Numerical Gridding Stability Charts Estimation using Quasi-polynomial Approximation for TDS

Libor Pekař
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín, Czechia
pekar@utb.cz

Martin Strmiska
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín, Czechia
strmiska@utb.cz

Mengjie Song
School of Mechanical Engineering
Beijing Institute of Technology
Beijing, China
mengjie.song@bit.edu.cn

Petr Dostálek
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín, Czechia
dostalek@utb.cz

***Abstract*—The aim of this study is to present and summarize our numerical algorithm for the determination of stability charts in the delay space for linear time-invariant time systems with constant delays (TDS), both retarded and neutral ones. The core of algorithm lies in a successive (iterative) approximation of the infinite-dimensional characteristic quasi-polynomial in each grid node of the delay space. This approximation resulting in a polynomial or an exponential polynomial with commensurate delays is made in the neighborhood of the dominant characteristic value (pole) that has recently been estimated in the closest grid node. Two different approximation techniques are presented; namely, continuous-time and discrete-time ones. A complete numerical example for retarded TDS is presented, whereas the approximation issues are highlighted in another example for neutral TDS.**

Keywords—constant delay, numerical method, quasi-polynomial approximation, stability charts, time delay systems

I. INTRODUCTION

Time delay systems (TDS), models, and processes have been popular and intensively studied within a wide range of applications [1]-[4], including epidemic control [5], recently. They incorporate a finite number of parameters and delay values, determining infinitely many system modes characterized by pole loci, i.e., by the positions of the characteristic values. Despite the infinite spectrum, only a subset of the so-called dominant (usually the right-most) roots has a decisive impact on TDS properties. Thus, it is desirable to compute those loci based on known model parameters.

The Lambert W function represents the only known analytic tools for exact calculating of all quasi-polynomial roots [6]; however, the function can be used only for commensurate delays. Besides, some partial analytically derived research results provide explicit quasi-polynomial root loci for commensurate delays by different techniques up to the selected model order [7], [8]. There, however, exist several numerical methods to estimate quasi-polynomial root loci based, e.g., on bifurcation analysis [9], mapping-based algorithms for large-scale computation of quasi-polynomial roots (QPmR) [10], cluster treatment of poles [11], partial- or full-discretization methods [12], [13], Galerkin approximation [14], continuous-time approximation [15], and the continuation property of the characteristic values [16]. Besides, scholars have recently derived rigorous theoretical results determining the complex areas in which roots are located [17].

This work was conducted under The Centre for Security, Information and Advanced Technologies (CEBIA – Tech).

Analyzing its stability represents the most critical study for any dynamic system [18]. In the sense of exponential stability, the imaginary axis constitutes the stability border; thus, one can determine TDS parameters, including delay values, that induce the right-most pole loci (expressed by the spectral abscissa), while the rest of the spectrum resides at the left half-plane. These stability charts (or stability regions) in the parameter or delay spaces form non-convex or disjoint sets in many cases [16], [19], which goes beyond the habitual task of the delay margin (i.e., the smallest one destabilizing delay value) determination [20].

In this contribution, we present a numerical algorithm to search for stability charts in the delay space, partially introduced in [21], [22]. In the former work, the discrete-time version based on the bilinear transformation with pre-warping was proposed; whereas, the latter work proposed the continuous-time technique utilizing the Taylor series based expansion. The algorithm uses an equidistant grid of discrete delays with a given dense in the search space. Both the techniques apply iterative characteristic quasi-polynomial approximation by a polynomial or an exponential polynomial with commensurate delays, which yields dominant pole estimation computing with a low effort. The approximation uses the dominant pole estimation in the nearest grid node. We herein complete and summarize both techniques and provide the reader with an extension to neutral TDS. Two numerical examples are given, one for a retarded TDS, another for neutral one. The latter example primarily demonstrates the proposed quasi-polynomial approximation possibility.

The rest of the contribution is organized as follows: Section II includes basic definitions and theorems about TDS, their characteristic quasi-polynomial, stability, and spectrum. The algorithm framework with the approximation techniques and neutral TDS specificities are given in section III. The illustrative examples can be found in section IV.

II. TDS SPECTRUM AND STABILITY

A. Characteristic Quasi-polynomial and TDS Spectrum

The characteristic equation of a linear time-invariant TDS with constant delays $\Delta(s,\boldsymbol{\tau})=0$ where the characteristic quasi-polynomial reads

$$\Delta(s,\boldsymbol{\tau})=\sum_{i=0}^{n} d_i(s,\boldsymbol{\tau})s^i \tag{1}$$

In (1), $s \in \mathbb{C}$ means the Laplace transform variable, $\boldsymbol{\tau} = \left[\tau_0, \tau_1, \ldots \tau_L\right]^T \in \mathbb{R}^L$ with $0 = \tau_0 < \tau_1 < \ldots < \tau_L$ is the delay vector, and $d_i\left(s, \boldsymbol{\tau}\right) = \sum_{j=0}^{L} d_{ij} \mathrm{e}^{-\tau_j s}$ with $d_{ij} \in \mathbb{R}$ are exponential polynomials, where $d_n\left(s, \boldsymbol{\tau}\right)$ represents the associated exponential polynomial. If $d_n\left(s, \boldsymbol{\tau}\right) \in \mathbb{R}$, the system is retarded; otherwise, it is neutral.

Under some particular conditions [18], the TDS spectrum (i.e., the set of characteristic values or system poles) is

$$\Sigma := \left\{ s : \Delta\left(s, \boldsymbol{\tau}\right) = 0 \right\} \tag{2}$$

The so-called essential spectrum for neutral TDS reads

$$\Sigma_{ess} := \left\{ s : d_n\left(s, \boldsymbol{\tau}\right) = 0 \right\} \tag{3}$$

We do let define the spectral abscissa $\alpha(\cdot) := \sup \mathrm{Re}\, \Sigma$ and the abscissa of the essential spectrum $\gamma(\cdot) := \sup \mathrm{Re}\, \Sigma_{ess}$. Both functions depend on system parameters and delays.

For retarded TDS, only a finite subset of Σ lies in the right half-plane (RHP), poles are isolated and $\mathrm{Im}\, \Sigma$ does not reach for a finite $\mathrm{Re}\, \Sigma$. Pole loci behave continuously with respect to delays; however, α is can be a non-smooth function [23].

Contrariwise, infinitely many characteristic values can be located in the RHP constituting strips parallel to the imaginary axis for neutral TDS. Besides, values of λ can behave discontinuously when even small changes in $\boldsymbol{\tau}$ [24], [25].

B. Exponential and Strong Stability

The exponential stability of a TDS can be expressed as

$$\alpha(\cdot) < \varepsilon \tag{4}$$

where $\varepsilon = 0$ and (any) $\varepsilon > 0$ hold for retarded and neutral systems, respectively [18].

Define λ_δ as the value of λ subjected to infinitesimal changes in $\boldsymbol{\tau}$. Strong stability expresses that λ_δ remains in the RHP, and it can be checked by the condition [24]

$$\sum_{j=1}^{L} \left| d_{nj} \right| < 1 \tag{5}$$

Note that continuous upper bound function $\boldsymbol{\tau} \mapsto \overline{\lambda}_\delta\left(\boldsymbol{\tau}\right)$ on λ_δ can be computed to estimate the true λ_δ [25].

Hence, stability of neutral TDS cannot be judged solely based on (4) but also (5) must be considered. I.e., the factual real part of the right-most pole is given by both α and λ_δ.

III. Stability Charts Searching Algorithm

A. Algorithm Framework

The numerical stability charts searching algorithm is concisely summarized first where details are provided to the reader if necessary. Then, specific aspects of neutral TDS and quasi-polynomial approximations are given.

Algorithm 1. (Stability charts estimation in the delay space)

1) Take (1), set an equidistant mesh grid $\left[\tau_{ij}\right] \in \underbrace{\mathbb{R}^N \times \mathbb{R}^N \times \cdots \times \mathbb{R}^N}_{L}$ in the delay space for the discretization step $\Delta\tau_{\cdot,j} = \tau_{\cdot,j+1} - \tau_{\cdot,j}$ and selected ranges of delay values, and let $\tau_{i0} = 0, i = 1, 2, ..., L$.

2) Compute the right-most pole (or a pair) for $\boldsymbol{\tau} = 0$ exactly.

3) Perform the nested loop, where the outer one goes through 1 to L and the inner one from 0 to N, and in every current grid node $\boldsymbol{\tau}_c$, do:

3a) Approximate Δ by a polynomial or exponential polynomial $\tilde{\Delta}$ in the neighborhood of the right-most pole estimation s_p already made in the closest grid node $\boldsymbol{\tau}_p$ (see subsections III.B and III.C for more detail).

3b) Compute the estimation of the right-most root s_c of $\tilde{\Delta}$. The value of s_p must be sufficiently close to s_c. Evident outliers can be canceled as follows. Define the function

$$\delta^s\left(\boldsymbol{\tau}\right) := \mathrm{Re}\left(\nabla s\left(\boldsymbol{\tau}\right)\right) \tag{6}$$

and the function $\Re^s\left(\boldsymbol{\tau}\right) := \boldsymbol{\tau} \mapsto \mathrm{Re}\, s$ for a particular root s corresponding to delay vector $\boldsymbol{\tau}$ where ∇ means the gradient. Due to the discontinuity of α, function $\delta^s\left(\boldsymbol{\tau}\right) \in \mathbb{R}^L$ might not be defined in some points. Hence, its numerical approximation can be used instead:

$$\tilde{\delta}_i^{s_1}\left(\boldsymbol{\tau}^1\right) = \frac{\Re^{s_1}\left(\boldsymbol{\tau}^1\right) - \Re^{s_0}\left(\boldsymbol{\tau}^0\right)}{\tau_i^1 - \tau_i^0}, i = 1, 2, ..., L \tag{7}$$

where $\boldsymbol{\tau}^1, \boldsymbol{\tau}^0$ are to close delay points and i means the delay number. Then, one may cancel the computed roots of $\tilde{\Delta}$ satisfying

$$\left\{ s : \mathrm{Re}\, s > \Re^{s_p}\left(\boldsymbol{\tau}_p\right) + c\left(\boldsymbol{\tau}_c^T - \boldsymbol{\tau}_p^T\right)\delta^{s_p}\left(\boldsymbol{\tau}_p\right) \right\} \tag{8}$$

for some $c > 1$.

3c) If a crossing of the imaginary axis by the rightmost-pole estimation is detected, compute iteratively the estimation $\boldsymbol{\tau}^0$ of the switching delay (i.e., the delay value that makes the system switching from/to stability/instability)

$${}^{(i+1)}\tau_l^0 = {}^{(i)}\tau_l^0 - \frac{\Re^{s_0}\left(\boldsymbol{\tau}^0\right)}{\delta_l^{s_0}\left(\boldsymbol{\tau}^0\right)}, l = 1, 2, \ldots L \tag{9}$$

where superscript (i) indicates the iteration step, with the initial estimation (10). The dominant pole estimation s_0 update according to step 3b is made in every single iteration step. It eventually gives rise to the switching poles and their

frequencies estimates $\omega_0 = \mathrm{Im}\, s_0$ that are collected in sets $\Sigma_0 := \{s_0\}$ and $\Omega_0 := \mathrm{Im}\,\Sigma_0$, respectively.

$$ {}^{(1)}\boldsymbol{\tau}^0 = \frac{\left|\Re^{s_c}(\boldsymbol{\tau}_c)\right|\boldsymbol{\tau}_p + \left|\Re^{s_p}(\boldsymbol{\tau}_p)\right|\boldsymbol{\tau}_c}{\left|\Re^{s_p}(\boldsymbol{\tau}_p)\right| + \left|\Re^{s_c}(\boldsymbol{\tau}_c)\right|} \tag{10} $$

The set of all switching delays let be $\mathrm{T}_0 := \{\boldsymbol{\tau}_0\}$. It is worth noting that $\boldsymbol{\tau}_0$ is excluded from T_0 if $\left\|\delta^{s_0}(\boldsymbol{\tau}_0)\right\| = 0$, i.e., the pole has not a "sufficient velocity" to cross the imaginary axis.

If the crossing is not detected, the current loop in step 3 is finished, and go to step 3a.

4) Outputs: $\Sigma_0, \Omega_0, \mathrm{T}_0$.

A detailed view of step 3a of Algorithm 1 follows. Two versions of how to get $\tilde{\Delta}$ follow.

B. Approximation Based on Digital Filter Design

The first technique adopts a transition from the continuous-time to the discrete-time space via basic methods used in digital filter design. Delays and *s*-powers in Δ are subjected to two different substitutions, respectively, as follows:

$$ \mathrm{e}^{-\tau s} \leftarrow q^{\frac{\tau}{T_s}} \tag{11} $$

$$ s \leftarrow \frac{\omega_f}{\tan\left(0.5\omega_f T_s\right)}\frac{1-q}{1+q} \tag{12} $$

where q is the backward shift operator expressing the unit delay that corresponds to z^{-1} in the *z*-plane, T_s is a sampling period, and ω_f means the desired frequency for which the frequency responses in the *s*-plane and the *z*-plane coincide. We herein apply $\omega_f = \left|s_p\right|$.

The goal is to get $\tilde{\Delta}(z, \boldsymbol{\tau}_c) = \sum_{i=0}^{\tilde{n}} \tilde{d}_i(\boldsymbol{\tau}_c) z^i$ where $\tilde{d}_i(\boldsymbol{\tau}_c) \in \mathbb{R}$. Non-negative powers of *z* can easily be obtained from powers of $z^{-1} = q$ by the reciprocation. However, it usually holds that $\kappa = \tau_{c,i}/T_s \notin \mathbb{N}$; hence, non-integer *z*-powers can arise from (11). Then, for instance, a quadratic extrapolation of $z^{-\kappa}$ can be used [21]. This extrapolation is calculated in the vicinity of the *z*-plane image z^p of the right-most pole s^p via the known correspondence

$$ z = \mathrm{e}^{T_s s} \leftrightarrow s = \frac{1}{T_s}\log z \tag{13} $$

Formula (12) represents Tustin's transform method with pre-warping to prevent frequency warping.

The setting of T_s represents the crucial yet ambiguous task. A trade-off between the suggested value of the sampling period for the z-transform with respect to the TDS dynamics (given by s_p) and the derivative operator discretization ought to be found. It poses a numerical optimization subtask, the solution of which remains an open problem.

Once $\tilde{\Delta}(z, \boldsymbol{\tau}_c)$ is found, its dominant root z_c can be computed (e.g., using standard Matlab function), yielding the estimation of s_c by applying (13), see also step 3b of Algorithm 1. It must be emphasized that the estimation of $\tilde{\Delta}(z, \boldsymbol{\tau}_c)$ and s_c is computed iteratively with setting ${}^{(1)}s_c = s_p$ until $\left|{}^{(i)}s_c - {}^{(i-1)}s_c\right| < \varepsilon$ for some suitable $\varepsilon > 0$.

C. Approximation Based on Taylor Series

The second technique uses a simple idea of the iterative quasi-polynomial approximation by a polynomial based on the Taylor series expansion of $\Delta(s, \boldsymbol{\tau}_c)$ in ${}^{(i)}s_c$ where ${}^{(1)}s_c = s_p$ again. Thus, the aim is to get polynomial $\tilde{\Delta}(s, \boldsymbol{\tau}_c) = \sum_{i=0}^{\tilde{n}} \tilde{d}_i(\boldsymbol{\tau}_c) s^i$ with $\tilde{d}_i(\boldsymbol{\tau}_c) \in \mathbb{R}$. The expansion is equivalent to the following condition

$$ \left.\frac{\partial^j d(s, \boldsymbol{\tau}_c)}{\partial s^j}\right|_{s={}^{(i)}s_c} = \left.\frac{\partial^j \tilde{d}(s, \boldsymbol{\tau}_c)}{\partial s^j}\right|_{s={}^{(i)}s_c}, \; j = 0,1,..,\tilde{n} \tag{14} $$

the solution of which can be expressed as

$$ \begin{aligned} &\tilde{\mathbf{d}} = \mathbf{A}^{-1}\mathbf{b}, \\ &\mathbf{A} = \left[\frac{(M-1)!}{(M-N)!}\,{}^{(i)}s_c^{M-N}\right] \in \mathbb{C}^{(\tilde{n}+1)\times(\tilde{n}+1)}, \\ &\mathbf{b} = \left[\left.\frac{\partial^{N-1}\Delta(s, \boldsymbol{\tau}_c)}{\partial s^{N-1}}\right|_{s={}^{(i)}s_c}\right] \in \mathbb{C}^{(\tilde{n}+1)} \end{aligned} \tag{15} $$

where *N*, *M* are, respectively, row and column indexes of a particular matrix and $\tilde{\mathbf{d}}^{\mathrm{T}} = \left[\tilde{d}_0, \tilde{d}_1, \ldots, \tilde{d}_{\tilde{n}}\right]$, see [22] for details.

The choice of $\tilde{n}$ is ambiguous. Intuitively, the higher the value is, the better the TDS spectrum can be estimated. Contrariwise, there may then appear artificial roots of $\tilde{\Delta}$, see also (6)-(8) and text around. Inspired by the work [19], we herein set $\tilde{n} = n + L$.

Recall that the estimation of s_c is computed iteratively by recalculation of $\tilde{\Delta}$ in the vicinity of the current dominant root estimation in every single iteration step.

D. Neutral TDS Specificities

When approximating neutral quasi-polynomial and making the decision about exponential stability, one has to be careful about strong stability condition (5), i.e., the sensitivity to infinitesimal changes in $\boldsymbol{\tau}$. If (5) holds (i.e., $\lambda_\delta < 0$), exponential stability is not affected; contrariwise, the system might be exponentially yet not strongly stable. As strong stability is not affected by delay values, it is not reasonable to determine exponential stability charts if (5) does not hold.

The whole neutral TDS spectrum $\tilde{\Sigma}$ is estimated as the union of $\tilde{\Sigma}_{ess}$ given by roots of approximation $\tilde{d}_n(s, \tau_b, \boldsymbol{\tau}_c)$

of $d_n(s,\boldsymbol{\tau}_c)$ and $\tilde{\Sigma}_{pol}$ that includes all roots of a polynomial approximation $\tilde{\Delta}(s,\boldsymbol{\tau}_c)$ of $\Delta(s,\boldsymbol{\tau}_c)$. Let us propose the computation of $\tilde{\Sigma}_{ess}$ first.

Assume $\tilde{d}_n(s,\tau_b,\boldsymbol{\tau}_c)$ as an exponential polynomial with commensurate delays

$$\tilde{d}_n(s,\tau_b,\boldsymbol{\tau}_c)=\sum_{i=0}^{\tilde{L}}\tilde{d}_{n,i}\,\mathrm{e}^{-i\tau_b s} \tag{16}$$

where τ_b is the base delay, $\tilde{L}$ expresses the degree of commensuracy, and $\tilde{d}_{n,i}\in\mathbb{R}$. The form of (16) is motivated by some results [7], [8], in which root loci for neutral TDS with commensurate delays were calculated analytically. The base delay is considered as $\tau_b=\tau_{\min}/n_b$ where $n_b\in\mathbb{N}$, $\tau_{\min}=\min_{\tau_{n,i}>0}(\tau_{n,i}), i=1,2,...,L$. The choice of n_b can be made numerically by a trial-and-reset test, such that the convergence of (16) is satisfied (see the iterative approximation technique below) and taking into account the system dynamics represented by s_c. Indeed, τ_b can be considered as the sampling period as $q=\mathrm{e}^{-\tau_b s}$.

1) Computation of $\tilde{d}_n$ and $\tilde{\Sigma}_{ess}$: The initial linear approximation is made as follows

$$\mathrm{e}^{-\tau s}\approx d_0\,\mathrm{e}^{-\nu\tau_{\min}s}+d_1\,\mathrm{e}^{-(\nu+1)\tau_{\min}s} \tag{17}$$

where $\tau=(\nu+\rho)\tau_{\min}$, $\nu\in\mathbb{N},\rho\in[0,1)$, $d_0=1-\rho, d_1=\rho$. Using $\mathrm{e}^{-\tau_{\min}s}\leftarrow q=z^{-1}$, one gets $\tilde{d}_n(z,\tau_{\min},\boldsymbol{\tau}_c)=\sum_{i=0}^{\tilde{L}}\tilde{d}_{n,i}z^{-i}$, the zeros of which can easily be computed (see subsection III.B), giving rise to the initial essential spectrum approximation in the z-plane. Their images in the s-plane are given by (13) (where $\tau_{\min}$ is used instead of T_s) yielding $\tilde{\Sigma}_{ess}$. Then, ${}^{(1)}s_{c,ess}=\arg\max_i(\mathrm{Re}\,s_i)$ where $s_i\in\tilde{\Sigma}_{ess}$.

Once the initial value ${}^{(1)}s_{c,ess}$ is obtained, n_b can be determined yielding τ_b and $\tilde{d}_n(z,\tau_b,\boldsymbol{\tau}_c)$ via $\mathrm{e}^{-\tau_b s}\leftarrow q=z^{-1}$. As z-powers in $\tilde{d}_n$ might be non-integers, e.g., the quadratic extrapolation [21] giving integer z-powers can be applied. Then the right-most estimate ${}^{(i)}s_{c,ess}$ of the essential spectrum $\tilde{\Sigma}_{ess}$ (in every iteration step) is obtained analogously to the preceding paragraph. Exponential polynomial $\tilde{d}_n$ is iteratively recalculated based on ${}^{(i)}s_{c,ess}$ until $\left|{}^{(i)}s_{c,ess}-{}^{(i-1)}s_{c,ess}\right|<\varepsilon$.

It is worth noting that $s_{p,ess}$ is the estimate of λ. Besides, if $s_i\in\tilde{\Sigma}_{ess}$, then

$$s_k=s_i\pm 2k\pi\tau_b^{-1}\mathrm{j}\in\tilde{\Sigma}_{ess}, k\in\mathbb{N} \tag{18}$$

That is, roots of $\tilde{d}_n(s,\cdot)$ constitute vertical strips (i.e., parallel to the imaginary axis).

Note that d_n does not depend on delays in most cases.

2) Computation of $\tilde{\Delta}(s,\boldsymbol{\tau}_c)$ and $\tilde{\Sigma}_{pol}$: The polynomial approximation $\tilde{\Delta}(s,\boldsymbol{\tau}_c)$ of $\Delta(s,\boldsymbol{\tau}_c)$ is simply made as for a retarded quasi-polynomial according to the techniques introduced either in section III.B or III.C. Then $\tilde{\Sigma}_{pol}:=\{s:\tilde{\Delta}(s,\boldsymbol{\tau}_c)=0\}$ and $s_{c,pol}=\arg\max_i(\mathrm{Re}\,s_i)$ $s_i\in\tilde{\Sigma}_{pol}$, see also steps 1 to 3b) of Algorithm 1. Note that $s_{p,pol}$ from the previous grid node is saved.

When determining $\tilde{\Sigma}_{pol}$, one may consider regions in which neutral TDS poles definitely do not exist, see, e.g. [17].

The summary of step 3 of Algorithm 1 for neutral TDS follows.

Algorithm 2. (Dominant pole determination for neutral TDS)

1) If (5) does not hold for (1), abandon the algorithm; else, go to step 2.

2) Compute ${}^{(1)}s_{c,ess}$ using (17) and (13) according to subsection III.D.1. Set n_b, τ_b, $\varepsilon>0$, and $i=1$.

3) Compute polynomial $\tilde{d}_n(z,\tau_b,\boldsymbol{\tau}_c)$ via the quadratic extrapolation in ${}^{(i)}s_{c,ess}$ and then $\tilde{d}_n(s,\tau_b,\boldsymbol{\tau}_c)$ using (13).

4) Compute spectrum $\tilde{\Sigma}_{ess}$ of $\tilde{d}_n(s,\tau_b,\boldsymbol{\tau}_c)$, find ${}^{(i+1)}s_{c,ess}$ such that $\left|{}^{(i+1)}s_{c,ess}-{}^{(i)}s_{c,ess}\right|\to\min$. Set $i=i+1$. If $\left|{}^{(i)}s_{c,ess}-{}^{(i-1)}s_{c,ess}\right|>\varepsilon$, go to step 3; else, go to step 5.

5) Set $\lambda=\mathrm{Re}({}^{(i)}s_{c,ess})$ and add roots to $\tilde{\Sigma}_{ess}$ via (18).

6) Compute $\tilde{\Delta}(s,\boldsymbol{\tau}_c)$ and $\tilde{\Sigma}_{pol}$ iteratively based on known $s_{p,pol}$ according to subsection III.B or III.C.

7) Output: $\tilde{\Sigma}=\tilde{\Sigma}_{ess}\cup\tilde{\Sigma}_{pol}$ with $s_c=\arg\max\mathrm{Re}(\tilde{\Sigma})$.

Despite that poles left from λ might be neglected when computing $\tilde{\Sigma}_{pol}$, due to the decisive impact of low-frequency poles on the TDS dynamics, we recommend also considering roots of $\tilde{\Delta}(s,\boldsymbol{\tau}_c)$, the real part of which is left from yet sufficiently close to λ.

IV. Numerical Examples

A. Example 1

Consider an exponentially unstable retarded TDS model governed by the transfer function

$$G(s)=\frac{0.2}{s^2\left(s^2-\mathrm{e}^{-\tau_2 s}\right)}\mathrm{e}^{-(\tau_1+\tau_2)s} \tag{19}$$

which describes movements of a skater on a remotely controlled swaying bow [21], [22], [26]. Nominal delays and the corresponding spectral abscissa are $\tau_1 = 0.3\,\mathrm{s}$, $\tau_2 = 0.1\,\mathrm{s}$, $\alpha = 0.953$, respectively. When applying a finite-dimensional linear controller [27], the closed-loop characteristic quasi-polynomial reads

$$\Delta(s,\boldsymbol{\tau}) = \left(1.64s^3 + 21.3s^2 + 5.25s + 1.12\right)\cdot 10^6\, \mathrm{e}^{-(\tau_1+\tau_2)s} + s^2\left(s^3 + \left(0.47s^2 + 0.64s + 10.56\right)\cdot 10^6\right)\left(s^2 - \mathrm{e}^{-\tau_2 s}\right) \tag{20}$$

Surprisingly, the delay-free case $\boldsymbol{\tau} = [0,0]^T$ is unstable. Following Algorithm 1, we set $\Delta\tau = 0.01$ within the range $\boldsymbol{\tau} \in [0,0.7]\times[0,0.7]$. Note that the value of $\Delta\tau$ affects the initial precision of the stability border determination and the density of the borderline points. The higher the value is, the rougher the estimation is and the sparser the border points are distributed. Contrariwise, the computation time is proportional to $(\Delta\tau)^{-2}$. A particular setting is on the user's decision.

It was eventually set $T_s = 1/(8\omega_f) = 1/(8|s_p|)$, $\varepsilon = 10^{-6}$ for the discrete-time technique (see subsection III.B). The eventual stability charts for both the approximation techniques and the CTCR algorithm [19] are displayed in Fig. 1. As differences are indistinguishable by sight, a numerical comparison is provided in Table 1, in which $\alpha_j(\boldsymbol{\tau}_0)$ (where $j = 1$ holds for switching delay points computed directly by Algorithm 1, while $j = 2$ stands for those obtained by their quadratic interpolation) are abscissae of roots s_i of $\Delta(s,\boldsymbol{\tau}_0)$, $\Delta\omega(\boldsymbol{\tau}_0) = |\omega_0| - |\mathrm{Im}\, s_i|$, and $\boldsymbol{\tau}_0 \in \mathrm{T}_0, \omega_0 \in \Omega_0$. Lower and upper bounds on particular values are displayed.

As can be seen from Fig. 1 and Table 1, Algorithm 1 with both the quasi-polynomial approximation techniques gives results almost comparable to the well-established CTCR paradigm. However, its disadvantage is that the appropriate setting of some parameters remains unclear.

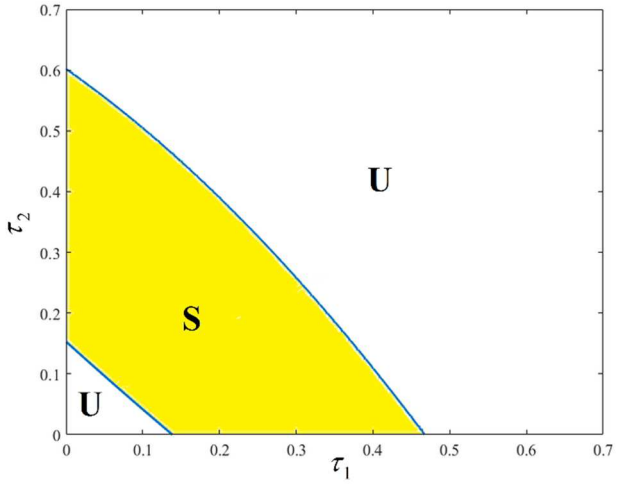

Fig. 1. Stability charts of Example 1 (S-stable area, U-unstable area).

TABLE I. NUMERICAL RESULTS COMPARISON (EXAMPLE 1)

Method	Comput. time	$\lvert\alpha_1(\boldsymbol{\tau}_0)\rvert$	$\lvert\alpha_2(\boldsymbol{\tau}_0)\rvert$	$\lvert\Delta\omega(\boldsymbol{\tau}_0)\rvert$
Algorithm 1 (Subsect. III.B)	748 s	$3.8\cdot10^{-12}$ $1.1\cdot10^{-6}$	$2.4\cdot10^{-5}$ $8.5\cdot10^{-3}$	$5.3\cdot10^{-12}$ $2.4\cdot10^{-8}$
Algorithm 1 (Subsect. III.C)	1054 s	$5.1\cdot10^{-12}$ $1.2\cdot10^{-6}$	$1.7\cdot10^{-5}$ $1.0\cdot10^{-2}$	$2.0\cdot10^{-11}$ $1.3\cdot10^{-9}$
CTCR	465	$9.6\cdot10^{-20}$ $1.2\cdot10^{-15}$	$1.0\cdot10^{-5}$ $7.9\cdot10^{-3}$	$1.9\cdot10^{-11}$ $8.6\cdot10^{-8}$

B. *Example 2*

This example aims to test Algorithm 2 for neutral quasi-polynomial (21) with a single delay τ and the fixed exponential polynomial $d_n(s)$.

$$\Delta(s,\tau) = \left(1 + 0.5\mathrm{e}^{-0.9s} - 0.4\mathrm{e}^{-2/3\pi s}\right)s + 0.3 - 2\mathrm{e}^{-\tau s} + 2\mathrm{e}^{-2\tau s} \tag{21}$$

Condition (5) holds, i.e., 0.9 < 1. The eventual approximation $\tilde{d}_n(s,0.3,\cdot)$ with commensurate delays of $d_n(s)$ reads

$$\tilde{d}_n(s,0.3,\cdot) = 1 + 0.5\mathrm{e}^{-0.9s} + 10^{-3}\begin{pmatrix}(-2.4+3.1\mathrm{j})\mathrm{e}^{-1.8s} \\ +(-399.6-7.1\mathrm{j})\mathrm{e}^{-2.1s} + (2+2.9\mathrm{j})\mathrm{e}^{-2.4s}\end{pmatrix} \tag{22}$$

The strong stability condition still holds (0.9071 < 1), and step 5 of Algorithm 2 yields $\gamma = -0.107$. Notice that (22) has complex-valued coefficients. This implies that the corresponding roots are distributed asymmetrically to the imaginary axis.

By applying the continuous-time approximation of the whole quasi-polynomial (21) with $\Delta\tau = 0.01$, $\tilde{\Delta}(s,\tau)$ is obtained for every single delay value. Distances $\tilde{\Delta}(s,\tau)$ zeros from those of $\Delta(s,\tau)$ (computed using the QPmR Matlab function [10]) vary within the range $\left[8.5\cdot10^{-18}, 3.8\cdot10^{-10}\right]$.

The stabilizing delay value ranges are eventually computed as $\tau_s \in (0,0.0821)\cup(0.2789,0.6734)$ with $\Omega_0 = \{0.433, 2.848, 3.198\}$.

CONCLUSIONS

A gridding-based numerical algorithm for the computation of stability charts in the space of delays has been proposed. Two computational methods for finite-dimensional approximation of the characteristic quasi-polynomial have been presented, which has represented the core of the algorithm. Both the methods are based on simple mathematical operations known from introductory courses in mathematical analysis or from the design of digital filters. Besides the habitual retarded models, specificities of neutral delay models and their sensitivity to small delay variations have been discussed as well. Not only has the delay margin been determined, but the whole stability image within the

given delay range has been provided. Two concise numerical examples have concluded the text body. These examples have shown that the proposed method is almost comparable to the well-established CTCR algorithm.

The crucial disadvantage and an open problem for future research consist in the necessity of the appropriate setting of algorithm-control parameters, which has an ambiguous solution so far.

REFERENCES

[1] E. Fridman, Introduction to Time-Delay Systems: Analysis and Control. Springer. Cham: Springer, 2014.

[2] Q. Gao and H. R. Karimi, Stability, Control and Application of Time-Delay Systems. Oxford: Butterworth-Heinemann, Elsevier, 2019.

[3] Y. Zhang, H. Zhao, and Q. Zhang, "The modeling and control of a singular biological economic system with time delay in a polluted environment," Discrete Dyn. Nat. Soc., vol. 2016, art. no. 5036305, October 2016.

[4] O. V. Popovych, B. Lysyansky, and P. A. Tass, "Closed-loop deep brain stimulation by pulsatile delayed feedback with increased gap between pulse phases," Sci. Reports, vol. 7, art. no. 1033, April 2017.

[5] L.-S. Young, S. Ruschel, S. Yanchuk, and T. Pereira, "Consequences of delays and imperfect implementation of isolation in epidemic control," Sci. Reports, vol. 9, art. no. 3505, March 2019.

[6] N. Choudhary, J. Sivaramakrishnan, and I. N. Kar, "Analysis of higher order time delay systems using Lambert W function," J. Dyn. Syst. Meas. Control, vol. 139, art. no. 114506, November 2017.

[7] C. Bonnet, A. R. Fioravanti, and J. R. Partington, "Stability of neutral systems with commensurate delays and poles asymptotic to the imaginary axis," SIAM J. Control Optim., vol. 49, pp. 498-516, March 2011.

[8] L. H. V. Nguyen and C. Bonnet, "H1-stability analysis of various classes of neutral systems with commensurate delays and with chains of poles approaching the imaginary axis," in Proc. 54th IEEE Conf. on Decision and Control, Osaka, Japan, pp. 6416-6421, 2015.

[9] K. Engelborghs, T. Luzyanina, and D. Roose. "Numerical bifurcation analysis of delay differential equations using DDE-BIFTOOL," ACM Trans. Math. Softw., vol. 28, pp. 1–21, March 2002.

[10] T. Vyhlídal and P. Zítek, "Mapping based algorithm for large-scale computation of quasi-polynomial zeros," IEEE Trans. Autom. Control, vol. 54, pp. 171–177, January 2009.

[11] N. Olgac and R. Sipahi, "An exact method for the stability analysis of time-delayed linear time-invariant (LTI) systems," IEEE Trans. Autom. Control, vol. 47, pp. 793–797, May 2002.

[12] Q. Mou, H. Ye, and Y. Liu, "Enabling highly efficient eigen-analysis of large delayed cyber-physical power systems by partial spectral discretization," IEEE Trans. Power Syst., vol. 35, pp. 1499-1508, March 2020.

[13] C. Ozoegwu and P. Eberhard, "Stability analysis of multi-discrete delay milling with helix effects using a general order full-discretization method updated with a generalized integral quadrature," Mathematics, vol. 8, art. no. 1003, June 2020.

[14] S. S. Kandala, S. Samukham, T. K. Uchida, and C. P. Vyasarayani, "Spurious roots of delay differential equations using Galerkin approximations," J. Vib. Control, vol. 26, pp. 1178-1184, August 2020.

[15] J.-Q. Sun and B. Song, "Control studies of time-delayed dynamical systems with the method of continuous time approximation," Commun. Nonlinear Sci. Numer. Simul., vol. 14, pp. 3933–3944, November 2009.

[16] S. Samukham, T. K. Uchida, and C. P. Vyasarayani, "Fast generation of stability charts for time-delay systems using continuation of characteristic roots," ASME J. Comput. Nonlinear Dynam., vol. 15, art. no. 111008, November 2020.

[17] T. Cai and H. Zhang, "Novel properties of spectrum for stability test of linear delay differential systems," IEEE Access, vol. 7, pp. 116424-116429, August 2019.

[18] W. Michiels and S.-I. Niculescu, Stability, Control, and Computation for Time-Delay Systems: An Eigenvalue-Based Approach, 2nd ed. Philadelphia: SIAM, 2014.

[19] R. Sipahi and N. Olgac, "A unique methodology for the stability robustness of multiple time delay systems," Syst. Control Lett., vol. 55, pp. 819–825, October 2006.

[20] C. Dong et al., "Time-delay stability switching boundary determination for DC microgrid clusters with the distributed control framework," Appl. Energy, vol. 228, pp. 189-204, October 2018.

[21] L. Pekař, R. Matušů, and R. Prokop, "Gridding discretization-based multiple stability switching delay search algorithm: The movement of a human being on a controlled swaying bow," PLoS ONE, vol. 12, art. no. e0178950, June 2017.

[22] L. Pekař and R. Prokop, "Direct stability-switching delays determination procedure with differential averaging," Trans. Inst. Meas. Control, vol. 40, pp. 2217-2226, April 2018.

[23] J. Vanbiervliet, K. Verheyden, W. Michiels, and S. Vandewalle, "A non-smooth optimization approach for the stabilization of time-delay systems," ESAIM Control Optim. Calc. Var., vol. 14, pp. 478-493, July-September 2008.

[24] T. Vyhlídal, W. Michiels, and P. McGahan, "Synthesis of a strongly stable state derivative controller for a time delay system using constrained non-smooth optimization," IMA J. Math. Control, vol. 27, pp. 437-455, December 2010.

[25] W. Michiels and T. Vyhlídal, "An eigenvalue based approach for the stabilization of linear time-delay systems of neutral type," Automatica, vol. 41, pp. 991-998, June 2005.

[26] P. Zítek, V. Kučera, and T. Vyhlídal, "Meromorphic observer-based pole assignment in time delay systems," Kybernetika, vol. 44, pp. 633-648, 2008.

[27] L. Pekař and R. Prokop, "Algebraic optimal control in RMS ring: A case study," Int. J. Math. Comput. Simul., vol. 7, pp. 59-68, January 2013.