbol{\tau})$ in $\hat{s}_0$, find the leading root $\hat{s}_1$ and update the value $\hat{s}_{new} := \hat{s}_1$. The Regula-Falsi then yields

$$\overline{\tau}_k = \tau_{k,j_k-1} - \operatorname{Re}\hat{s}_{old}\left(\tau_{k,j_k} - \tau_{k,j_k-1}\right) / \left(\operatorname{Re}\hat{s}_{new} - \operatorname{Re}\hat{s}_{old}\right)$$

When the loop is finished, go to Step 9.

Step 9. Compute the leading zero $\hat{s}_1$ of $\hat{a}(s, \overline{\boldsymbol{\tau}}_i)$ in the vicinity of $\hat{s}_0 := \hat{s}_{new}$ according Steps 4-5, where $\overline{\boldsymbol{\tau}}_i = \left[\overline{\tau}_1, ...\overline{\tau}_M, 0, ...0\right]$ represents an estimated crossing delay vector, and finally set $\hat{s}_{j_1,...,j_L} := \hat{s}_1$. with the imaginary part (i.e. the crossing frequency) $\omega_i \in \hat{\boldsymbol{\Omega}}$ as the estimation of $\boldsymbol{\Omega}$.

Output. Sets $\{\overline{\boldsymbol{\tau}}_i\}$, $\hat{\boldsymbol{\Omega}}$, $\hat{\boldsymbol{\Sigma}} := \left\{\hat{s}_{j_1,...,j_L}\right\}$.

Remarks. It is worth noting that the problem can emerge in Steps 4, 5 and 9 of the algorithm if $|\hat{s}_1 - \hat{s}_0| > \delta$ for some $\delta > 0$ and any grid step length $\Delta\boldsymbol{\tau}$, see Property 1, due to a stepwise discontinuity in $\alpha(\boldsymbol{\tau})$. We omit it during practical computing since such an abrupt jump in the leading zeros near the imaginary axis is very rare.

When making the estimation of $\overline{\boldsymbol{\tau}}_i$ more precise in Step 8, there might come to pass $\operatorname{sgn}(\operatorname{Re}\hat{s}_{new}) = \operatorname{sgn}(\operatorname{Re}\hat{s}_{old})$; however, it is no problem since both the roots are considered to be sufficiently close to the imaginary axis such that the Regula Falsi works well.

An attentive reader can notice that rearranging lops of Step 8 can lead to different points on the stability border, but these cases have not been studied in this paper.

As mentioned above, although nested loops (in Step 2) may lead up to the number of N^L leading pole calculations, the algorithm remains fast due to a rapid evaluation of polynomial roots e.g. in MATLAB®.

SIMULATION EXAMPLE

Let the controlled plant be governed by the transfer function

$$G_P(s) = \frac{Y(s)}{U(s)} = \frac{0.2\exp(-(\tau_1 + \tau_2)s)}{s^2\left(s^2 - \exp(-\tau_2 s)\right)} \qquad (5)$$

expressing the model of a skater on the swaying bow where $u(t)$ is the input power and $y(t)$ stands for the output angle deviation. Delays τ_1, τ_2 mean the skater's and servo latencies, respectively, the nominal values of which read $\tau_1 = 0.3, \tau_2 = 0.1$ (Zítek et al. 2008).

Consider the following finite-dimensional linear controller

$$G_C(s) = \frac{\sum_{i=0}^{3} q_i s^i}{s^3 + \sum_{i=0}^{2} p_i s^i} \qquad (6)$$

Via an pole-placement model matching tuning algorithm (PPSA), it is possible to find suboptimal controller parameters' values for nominal delays as

$$p_0 = 35370.9, p_1 = 2168.4, p_2 = 1578.8, q_0 = 18678.2,$$
$$q_1 = 87424, q_2 = 356100.4, q_3 = 275080.6 \tag{7}$$

see (Pekař and Prokop 2013). The values are quite large due to the complexity of delayed plant (5). Note that it is not possible to take a lower-degree controller due to the 4th-order unstable plant. Stabilizing properties of controller (6) on plant (5) with parameters (7) are demonstrated via the step response in Figure 1.

The characteristic quasipolynomial, hence, reads

$$a(s,[\tau_1,\tau_2]) = s^2\left(s^2 - \exp(-\tau_2 s)\right)\left(s^3 + \sum_{i=0}^{2} p_i s^i\right) + 0.2\exp(-(\tau_1+\tau_2)s)\left(\sum_{i=0}^{3} q_i s^i\right) \tag{8}$$

Since it is easy to show (e.g. by means of function *roots* in MATLAB®) that polynomial $a(s,[0,0])$ has the spectral abscissa $\alpha([0,0]) = 0.1323$, the feedback system with both variable delays can not be DIS, see Definition 3. Similarly, if the value $\tau_2 = 0.1$ is fixed, we have $\alpha([0,0.1]) = 0.077$, which can be verified e.g. by using of the QPmR – QuasiPolymial matrix Rootfinder (Vyhlídal and Zítek 2014). It means that (8) can not be DIS even for $\tau_1 = 0.1$. However, the setting $\tau_1 = 0.3$ yields $\alpha([0.3,0]) = -0.392$, which indicates that the feedback control system might be DIS.

To verify or deny DIS by using another test, let us apply the methodology (Delice and Sipahi 2010). The chain of signs of a specific discriminant is the conclusion of the algorithm. If the system is DIS, this chain does not include any sign change. For $a(s,\boldsymbol{\tau})$ with both variable delays, it is possible to obtain the resulting chain as [1,1,1,1,1,...1,-1,1,-1,1] that clearly has some sign changes.

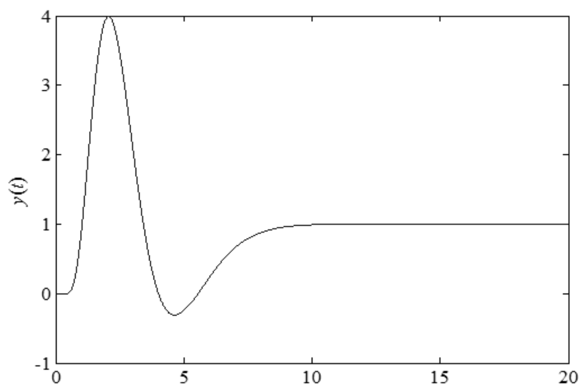

Figure 1 : Step Response of the Feedback Loop with (5) Controlled by (6) with Parameters (7)

The procedure can not be applied for variable τ_1 or τ_2 directly because of fixed exponential terms, see (Delice and Sipahi 2010) for details; therefore, DIS has been tested after the use of the well-known 1st-order Padé approximation applied to separate delay terms. The results for fixed nominal τ_1, τ_2 are the following

$$[-1,-1,-1,1,-1,1,-1,1,1]$$
$$[-1,-1,-1,1,-1,1,1,-1,1]$$

respectively. It is, hence, reasonable to test DDS for both delays or even a single delay value range.

Thus, the control system is unstable for $\boldsymbol{\tau} = [0,0]$, yet the nominal case $\boldsymbol{\tau} = [0.3,0.1]$ gives the stable feedback. The natural question now is where the stability border line(s) lie(s). To avoid excessive computing time to determine $\tau_{1,\max}, \tau_{2,\max}$, let us use the QPmR with a very rough grid of $\Delta\tau_{\cdot,j} = 0.05$, see Figure 2 where the stable region is indicated by a light-coloured polygon, whereas the dark-coloured one means the unstable feedback system. As a conclusion, let us select two reduced disjunctive testing regions

$$R_1 := \tau_1 \times \tau_2 \in [0.05,0.1] \times [0.05,0.1]$$
$$R_2 := \tau_1 \times \tau_2 \in [0.25,0.3] \times [0.25,0.3]$$

in which our novel DDS algorithm is applied with $\Delta\tau_{\cdot,j} = 0.01$.

Following the algorithm, let the polynomial $\hat{a}(s,\boldsymbol{\tau})$ be of the order $\hat{n} = n + L = 7 + 2 = 9$. Consider the Padé-like approach first, i.e. fix $\hat{a}_9 = 1$. The eventual estimations of the crossing delays and corresponding crossing frequencies for R_1 are given in (9) and (10), respectively. This result and its accuracy is graphically verified by the comparison with the QPmR results of the gridding $\Delta\tau_{\cdot,j} = 0.001$ in Figure 3.

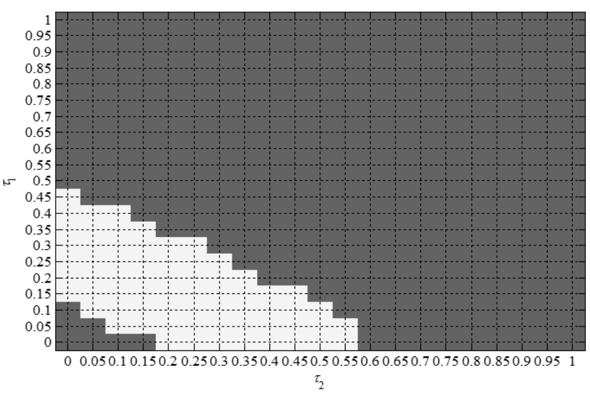

Figure 2 : Stability Regions Computed by Using the QPmR with $\Delta\tau_{\cdot,j} = 0.05$

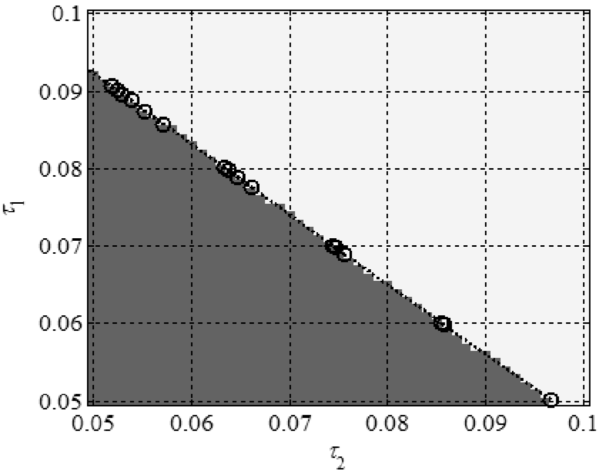

Figure 3 : Stable (Light) and Unstable (Dark) Areas in Region R_1 via the QPmR vs. the Crossing Delays by Means of the DDS Algorithm (Circles)

$$\left\{\bar{\boldsymbol{\tau}}_{R_1,i}\right\}_{i=1}^{16}=\left\{\begin{array}{l}[0.050066,0.096637],[0.059821,0.085782],\\ [0.060074,0.085512],[0.06895,0.075645],\\ [0.069761,0.074779],[0.070074,0.074446],\\ [0.077563,0.066112],[0.078837,0.064745],\\ [0.079698,0.06383],[0.080068,0.063439],\\ [0.085744,0.057106],[0.087409,0.055304],\\ [0.088726,0.053896],[0.089634,0.052934],\\ [0.090057,0.05249],[0.090597,0.051873]\end{array}\right\} \quad (9)$$

$$\hat{\boldsymbol{\Omega}}_{R_1}=\left\{\begin{array}{l}3.848074,3.850265,3.850279,3.852162,\\ 3.852201,3.85221,3.853773,3.853848,\\ 3.853869,3.85387,3.855079,3.855223,\\ 3.855271,3.855271,3.855262,3.855425\end{array}\right\} \quad (10)$$

For R_2, the final values returned by the algorithm read

$$\left\{\bar{\boldsymbol{\tau}}_{R_2,i}\right\}_{i=1}^{8}=\left\{\begin{array}{l}[0.269982,0.299214],[0.279943,0.285429],\\ [0.280154,0.285125],[0.289974,0.271378],\\ [0.290363,0.27081],[0.29119,0.269596],\\ [0.299954,0.257242],[0.300082,0.257054]\end{array}\right\} \quad (11)$$

$$\hat{\boldsymbol{\Omega}}_{R_2}=\left\{\begin{array}{l}1.378578,1.373056,1.372952,1.367779,\\ 1.3676,1.367226,1.362806,1.36275\end{array}\right\} \quad (12)$$

The estimation of crossing delays (11) calculated by using our DDS algorithm compared with the QPmR is displayed in Figure 4. Note that the initial estimation of the leading root for nonzero delays has been made by the QPmR.

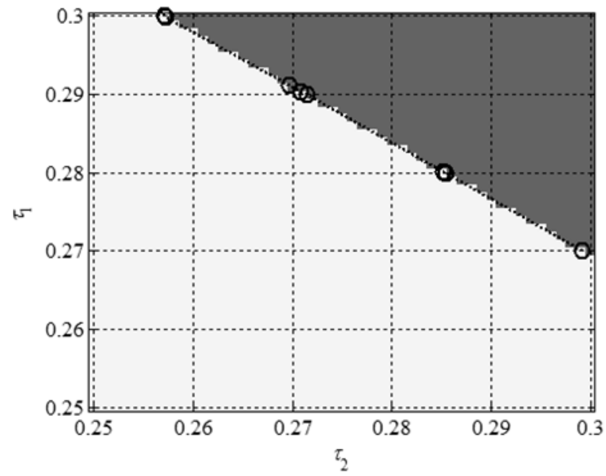

Figure 4 : Stable (Light) and Unstable (Dark) Areas in Region R_2 via the QPmR vs. the Crossing Delays by Means of the DDS Algorithm (Circles)

Note that other values of $\hat{n}$ have been examined as well, and we have found out that the algorithm works with the Padé-like rationalization for $\hat{n}\in\{7,8,9\}$ in the sense that the estimation of $\hat{s}_1$ in the vicinity of $\hat{s}_0$ made in Steps 4 and 5 converges.

As second, the DDS algorithm with the Taylor series expansion, i.e. $\hat{a}_9 \neq 1$, has been performed to determine the crossing delays and frequencies. Amazingly, these sets are almost identical with (9) – (12); deviations in $\bar{\boldsymbol{\tau}}_{R_1,i}$ are approximately of the order 10^{-12}, those in $\bar{\boldsymbol{\tau}}_{R_2,i}$ are even 10^{-14}. Considering frequencies, errors of the same orders have been obtained. However, the convergence with respect to $\hat{n}$ is better since the algorithm works well for $\hat{n}\geq 2$.

Concluding the simulation example, our simple method seems to be usable for the easy and fast estimation of stable/unstable regions depending on delay values for LTI TDS with multiple delays. Moreover, if the control feedback loop includes some undetermined controller parameters, say $\mathbf{K}$, the characteristic quasipolynomial $a(s,\mathbf{K},\boldsymbol{\tau})$ can be subjected to the algorithm with variable $\mathbf{K}$ rather than delays, which is a more advanced and neglected task (Delice and Sipahi 2010).

CONCLUSIONS

A simply implementable algorithm helping to decide about the exponential stability of time delay systems with respect to delay and/or controller parameters values has been provided to the reader in this paper. The leading idea stems from the gridding iterative rationalization of the characteristic quasipolynomial, the roots of which are computed simply by standard software tools. The accuracy is then enhanced by the use of the Regula-Falsi interpolation (extrapolation).

The decision about stability or the seeking of stable/unstable regions can be performed with respect to varying delays as well as (controller) tunable parameters.

A simulation example presented in the second part of the paper indicates a very good agreement of the algorithm results with those obtained from the quasipolynomial matrix rootfinder algorithm by which quasipolynomial roots in a selected region can be numerically calculated.

There are, naturally, still gaps and imperfections that can be improved and several ways how to extend the DDS algorithm. For instance, another (more accurate) rationalization can be used, faster software tools and hardware equipments may be implemented, a nonlinear zero seeking idea can be utilized rather then Regula-Falsi, etc. A challenging task is to derive the algorithm modification that can overcome the obstacles of neutral LTI-TDS.

ACKNOWLEDGEMENTS

The work was performed with the financial support of the research project NPU I No. MSMT-7778/2014 by the Ministry of Education of the Czech Republic and also by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

REFERENCES

Delice, I.I. and R. Sipahi. 2010. "Controller Design for Delay-Independent Stability of Multiple Time-Delay Systems via Descartes's Rule of Signs". In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *Time Delay Systems* 9, 144-149.

Ergenc, A.F. 2010. "A New Method for Delay-Independent Stability of Time-Delay Systems. In *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *Time Delay Systems* 9, 51-56.

Gu, K.; V.L. Kharitonov; and J. Chen. 2003. *Stability of Time-Delay Systems*. Birkhäuser, Boston.

Michiels, M. and S.-I. Niculescu. 2007. *Stability and Stabilization of Time-Delay Systems*. *Advances in Design and Control* 12. SIAM, Philadelphia.

Olgac, N.; R. Sipahi; and A.F. Ergenc. 2007. "'Delay Scheduling' as Unconventional Use of Time Delay for Trajectory Tracking". *Mechatronics* 17, No. 4-5, 199-206.

Pekař, L.; R. Prokop; and P. Dostálek. 2009. „Circuit Heating Plant Model with Internal Delays". *WSEAS Transactions on Systems* 8, No. 9, 1093-1104.

Pekař, L. and R. Prokop. 2013. "Algebraic Optimal Control in RMS Ring: A Case Study". *International Journal of Mathematics and Computers in Simulation* 7, No. 1, 59-68.

Rekasius, Z.V. 1980. "A Stability Test for Systems with Delays". In *Proceedings Joint Automatic Control Conference* (San Francisco, CA). TP9-A.

Richard, J.P. 2003. "Time-delay Systems: An Overview of Some Recent Advances and Open Problems". *Automatica* 39, No. 10, 1667–1694.

Shao, H; X. Zhu; Z. Zhang; and G. Zhong. 2013. "Delay-Dependent Hinf Control for Systems with Two Additive Time-Vary Delays". *Journal of Control Engineering and Technology* 3, No. 2, 90-97.

Vanbiervliet, T.; K. Verheyden; W. Michiels; and S. Vandewalle. 2008. "A Nonsmooth Optimization Approach for the Stabilization of Time-Delay Systems". *ESIAM: Control, Optimisation and Calculus of Variations* 14, No. 3, 478-493.

Vyhlídal, T. and P. Zítek. 2014. "QPmR - Quasi-Polynomial Root-finder: Algorithm Update and Examples". In *Delay Systems: From Theory to Numerics and Applications*, T. Vyhlídal, J.-F. Lafay, and R. Sipahi (Eds.). Springer, New York, 299-312.

Zítek, P. 1983. „Anisochronic State Theory of Dynamic Systems". *Acta Technica ČSAV* 4.

Zítek, P.; V. Kučera; and T. Vyhlídal. 2008. "Meromorphic Observer-Based Pole Assignment in Time Delay Systems". *Kybernetika* 44, No. 5, 633-648.

AUTHOR BIOGRAPHY

LIBOR PEKAŘ was born in Zlín, Czech Republic, in 1979. He studied Automation and Control Engineering at Tomas Bata University in Zlín where he obtained his MSc. degree in 2005 and the Ph.D. degree in Technical Cybernetics in 2013. Nowadays, he has been working as a senior lecturer at the same institute. His research interests are modelling, simulation, autotuning, optimal tuning and algebraic control of time delay systems. His e-mail address is: `pekar@fai.utb.cz`.