

RESEARCH ARTICLE

Gridding discretization-based multiple stability switching delay search algorithm: The movement of a human being on a controlled swaying bow

Libor Pekař^{1*}, Radek Matušů^{2☯}, Roman Prokop^{3☯}

1 Department of Automation and Control Engineering, Faculty of Applied Informatics, Tomas Bata University in Zlín, Zlín, Czech Republic, **2** Regional Research Centre CEBIA-Tech, Faculty of Applied Informatics, Tomas Bata University in Zlín, Zlín, Czech Republic, **3** Department of Mathematics, Faculty of Applied Informatics, Tomas Bata University in Zlín, Zlín, Czech Republic

☯ These authors contributed equally to this work.

* pekar@fai.utb.cz



OPEN ACCESS

Citation: Pekař L, Matušů R, Prokop R (2017) Gridding discretization-based multiple stability switching delay search algorithm: The movement of a human being on a controlled swaying bow. PLoS ONE 12(6): e0178950. <https://doi.org/10.1371/journal.pone.0178950>

Editor: Yilun Shang, Tongji University, CHINA

Received: February 29, 2016

Accepted: May 22, 2017

Published: June 8, 2017

Copyright: © 2017 Pekař et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme project no. LO1303 (MSMT-7778/2014), <https://www.isvav.cz/projectDetail?rowId=LO1303>. This work was supported by the European Regional Development Fund under the project CEBIA-Tech Instrumentation No. CZ.1.05/2.1.00/19.0376, <http://www.risy.cz/cs/vyhledavace/>

Abstract

Delay represents a significant phenomenon in the dynamics of many human-related systems—including biological ones. It has i.a. a decisive impact on system stability, and the study of this influence is often mathematically demanding. This paper presents a computationally simple numerical gridding algorithm for the determination of stability margin delay values in multiple-delay linear systems. The characteristic quasi-polynomial—the roots of which decide about stability—is subjected to iterative discretization by means of pre-warped bilinear transformation. Then, a linear and a quadratic interpolation are applied to obtain the associated characteristic polynomial with integer powers. The roots of the associated characteristic polynomial are closely related to the estimation of roots of the original characteristic quasi-polynomial which agrees with the system's eigenvalues. Since the stability border is crossed by the leading one, the switching root locus is enhanced using the Regula Falsi interpolation method. Our methodology is implemented on—and verified by—a numerical bio-cybernetic example of the stabilization of a human-being's movement on a controlled swaying bow. The advantage of the proposed novel algorithm lies in the possibility of the rapid computation of polynomial zeros by means of standard programs for technical computing; in the low level of mathematical knowledge required; and, in the sufficiently high precision of the roots loci estimation. The relationship to the direct search QuasiPolynomial (mapping) Rootfinder algorithm and computational complexity are discussed as well. This algorithm is also applicable for systems with non-commensurate delays.

Introduction

The stability of Time Delay Systems (TDS) has been a challenging and intensively studied research topic over the past few decades [1–4]. It is, i.a., given by the fact that delay appears

[projekty-eu/detail?id=141580](https://doi.org/10.1371/journal.pone.0178950). The authors are members of the team that received the funding but they are not responsible persons. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abbreviations: ACP, Associated Characteristic Polynomial; CQP, Characteristic Quasipolynomial; DDS, Delay Dependent Stability; grad, Gradient; LMI, Linear matrix inequality; QPmR, QuasiPolynomial mapping Rootfinder; RF, Regula Falsi; RT, Root Tendency; sgn, Signum.

widely in many real-world systems—e.g. networked, mechanical, electrical, social, economical, etc. [5–7]. Moreover, delay can act as a factor that causes performance degradation and system instability; hence, the latencies and the after-effect phenomena should be considered in the models. Biological systems and human-body models are not unaffected; for instance, a robust estimator of neuronal interaction delays was proposed in [8]; the stabilization mechanism of unstable human quietstanding and motion was studied in [9–11]—where delays are, however, only marginally considered. The human nervous system with reflex delay can also be viewed as the cause of traffic jams [12]. Robust compensation of instability caused by delays for a human standing still was proposed by Asai, et al. [13]. Apart from the above-mentioned habitual studies that take fixed delay values into account; from the robustness point-of-view, it was desirable to also perform research on the influence of particular delay values on stability. This task gives rise to the notion of Delay Dependent Stability—(DDS), which means that the system remains stable in a finite set of delay intervals while taking into consideration fixed or variable system parameters.

Two basic families of DDS methods for computing the delay stability margins are paramount in the literature; namely, Time-domain Indirect and Frequency-domain Direct methods. The former group is based on the Lyapunov-Krasovskii or Lyapunov-Razumikhin approaches, and linear matrix inequalities [14–17]; the Jensen inequality approach [18]; semi-definite programming methods [19]; or the descriptor system approach [20,21]. Such methods suffer from significant computational burdens and provide purely theoretical—and rather conservative results [22]. However, they can deal with both constant and time-varying delays and are even applicable to non-linear systems in most cases. To highlight some important results regarding time-varying delay systems, the maximal allowed upper bounds of multiple time-varying delays expressed in terms of several linear matrix inequalities (LMIs) were investigated e.g. in [23,24]. In [25,26], these bounds were derived from feasible LMIs by employing MATLAB[®] LMI Toolbox [27] to reach average consensus asymptotically; moreover, the rate of convergence was estimated. Compared to the presented paper, these methods provide only the upper bound, the given inequalities are obtained analytically and they are more mathematically involved; however, the domain of time-varying delays provides more general results. The latter family of frequency domain methods—usually better implementable and applicable—includes approaches based on various ideas, see e.g. [28] for their listing. Namely, from the historical point of view, methods based on the Rekasius substitution, giving rise to a delay-free Associated Characteristic Polynomial (ACP), constitute the group worthy of the highest research interest within this family [29–31]. An interesting solution can be obtained via the argument principle [32]. Besides this, Schur-Cohn methods [3,33]—or those eliminating exponential (delay) terms from the Characteristic QuasiPolynomial (CQP) due to its symmetry [28,34]—have also attracted great attention from the control scientific community. The crucial step of these procedures rests in the determination of all characteristic roots (system poles), i.e. the characteristic quasipolynomial zeros located exactly on the imaginary axis—which expresses the stability margin. However, only the rightmost subset of poles makes the system switch from/to stability/instability, respectively. Hence, let the corresponding delays be called switching delays. The necessary condition for the existence of roots crossing the imaginary axis is that the so-called root tendency is nonzero [30,34]. The frequency-domain direct methods, which are of a low—or even none conservatism and computationally acceptable in many cases, were successfully applied to investigate the stability of communications [28], power [35], electromechanical [36], logistic [37]; and many other systems with constant time delay. Note, however, that they cannot be directly applied to nonlinear TDS, or those with time-varying delays.

In spite of all the favorable features introduced above, a significant disadvantage of many frequency-domain direct methods is that they are only applicable to systems (or models) with

commensurate delays where the higher order of commensuracy yields the higher complexity of the analytically derived relations [6]. Motivated by the above deficiency—and also by the endeavor to use easily-accessible standard programs for technical computing (e.g. MATLAB®) by engineers and practitioners, this paper focuses on deriving a computationally simple—yet sufficiently rapid and accurate gridding algorithm for searching for delay margins in both commensurate and non-commensurate TDS with an arbitrary number of independent constant delays. Although the method is not derived analytically in terms of exact analytic formulas, but rather, in a numerical way, the techniques used enable one to retain a very low conservatism level and to reach a high degree of precision; i.e., a discrete time reformulation of the CQP by means of the iterative use of the bilinear transformation with pre-warping [38], followed by the utilization of linear interpolation to obtain the integer-powered ACP, means that the zeros can easily be computed. In addition, a linear, Regula Falsi (RF), interpolation concept is then used for the eventual switching delay estimation.

In this contribution, the designed algorithm is applied to the stability delay margin analysis of a control system with a human on a swaying bow [39]. The skater balances on a hemispheric base and remotely steers the power input to the servo that causes the angle deviation from the bow horizontal position; and consequently, the angle between the skater and the bow symmetry axis is the controlled system output. For fixed controller parameters' values, switching delays decide about the marginal balance values that cause stability/instability feedback behavior. The model is specified, i.a., by the fact that the delay-free feedback system is unstable—which indicates that the delay margins have upper and lower bounds. Moreover, the unstable controlled plan includes two independent delays that can generally be viewed as non-commensurate ones. Note that a problem of stabilizing the skater moving on a skateboard affected by reflex delay (with a different model of the dynamics) was recently solved in [40].

The efficiency and asymptotical complexity of the proposed algorithm is benchmarked against the possibility of seeking out switching delays by using the direct QuasiPolynomial mapping Rootfinder (QPmR) algorithm [41–43] that proved to be suitable for finding the quasipolynomial zeros within a selected complex plane rectangular region. A detailed example of the use of the algorithm, which provides the reader with real time requirements and the accuracy obtained in comparison to trial-and-error switching delays search procedures, is introduced for illustration as well.

Thorough the paper, $\mathbb{C}, \mathbb{R}, \mathbb{N}, \mathbb{N}_0$ denote the set of, respectively, complex numbers, real numbers, positive and non-negative integers. The n -dimensional Euclidean space is denoted as \mathbb{R}^n and let $\mathbb{R}_{0+}^n := \{\mathbf{a} = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n | a_i \geq 0, i = 1, 2, \dots, n\}$. For $s \in \mathbb{C}$, $\text{Re}(s)$ and $\text{Im}(s)$ denote, respectively, the real part and imaginary part of s . The empty set is denoted as \emptyset . The notation $d(s|a)$ means a polynomial in variable s , the coefficients of which are determined by the parameter a under some particular operation.

Methods

The general TDS model and its spectrum

Let,

$$G(s) = n(s, \boldsymbol{\tau})/d(s, \boldsymbol{\tau}) \tag{1}$$

stand for the Laplace transfer function in $s \in \mathbb{C}$ of a single-input single-output linear time-invariant TDS with multiple delays as a frequency-domain model where $n(s, \boldsymbol{\tau}), d(s, \boldsymbol{\tau})$ are quasipolynomials of the general form: $x(s, \boldsymbol{\tau}) = s^n + \sum_{i=0}^n \sum_{j=1}^{h_i} x_{ij} s^i \exp\left(-s \sum_{k=1}^L \lambda_{ij,k} \tau_k\right)$, in which $\boldsymbol{\tau} = (\tau_1, \tau_2, \dots, \tau_L) \in \mathbb{R}_{0+}^L, L > 1$, represent independent delays: $\lambda_{ij,k} \in \mathbb{N}_0$ and $x_{ij} \in \mathbb{R}$.

Let $x_a(s, \tau) = 1 + \sum_{j=1}^{h_n} x_{nj} \exp\left(-s \sum_{k=1}^L \lambda_{nj,k} \tau_k\right)$ be the associated exponential polynomial related to $x(s, \tau)$. If, $d_a(s, \tau) \in \mathbb{R}$, then the system defined in Eq (1) is designated as retarded; otherwise, the system is of a neutral type.

In the following section, we assume that there are no common roots of the numerator and denominator in Eq (1). Under this assumption, the system poles (characteristic roots) s_i agree with the zeros of $d(s, \tau)$, i.e. $s_i := \{s: d(s, \tau) = 0\}$, and thus: $CQP \equiv d(s, \tau)$. Let us denote the spectrum of the system as $\Sigma := \{s_i\}$, the set of all zeros of $d_a(s, \tau)$ as Σ_a and introduce some basic spectral properties that are utilizable in this study.

Property 1 [4,6,44]: The system defined in Eq (1) has the following properties:

1. If nonzero $d_{ij}, \lambda_{ij,k}$ exist (where d_{ij} stands for the real-valued coefficient in $d(s)$ for s^i with j th delay) for some positive τ_k and some i, j, k , then the number of poles is infinite.
2. For a retarded system, for any $\beta \in \mathbb{R}$ with $|\beta| < \infty$, only a finite number of poles are located in the half-plane $\text{Re}(s) > \beta$; while infinitely many others are located in $\text{Re}(s) < \beta$.
3. For a neutral system, a vertical chain of poles (s_i) exists at $\gamma := \max \text{Re} \Sigma_a$ such that $\lim_{i \rightarrow \infty} \text{Re}(s_i) = \gamma, \lim_{i \rightarrow \infty} \text{Im}(s_i) = \infty$ for $|s_i| < |s_{i+1}|, |s_i| < |s_{i+1}|$.
4. Let $\bar{\gamma} := \sup_{\Delta\tau \rightarrow 0} \gamma$, then the number of poles with $\text{Re} s_i > \bar{\gamma}$ is always finite, and they are isolated.
5. Isolated poles behave continuously and smoothly with respect to τ on \mathbb{C} .

1. If one defines the spectral abscissa as the function: $\alpha(\tau) := \tau \mapsto \sup \text{Re} \Sigma$, for which the following holds true (*Property 2*): Function $\alpha(\tau)$ may be nonsmooth and hence, not differentiable; e.g. in points with more than one real pole or conjugated pairs with the same maximum real part [45].
2. It is non-Lipschitz, for instance, at points where the maximum real part has a multiplicity greater than one [46].

Rephrasing Property 2 –especially the second paragraph, abrupt changes may occur in the value of $\alpha(\tau)$; however, such cases are rather rare—in at most, a finite number of delay vector points.

TDS stability

Proposition 1 [3,4]: The system as in Eq (1) with a fixed τ is exponentially stable if $\alpha(\tau) < 0$ for retarded systems; and if $\alpha(\tau) < -\epsilon, \epsilon > 0$ for neutral systems.

Note that, if $\gamma < 0$ (including infinity); then a TDS is so-called formally stable and Proposition 1 can collapse into the condition $\alpha(\tau) < 0$. The value of γ is, moreover, sensitive to even infinitesimal delay deviations (see item 4) of Property 1). This feature gives rise to the notion of strong stability—which means that a system is preserved from formal instability under small delay changes [4] (i.e. $\bar{\gamma} < 0$). The necessary and sufficient strong stability condition can be expressed as:

$$\sum_{j=1}^{h_i} |d_{nj}| < 1 \tag{2}$$

To avoid dealing with a potentially infinite chain of vertical system poles reaching or crossing the imaginary axis, retarded and/or strongly stable neutral systems satisfying Eq (2) are the only ones to be further considered in this study.

Let us now define some other useful notions regarding TDS stability with respect to delay values. A TDS is said to be DDS, if it is exponentially stable for some delay vectors values: $\tau_i \in \mathbb{R}_{0+}^L \setminus \infty$, constituting open sets (regions) of stabilizing delays. The root tendency (RT) is defined as:

$$RT = \text{sgn Re}\{\text{grad } s(\tau)\} \tag{3}$$

Crossing delays $\tau_{c,i}$, and the corresponding crossing frequencies, $\omega_{c,i}$, satisfy:

$$d(j\omega, \tau) \Big|_{\substack{\omega = \omega_{c,i} \\ \tau = \tau_{c,i}}} = 0 \tag{4}$$

For switching delays $\bar{\tau}_i$, and the corresponding switching frequencies $\bar{\omega}_i$, the condition in Eq (4) holds—and moreover, the corresponding pole ($\bar{s}_i = j\bar{\omega}_i$) on the imaginary axis is the leading (furthest right) one; i.e. $\alpha(\bar{\tau}_i) = \text{Re } \bar{s}_i = 0$, and the imaginary axis is crossed with a non-zero velocity [17,30,47], i.e.:

$$RT \Big|_{\substack{\tau = \bar{\tau}_i \\ s = \bar{s}_i}} \neq 0 \tag{5}$$

Let us denote the set of all switching delays and frequencies as: \bar{T} and $\bar{\Omega}$, respectively. It is evident from Proposition 1, that stability may only be violated at switching delays, while poles cross the imaginary axis at switching frequencies. This is the key idea of the direct frequency-domain methods cited above. Apart from the cases listed in Property 2, such an imaginary axis crossing is smooth.

Model of a human being on a controlled swaying bow

In this study, we are concerned with a DDS stability analysis of a control system with a model of a (human) skater on a remotely-controlled swaying bow, see Fig 1. The skater controls the power input, $P(t)$ to the servo—giving rise to the angle deviation, $u(t)$ from the horizontal position, and consequently, the angle between the skater and the bow symmetry axis, $y(t)$ emerges.

If friction is neglected, the following transfer function: ($u(t) \rightarrow y(t)$) can be written:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b \exp(-(\tau_1 + \tau_2)s)}{s^2(s^2 + a \exp(-\tau_2 s))} \tag{6}$$

for some particular bow parameters, where delay τ_1 expresses the skater’s reaction time, and τ_2 means the latency. Nominal controlled system parameters and delay values may be read for instance as: $a = -1$, $b = 0.2$, $\tau_1 = 0.3$, $\tau_2 = 0.1$, as given in [39]. It can be computed that, for such parameters, the controlled TDS is unstable since it has the spectral abscissa value of $\alpha = 0.9534$.

Let us now assume that the habitual simple negative control feedback loop is applied to a plant given in Eq (6), to restore stability. Let the control loop be equipped with a finite-dimensional linear controller governed by the transfer function:

$$C(s) = \frac{\sum_{i=0}^3 q_i s^i}{s^3 + \sum_{i=0}^2 p_i s^i} \tag{7}$$

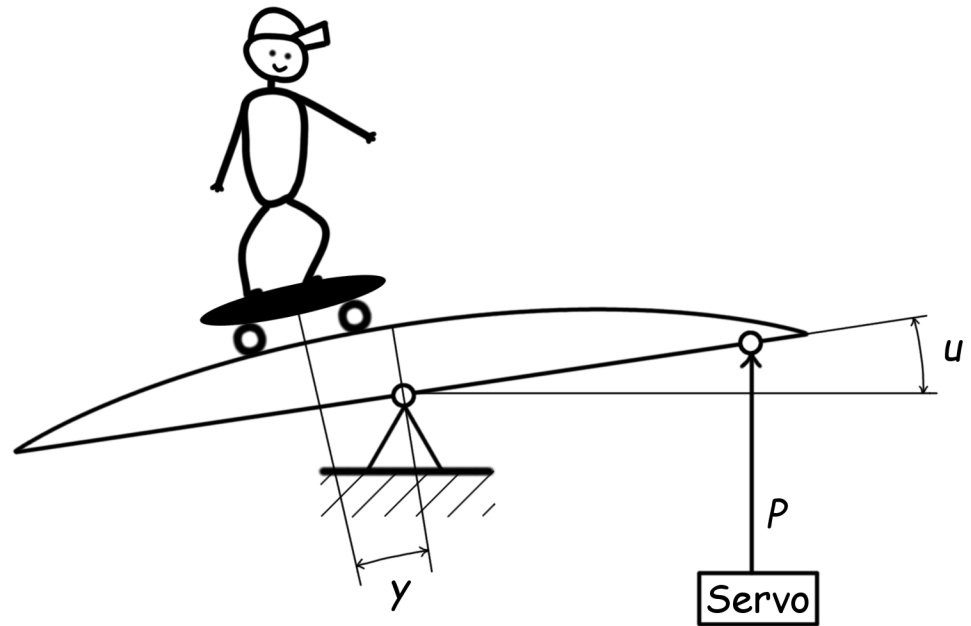


Fig 1. A skater on a remotely controlled swaying bow. The skater can send remote signals to the servomechanism (the rectangle labeled as Servo) that exerts power P in order to cause a horizontal-angle input deviation of u . The horizontal asymmetry yields the deviation of angle y from the bow symmetry axis—which is then taken as the system output.

<https://doi.org/10.1371/journal.pone.0178950.g001>

which gives the following retarded CQP:

$$d(s, (\tau_1, \tau_2)) = s^2(s^2 + a \exp(-\tau_2 s)) \left(s^3 + \sum_{i=0}^2 p_i s^i \right) + b \exp(-(\tau_1 + \tau_2) s) \left(\sum_{i=0}^3 q_i s^i \right) \quad (8)$$

An optimal spectral abscissa minimizing procedure for the above introduced parameters, see [48], yields the result:

$$p_2 = 469418.2, p_1 = 640264.2, p_0 = 10560107, q_0 = 5617613, q_1 = 26247749, q_2 = 106523133, q_3 = 8222650$$

By considering this fixed setting, the nominal spectral abscissa of the control feedback loop is read as follows: $\alpha((0.3, 0.1)) = -1.4454$; while the delay-free case is: $\alpha((0, 0)) = 0.1323$. This means that system stability is delay dependent, and some switching delays might be found for fixed controlled plant and controller parameters.

Discretization tools and techniques

This section intends to provide the reader with the basics of computational discretization techniques and implements useful for the course of the below proposed algorithm.

ACP polynomial approximation. The DDS algorithm introduced further on, in the results section, is designed to be practically implementable by means of most standard programs and utilities for scientific computing—without the necessity of the use of special software. Motivated by this, the first crucial step is to suggest a sufficiently accurate polynomial approximation of the CQP, resulting in the ACP, the leading zeros of which are close to those of the CQP, and can easily be computed.

The concept is based on the successive (iterative) discretization of the CQP, depending on the current leading pole estimation; i.e., a digital-filter-like implementation of the CQP is

obtained. The Tustin (or bilinear) transformation is:

$$s \rightarrow \frac{2}{T} \frac{1 - q}{1 + q} \tag{9}$$

and represents a standard technique for the corresponding designing of the digital filter to a continuous-time model, where T is the sampling period; and q expresses the shifting operator that agrees with z^{-1} in the z -transform [9]. Let us recall that the mapping given in Eq (9) stems from the Taylor series expansion of the mutual relation between the s -plane and the z -plane:

$$z = \exp(sT) \Leftrightarrow s = \frac{1}{T} \ln z \tag{10}$$

The crucial feature of this mapping is that it preserves stability, i.e. the poles in the complex left-half s -plane fall into the interior of the unit circle in the z -plane, and vice versa.

However, it is also known that—for finite-dimensional systems—the transformation given by Eq (9) does not preserve frequencies (i.e. so-called frequency warping occurs). The mutual relations are:

$$\omega = \frac{2}{T} \tan\left(\frac{\omega_d T}{2}\right) \Leftrightarrow \omega_d = \frac{2}{T} \arctan\left(\frac{\omega T}{2}\right) \tag{11}$$

where ω , ω_d stand for the frequency of the continuous-time model and the corresponding digital filter, respectively. Our intention is to make the ACP a digital implementation of the CQP, for which the imaginary part of the leading (dominant) root—after the use of the substitution as per Eq (10)—matches that of the CQP; which, i.a., means that both the corresponding switching imaginary poles might coincide. Hence, the CQP ought to be rescaled so that ω_d —calculated from Eq (11), equals the desired (original, non-scaled) ω . This idea yields the eventual pre-warping filter design mapping as follows:

$$s \rightarrow \frac{\omega}{\tan(\omega T/2)} \frac{1 - q}{1 + q} \tag{12}$$

The task of the choice of T is not unambiguous. With respect to delta models and derivative discretization, the value of T in Eqs (9) or (12) should be sufficiently small; while, on the contrary—the lower T is, the higher ACP degree is obtained. Note that for the z -transform, a recommendation for periodic systems has been given as: $T = [0.2/\omega_0, 0.5/\omega_0]$, where ω_0 expresses the frequency of undamped oscillations [49]. Adopting the basic knowledge about a second order finite-dimensional model: $\omega_0 = |s_0|$, where s_0 means a pole from the conjugate pair, one can write the interval as:

$$T = [0.2|s_0|^{-1}, 0.5|s_0|^{-1}] \tag{13}$$

Note that for infinite-dimensional systems; s_0 can be taken as the dominant pole.

The specific feature of TDSs is that the CQP includes exponential terms that are not subjected to either Eqs (9) or (12). In fact, the bilinear transformation acts only on s -powers in the CQP corresponding to output derivatives in the system model. Exponentials are processed by natural shifting:

$$\exp(-\vartheta s)X(s) \hat{=} x(t - \vartheta) \hat{=} q^{\vartheta/T} x(k) \hat{=} z^{-\vartheta/T} X(z) \tag{14}$$

Switching delay estimation. Once the ACP is found for each grid node given by the delay discretization, i.e. for a particular value of τ , its leading zero is calculated—thus expressing the

estimation of the leading zero of the CQP. As the gridding may be very sparse, the imaginary axis can usually be leaped over. A smoother switching delay estimation—and thus, that of the switching pole, can be made by the linear RF interpolation as follows:

$$\bar{\tau}_k(\tau_{k,j}, \hat{s}_j, \hat{s}_{j+1}) = \tau_{k,j} - \operatorname{Re} \hat{s}_j \frac{\tau_{k,j+1} - \tau_{k,j}}{\operatorname{Re} \hat{s}_{j+1} - \operatorname{Re} \hat{s}_j} \tag{15}$$

where $\bar{\tau}_k$ is the estimation of the k th element of the switching delay vector; $\tau_{k,j}, \tau_{k,j+1}$ stand for the delay elements in the j th and $(j + 1)$ th nodes, respectively; and \hat{s}_j, \hat{s}_{j+1} represents the zeros of the ACP in the corresponding nodes. To rephrase, Eq (15) approximates the zero of the function: $\operatorname{Re} \hat{s}(\tau_k)$. Let us recall that the necessary stability switching condition formulated in Eq (5) must hold for the eventual value of $\bar{\tau}$.

QPmR overview

The QPmR mapping algorithm was designed to compute all quasipolynomial zeros located in a given region in \mathbb{C} . Its leading idea is as follows: Let, $s = \beta + j\omega$, $\beta, \omega \in \mathbb{R}$, and thus split the CQP into two real functions $R(\beta, \omega) := \operatorname{Re}\{d(\beta + j\omega)\}$, $I(\beta, \omega) := \operatorname{Im}\{d(\beta + j\omega)\}$. The region of interest is covered by a rectangular mesh grid: $(\beta_0, \beta_1, \dots, \beta_{k_{\max}}) \times j(\omega_0, \omega_1, \dots, \omega_{l_{\max}})$. Then, the zero of the CQP can be found by the intersection of the curves given by the solutions of: $R(\beta, \omega) = 0, I(\beta, \omega) = 0$, within the gridded region. The accuracy of the zeros is subsequently increased by the Newton method.

The original version (v.1) was designed by Vyhřídál and Zitek [41], and implemented in Matlab based on the use of the Symbolic Math Toolbox. However, the toolbox led to a relatively large computational time. The QPmR was improved in the consecutive work [42], where optimization via the reduction of the scanned regions was implemented. This was achieved by using the argument principle and spectrum distribution analysis [50]. As a result, a significant reduction of computational time was obtained; however, the complexity of the algorithm increased rapidly. The enhanced algorithm was implemented once again in Matlab, as an advanced QPmR (aQPmR). The up-to-date version (v.2) of the algorithm was published in [43]. The authors avoided the use of the Symbolic Math Toolbox—and included the recursive grid density adaptation, which resulted in at least twice as fast computing rates.

Naturally, the QPmR can be utilized for switching delay searches, either as a trial-and-error procedure when directly searching within the given delay range, or as a tool for the evaluation of CQP zeros, instead of the polynomial approximation introduced above. Note however, that the attention of this paper is, i.a., devoted to achieving the goal of avoiding the necessity of using special software tools. Our intention is to employ the QPmR as a benchmark for the algorithm presented herein.

Results and discussion

Prior to the presentation of the proposed DDS algorithm and its implementation on stability margin detection when remote controlling a skater on a swaying bow, some additional methods have to be derived and computational ideas formulated.

Integer-powers polynomial interpolation

The discrete-time formulation of exponentials given by Eq (14) generally results in: $\vartheta/T \notin \mathbb{N}_0$; thus, it is necessary to apply an interpolation:

$$z^{-\vartheta/T} \approx a_0 z^{-[\vartheta/T]} + a_1 z^{-([\vartheta/T]+1)} + a_2 z^{-([\vartheta/T]+2)} + \dots$$

in order to acquire a polynomial with integer powers. A possible way to cope with this task is presented in the following theorem.

Theorem 1: A term $z^{-(n+m)} \in \mathbb{C}$, where $n \in \mathbb{N}_0$, $0 < m < 1$ can be approximated in the neighborhood of arbitrary $z_0 \in \mathbb{C}$ as:

1. Linear interpolation:

$$z^{-(n+m)} \approx (1 - m)z_0^{-m}z^{-n} + mz_0^{-m+1}z^{-(n+1)} \tag{16}$$

2. Quadratic interpolation:

$$z^{-(n+m)} \approx 0.5(2 - m)(1 - m)z_0^{-m}z^{-n} + m(2 - m)z_0^{-m+1}z^{-(n+1)} + 0.5m(m - 1)z_0^{-m+2}z^{-(n+2)} \tag{17}$$

The proof of Theorem 1 is provided in the Appendix.

Corollary 1: If Eq (16) is performed at $z_0 = 1$, which agrees with $s_0 = 0$ in the s -plane, the natural and intuitive result: $z^{-(n+m)} \approx (1 - m)z^{-n} + mz^{-(n+1)}$ is obtained—which expresses the linear distribution in the accordance with the “closeness” of m to 0 or 1.

In our algorithm, the value z_0 is selected as the leading (dominant, right-hand most) pole subjected to Eq (10).

Successive leading pole estimation

Once the ACP in z^{-1} is found by using Eqs (9) or (12), (14), (16) or (17), its dominant root can be computed simply by means of standard engineering computing software, which is a benefit of the algorithm presented below. In order to obtain a more precise estimation, the iterative procedure rather than a single-step calculation is used. It initially adopts the result obtained from the preceding delay discretization grid node (or more precisely, from the nearest previously solved node)—providing the preliminary estimation of the particular dominant pole: $\hat{s}_{0,(1)}$. The choice of ω in Eq (12) and the time period in Eq 13 in its i th iteration is given by: $\hat{s}_{0,(i)}$ and the interpolation introduced in Eqs (16) or (17) is performed at $\hat{z}_{0,(i)}$ —resulting from $\hat{s}_{0,(i)}$ by means of Eq (10), which gives rise to the ACP as $\hat{d}(z^{-1}|\tau, \hat{s}_{0,(i)}) = \sum_{i=0}^n \hat{d}_i(\tau, \hat{s}_{0,(i)})z^{-i}$ where $\tau, \hat{s}_{0,(i)}$ stand for the parameters’ values that determine the polynomial coefficients. In every single iteration step, the updated leading root $\hat{z}_{0,(i+1)}$ of the ACP is subjected to Eq (10) so as to consequently update: $\hat{s}_{0,(i+1)}$. In fact, the root closest to the preceding one is evaluated as the leading one for the next iteration step. The iterative procedure for the particular node of the grid is terminated if $|\hat{s}_{0,(i+1)} - \hat{s}_{0,(i)}| < \varepsilon$ for a selected $\varepsilon > 0$.

The motivation for the iterative calculation of the ACP in the neighborhood of the currently found leading pole can be explained as follows: The pole furthest right of the CQP spectrum is the only decisive one with respect to TDS stability—as clear from Proposition 1. Let this pole: s_0 , be isolated and known for a particular τ_0 exactly. Due to Property 1, for any $\varepsilon > 0$, there exists $\delta > 0$ such that if: $\tau_1 = \tau_0 + \Delta\tau, \|\Delta\tau\| < \delta$, the corresponding root $s_1 := \{s \rightarrow \min|s - s_0|: d(s, \tau_1) = 0\}$ satisfies $|s_1 - s_0| < \varepsilon$. Hence, for a sufficiently small; roots s_0 and s_1 are close to each other. Let us assume the ACP as: $\hat{d}(z^{-1}|\tau_1, s_0)$ and calculate its leading zero $\hat{z}_{1,(1)}$ (i.e. that with the maximum modulus). It is supposed that $\hat{s}_{1,(1)}$ resulting from $\hat{z}_{1,(1)}$ via the exact mapping given in Eq (10) is located in the vicinity of s_0 with $|s_1 - s_0| > |s_1 - \hat{s}_{1,(1)}|$. Now, by the iterative calculation of $\hat{d}_{(i)}(z^{-1}|\tau_1, s_{1,(i)})$, the corresponding s -plane maps of their zeros, $\hat{s}_{1,(i+1)}$, should satisfy $|s_1 - \hat{s}_{1,(i)}| > |s_1 - \hat{s}_{1,(i+1)}|$ and hence, there exists a descending sequence: $\{|\hat{d}_{(i)}(\hat{s}_{1,(i)}, \tau_1)|\}_{i=1}^{i_{\max}}$

which converges for a suitable choice of T . A problem can emerge from Property 2 if $|\hat{s}_{1,(i+1)} - \hat{s}_{1,(i)}| > \delta$ for some $\delta > 0$ and any $\Delta\tau$ due to the spectral abscissa discontinuity; however, such cases are rare and can be omitted in practice—since, for instance, the imaginary axis cannot be crossed by a multiple real pole [30,50]. Moreover, only strongly stable systems are taken into consideration.

Switching delay search algorithm

The proposed gridding switching delay search algorithm poses the main goal of this study and, it is summarized below by using ordered steps and a flowchart. The framework of the algorithm consists of the following steps: initialization, ACP computation and seeking its zeros, as well as switching delays and poles' values enhancement.

Firstly, the $L \times N \in \mathbb{N}^2$ mesh grid of delays within the selected ranges must be created, hence: $\tau_{k,j+1} = \tau_{k,j} + \Delta\tau_{k,j}$, $\tau_{k,0} = 0$, $k = 1 \dots L$, $j = 0 \dots N - 1$. The discretization step $\Delta\tau_{k,j}$ has to be chosen according to the minimum acceptable estimation error. Note, however, that numerical results have shown that this value may be approximately 10^3 -times higher than eventual obtained accuracy. The counter of found switching delays (poles) are to be initialized as well, $i = 1$, and select the admissible convergence error while computing the ACP, $\varepsilon > 0$. The initial leading pole, $\hat{s}_0 = s_0$, can easily be calculated exactly for $\tau = \mathbf{0}$ as the rightmost zero of the polynomial $d(s, \mathbf{0})$.

In each node of the grid (implemented as nested loops over τ_k), the iterative polynomial estimation of the CQP, i.e. the ACP, is computed. Once the final estimations of the dominant poles \hat{s}_0, \hat{s}_1 in two subsequent grid nodes are found, see the preceding section for details, the test whether the imaginary axis has been crossed is executed. If it passes, the RF as in Eq (15) is successively performed to receive $\bar{\tau}_k$ and updated values of \hat{s}_0, \hat{s}_1 are computed by iterations for each k . The eventual delay vector $\hat{\tau}$ has to be subjected to the RT condition test formulated in Eq (5). Estimations of switching delays (\hat{T}), switching poles ($\hat{\Sigma}$) and their imaginary parts ($\hat{\Omega}$), i.e. switching frequencies, are obtained as the algorithm outputs. The algorithm summary is as follows (Algorithm 1):

1. For a neutral TDS—verify the strong stability condition in Eq (2). If it does not pass, terminate the algorithm.
2. For the given $d(s, \tau)$, define the mesh grid: $\tau_{k,j+1} = \tau_{k,j} + \Delta\tau_{k,j}$, $\tau_{k,0} = 0$, $k = 1 \dots L$, $j = 0 \dots N - 1$ for a selected delay range, initialize the counter: $i = 1$ and choose $\varepsilon > 0$. Set: $\hat{T} = \hat{\Sigma} = \hat{\Omega} = \emptyset$.
3. Compute:

$$\hat{s}_{0,\dots,0} = s_{0,\dots,0} := \{\arg \max \operatorname{Re} s : d(s, \mathbf{0}) = 0\} \tag{18}$$

4. For: ($j_1 = 0 \dots N - 1$, for ($j_2 = 0 \dots N - 1 \dots$ (for $j_L = 0 \dots N - 1$ do steps 5–11)).
5. If: $j_1 = j_2 = \dots = j_L = 0$, the inner loop is finished; if not, define:

$$M := \max\{k : j_k \neq 0\} \tag{19}$$

and set:

$$\begin{aligned} \boldsymbol{\tau} &= (\tau_{1,j_1}, \tau_{2,j_2}, \dots, \tau_{L,j_L}) \\ \hat{s}_{old} &= \hat{s}_0 = \hat{s}_{j_1, \dots, j_{M-1}, j_M-1, 0, \dots, 0} \end{aligned} \tag{20}$$

6. Compute: $ACP \equiv \hat{d}(z^{-1} | \boldsymbol{\tau}, \hat{s}_0)$ according to Eqs (9) or (12), (14), (16) or (17). Find its roots (z_i) and subject them to Eq (10), which yield s_i . Define:

$$\hat{s}_1 := \arg \min |s_i - \hat{s}_0| \tag{21}$$

7. While $|\hat{s}_1 - \hat{s}_0| \geq \epsilon$, set $\hat{s}_0 := \hat{s}_1$; and then go to step 6.
8. Set: $\hat{s}_{new} = \hat{s}_{j_1, \dots, j_L} = \hat{s}_{j_1, \dots, j_{M-1}, j_M, 0, \dots, 0} = \hat{s}_1$; if $\text{sgn}(\text{Re } \hat{s}_{new}) = \text{sgn}(\text{Re } \hat{s}_{old})$, the inner loop is finished (see step 4); otherwise, $i = i + 1$ and go to step 9 (RF).
9. Set $\bar{\tau}_M = \bar{\tau}_M(\tau_{M, j_M-1}, \hat{s}_{old}, \hat{s}_{new})$ where function $\bar{\tau}_M$ is defined in Eq (15) and perform the cycle given in step 10.
10. For: $k = M - 1, \dots, 1$ do: If $j_k = 0$, set $\bar{\tau}_k = \tau_{k,0}$ and go back to step 10; or else, set $\hat{s}_0 = \hat{s}_{new}$ and

$$\boldsymbol{\tau}_{old} = (\tau_{1,j_1}, \dots, \tau_{k,j_k-1}, \bar{\tau}_{k+1}, \dots, \bar{\tau}_M, 0, \dots, 0), \boldsymbol{\tau} = (\tau_{1,j_1}, \dots, \tau_{k,j_k}, \bar{\tau}_{k+1}, \dots, \bar{\tau}_M, 0, \dots, 0) \tag{22}$$

and then compute the leading root \hat{s}_1 from $\hat{d}(z^{-1} | \boldsymbol{\tau}_{old}, \hat{s}_0)$ as in steps 6–7. Update values $\hat{s}_{old} = \hat{s}_0 = \hat{s}_1$ and find the leading root \hat{s}_1 from $\hat{d}(z^{-1} | \boldsymbol{\tau}, \hat{s}_0)$ according to the cycle in steps 6 and 7 again, and update the value $\hat{s}_{new} = \hat{s}_1$. Then: $\bar{\tau}_k = \bar{\tau}_k(\tau_{k,j_k-1}, \hat{s}_{old}, \hat{s}_{new})$. When the loop is finished, consolidate $\bar{\tau}_i = (\bar{\tau}_1, \dots, \bar{\tau}_M, 0, \dots, 0)$ and go to step 11.

11. Update $\hat{s}_0 = \hat{s}_{new}$, $\hat{\mathbf{T}} = \hat{\mathbf{T}} \cup \bar{\tau}_i$. Compute the leading zero iteratively: $\bar{s}_1 = \hat{s}_1$ from $\hat{d}(z^{-1} | \bar{\tau}_i, \hat{s}_0)$ in accordance with steps 6–7. Set $\hat{\Sigma} = \hat{\Sigma} \cup \bar{s}_1$, $\hat{\Omega} = \hat{\Omega} \cup |\text{Im } \bar{s}_1|$.
12. Outputs: $\hat{\mathbf{T}}, \hat{\Sigma}, \hat{\Omega}$.

A flowchart of Algorithm 1, beneficial particularly for practitioners, is depicted in Fig 2.

Application of the algorithm to the model of a skater

The particular example presented herein, is aimed at Algorithm 1's accuracy and speed verification, and demonstrates its usability and applicability in bio-cybernetics, by means of the estimation of the stability switching delays for the control loop of a model of a skater on a controlled swaying bow (see Eq (6) and Fig 1) and equipped with the controller given by Eq (7). A set of tests includes the determination of a suitable value of T , the comparison of the use of the linear and quadratic interpolations is according to Eqs (16) and (17) respectively, and the demonstration of the pre-warping asset.

Prior to the presentation of the course of Algorithm 1 itself, let us perform some numerical tests to determine a suitable value of T .

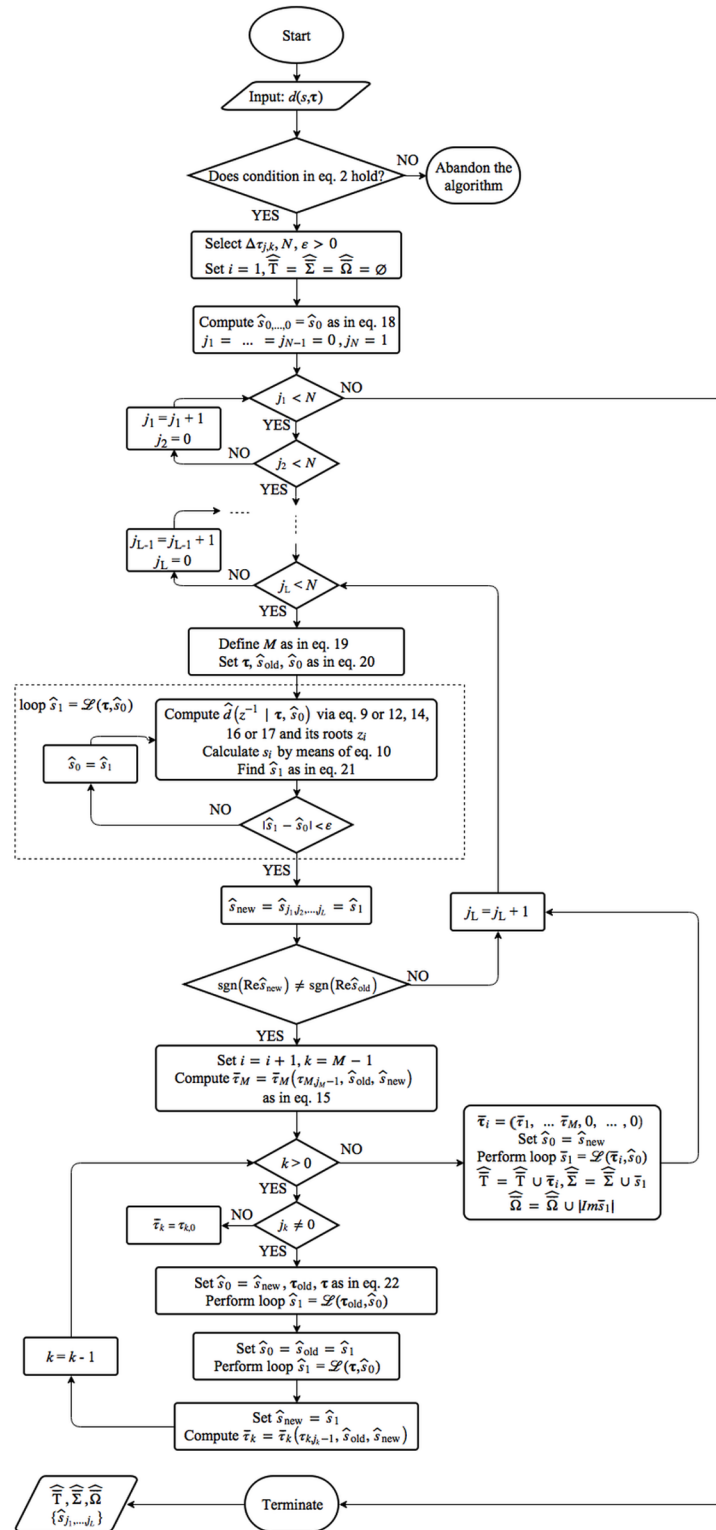


Fig 2. A Flowchart diagram of Algorithm 1. The flowchart diagram displays the designed DDS algorithm. Its step-by-step formulation is presented in Algorithm 1. The ellipse expresses the starting point, and the oval means the termination point. The rhomb, the trapezium and the square stand for the condition, the output and the execution block, respectively.

<https://doi.org/10.1371/journal.pone.0178950.g002>

Table 1. Leading pole estimation accuracy benchmark.

Method	k_T	Error	$iter_{max}$
Eqs (9) and (16)	2	7.025e-2	5
	3	3.109e-2	4
	5	1.112e-2	4
	10	2.79e-3	3
	20	6.99e-4	5
	25	4.52e-4	29
	100	-	Inf
Eqs (12) and (16)	2	2.86e-4	6
	3	1.28e-4	4
	5	4.6e-5	4
	10	1.2e-5	3
	20	3e-6	3
	25	1.0e-5	2
	30	3.6e-5	18
	100	-	Inf
Eqs (12) and (17)	2	2.86e-4	5
	3	1.28e-4	4
	5	4.6e-5	4
	10	1.2e-5	3
	20	3e-6	6
	25	4e-6	5
	30	1.9e-5	37
	100	-	Inf

There are the results of a comparison of the use of Eqs (9), (12), (16) and (17) in Algorithm 1, and applied to the control of a skater on a swaying bow for a fixed delay value and several values of T —see Eq (13). Error means the distance of computed leading poles from the exact analytic one found by the QPmR. The value of $iter_{max}$ stands for the number of iterations within the successive polynomial leading root computation until the precision of ϵ is reached.

<https://doi.org/10.1371/journal.pone.0178950.t001>

Sampling period determination. Let a particular delay vector be chosen and fixed as: $\tau = (0.07, 0.07)$, and let us parameterize T as:

$$T = \frac{1}{k_T |s_0|}, k_T \geq 2 \tag{23}$$

with respect to the upper bound of the condition given by Eq (13). Let us assume moreover, that the leading zero of $d(s, (0.07, 0.06))$ is known exactly—the use of the QPmR with the precision of 10^{-9} gives the value: $s_0 = 0.024805739 \pm 3.905723903i$ —which is used as the initial estimation. An accuracy comparison of leading pole estimations by means of the ACP described in steps 6 and 7 of Algorithm 1 is shown in Table 1. Both the first and second order interpolations as in Eqs (16) and (17) respectively are included—and, moreover, the beneficial impact of pre-warping governed by the Eq (11) is demonstrated. The error expresses the distance of the approximated leading pole from $s_0 = 0.007985448 \pm 3.868912827i$; found by the QPmR with the above-mentioned precision. The admissible iteration error was set to: $\epsilon = 10^{-6}$ (see step 1 of Algorithm 1) and the maximum acceptable number of iteration steps was set to 200. The eventual approximated leading pole loci are given to the reader in S1 Table.

As can be seen from Table 1, the degree of precision is one to three orders better with the use of pre-warping than the bilinear transformation without it. Notice moreover, that it is necessary to perform a lower number of iterations to reach the desired ε for higher k_T —yet, the value is bounded.

Generally, high values of k_T require enormous number of iterations—or even cause divergence—and, the error thus increases slightly. The growth of the error value can be explained as follows. Algorithm 1 uses the bilinear (Tustin) transformation as in Eqs (9) or (12) to get the associated polynomial, and the (back) transformation from the s -plane to the z -plane via the exponential formula as in Eq (10). Geometrically, the conformal mapping (Eq (9)) maps the closed left-hand side of the complex s -plane inside the circle in the z -plane with the radius of $1/T$, i.e., linearly with respect to k_T . The transformation according to Eq (10) provides the map inside the unit circle, i.e. the radius does not depend on the value of k_T . From the operator point of view, the Tustin formula represents the trapezoidal (feedback-feedforward) form of the derivative approximation. It is closely related to the delta transform. Hence, intuitively, the shorter the step T is (i.e. the higher k_T is), the approximation more precise is. However, the latter formula agrees with the transition from the z -transform to the s -transform (or vice-versa). It is well known that the z -transform fails for a very small values of the time period T (for instance, transfer function poles from the transfer function in s collapse to the point 1 in the z -plane). Moreover, the division by a very small number induces numerical problems when computing. The algorithm has to respect a compromise between both the principles. Due the properties of the z -transform and because of the numerical representation of real numbers inside the computer, the value of T must be sufficiently high, i.e. the value of k_T can not be excessively high.

Only a slight improvement is achieved by means of a quadratic interpolation—which is, however, sacrificed by the necessity of more iteration steps. In fact, two leading roots of the ACP do not constitute a complex conjugate pair due to complex polynomial coefficients. It should be noted that the quadratic interpolation yields better mutual convergence of these roots with ascending k_T .

The advantage of the iterative calculation of the ACP can be demonstrated for instance, by the evolution of the error $|\hat{s}_1 - s_0|$ —see steps 6 and 7 of Algorithm 1 for \hat{s}_1 ; and s_0 stands for the exact dominant system pole—with $k_T = 15$ and pre-warping as: $\{23.7, 5.1, 5\} \cdot 10^{-6}$. Note that ω in Eq (12) has been set as: $\text{Im}\hat{s}_1$.

It seems from Table 1 that the suitable value of T lies approximately in the interval $k_T \in [10, 25]$. Now, let us perform the second test as follows: Let the leading zero of $d(s, (0.06, 0))$ be known exactly and perform Algorithm 1 in a degenerate form for $\tau_1 \in [0.06, 0.07]$, $\tau_2 \in [0, 0.2]$ with $\Delta\tau = 0.01$. The numerical results are summarized in Table 2, where $\bar{\tau}$ expresses the switching delay estimation, s_0 is the corresponding leading system pole calculated by the QPmR, the error means the minimum distance of $\bar{\tau}$ from the one found by the QPmR with the delay resolution of $\Delta\tau = 10^{-7}$; and the number of iterations ($iter_{\max}$) expresses the mean value rounded to the nearest integer.

Again, it can be deduced from Table 2 that the use of pre-warping brings about a significant improvement of the switching delay estimation. The quadratic interpolation however, does not come up to expectations; nevertheless, it can reduce the number of iterations for the ACP in a specific range of k_T and results in a better mutual approach to the leading complex pair. In order to also include the time consumption factor—given mainly by the number of iterations, the eventual range of k_T for this example is approximately: $k_T \in [3, 15]$ for Eq (9) and $k_T \in [3, 20]$ for Eq (13)—which goes beyond the suggestion formulated in Eq (13) and stands somewhere between the z -transform and derivative discretization. Hence, after some additional numerical experiments for the switching delay search herein below, we finally decided to set a

Table 2. Switching delay estimation benchmark.

Method	k_T	$\bar{\tau}$	s_0	Error	$iter_{max}$
Eqs (9) and (16)	2	(7.00526, 7.77128)e-2	-5.8741e-3 + 3.840208i	3.269e-3	4
	3	(7.00726, 7.58688)e-2	-2.5627e-3 + 3.846954i	1.427e-3	5
	5	(7.00751, 7.49427)e-2	-9.170e-4 + 3.846954i	5.09e-4	3
	10	(7.00746, 7.45718)e-2	-2.324e-4 + 3.851742i	1.27e-4	3
	20	(7.00744, 7.44771)e-2	-6.99e-5 + 3.852092i	3.3e-5	11
	25	(7.00763, 7.44654)e-2	-4.52e-5 + 3.852128i	2.2e-5	14
	100	-	-	-	Inf
Eqs (12) and (16)	2	(7.00754, 7.44444)e-2	-5.7e-6 + 3.8522095i	2.7e-6	5
	3	(7.00747, 7.44445)e-2	-5.6e-6 + 3.8522096i	2.2e-6	4
	5	(7.00744, 7.44454)e-2	-5.6e-6 + 3.8522096i	2.1e-6	4
	10	(7.00743, 7.44456)e-2	-5.6e-6 + 3.8522096i	2.1e-6	3
	20	(7.00743, 7.44454)e-2	-5.4e-5 + 3.8522100i	2.0e-6	5
	25	(7.00744, 7.44455)e-2	-5.7e-5 + 3.8522093i	2.2e-6	34
	30	-	-	-	Inf
Eqs (12) and (17)	2	(7.00754, 7.44444)e-2	-5.7e-6 + 3.8522095i	2.7e-6	5
	3	(7.00747, 7.44445)e-2	-5.6e-6 + 3.8522096i	2.2e-6	4
	5	(7.00744, 7.44454)e-2	-5.6e-6 + 3.8522096i	2.1e-6	4
	10	(7.00743, 7.44456)e-2	-5.6e-6 + 3.8522096i	2.1e-6	3
	20	(7.00744, 7.44454)e-2	-5.8e-5 + 3.8522090i	2.2e-6	4
	25	(7.00732, 7.44455)e-2	-3.4e-5 + 3.8522136i	1.4e-6	18
	30	-	-	-	Inf

Algorithm 1 is tested when the initial CHP root is known exactly for $\tau \in (0.06, 0)$. Delay values go through intervals $\tau_1 \in [0.06, 0.07]$, $\tau_2 \in [0, 0.2]$ with $\Delta\tau = 0.01$. The value of $\bar{\tau}$ stands for the eventual estimated switching delay vector; and s_0 is the corresponding leading ACP root. Error means the minimum distance of $\bar{\tau}$ from the exact one. The number of iterations is rounded.

<https://doi.org/10.1371/journal.pone.0178950.t002>

uniform $k_T = 8$. Let us recall that T is adapted according to Eq (23) in every single iteration step of the ACP calculation with respect to the leading pole estimation (\hat{s}_0).

Switching delay computation. As the final numerical experiment—the complete course of Algorithm 1 is performed and its results are presented here below. Let the particular region be selected as: $R_1 := \tau_1 \times \tau_2 \in [0, 0.8] \times [0, 0.8]$, with $\Delta\tau = 0.01$, i.e. $N = 80$, and $\varepsilon = 10^{-6}$. The linear interpolation formula, according to Eq (16), is used. Entries of the found \hat{T} values are depicted in Fig 3 and compared with the stable/unstable region calculated by the QPmR of a rough delay resolution of $\Delta\tau = 0.01$. Supporting files S2A, S2B and S2C Table display signs of the rightmost CQP roots from the QPmR, computed leading roots of the ACP along the stability border lines (with the use of pre-warping), and particular values of \hat{T} and $\hat{\Sigma}$, respectively.

During computation, we have observed that the convergence of the ACP estimation may be poor when the leading pair of poles found is sufficiently far from the imaginary axis; or, when $\alpha(\tau)$ is near to its extreme (minimum). In such cases, the problem was solved by an interim small change in k_T . Both the plots in Fig 3 displaying results with/without the use of pre-warping, are almost indistinguishable by sight; hence, to provide the reader with results in more detail, select subsets $R_2 := [0.05, 0.1] \times [0.05, 0.1]$ and $R_3 := [0.069, 0.81] \times [0.063, 0.075]$ of R_1 and display found switching delay pairs back-grounded by stable/unstable grid nodes computed by using the QPmR with the gridding $\Delta\tau = 0.005$ and $\Delta\tau = 0.01$, respectively, see Fig 4A and 4B.

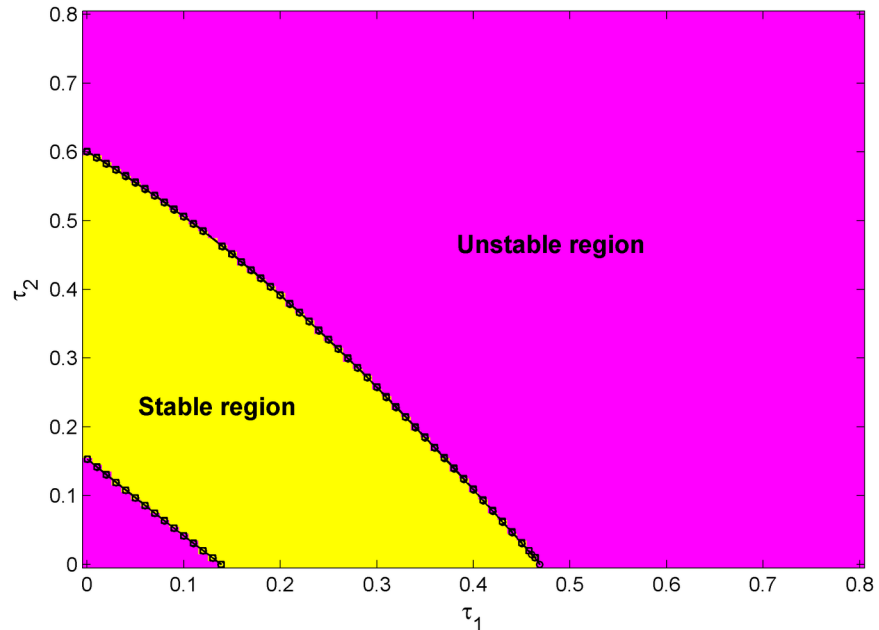


Fig 3. Switching delays found by Algorithm 1. Delay values computed by using pre-warping are indicated by circles and a joint via the solid line; the ones without pre-warping are represented by diamonds and joint by the dashed line. Results are back-grounded stability/instability regions computed by the QPmR with $\Delta\tau = 0.01$.

<https://doi.org/10.1371/journal.pone.0178950.g003>

Table 3 includes the estimated switching delays, the corresponding leading roots found by the QPmR with the precision of 10^{-9} , the error with the same meaning as in Table 2 and the RT vector within the region R_2 .

As can be seen from Table 3, the results correspond to those in Table 2—and, once again, one can state that the use of pre-warping mapping gives two orders higher precision. The overall closeness to the stability bound computed by the QPmR is very good.

Fig 5A, 5B and 5C display feedback output responses for $\tau = (0.3, 0.1)$, $\tau = (0.0700747, 0.74643)$ and $\tau = (0.04, 0.04)$ respectively, when there is the reference step change $\Delta r = 5^\circ = \pi/36$ rad from the zero steady state on the control feedback input at time $t = 10$ s, to verify the result in the time

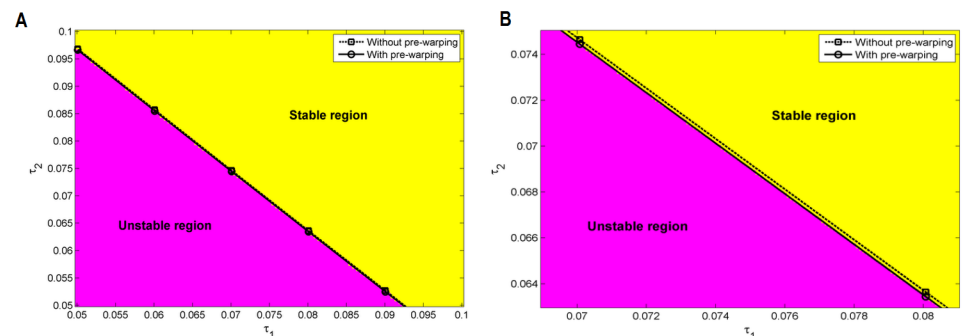


Fig 4. Details of Fig 3 in selected regions. Two regions were selected, a moderately detailed one R_2 (A) and a very detailed one R_3 (B). The stable and unstable areas found by the QPmR for $\Delta\tau = 0.005$ and $\Delta\tau = 0.001$, respectively, are indicated as well.

<https://doi.org/10.1371/journal.pone.0178950.g004>

Table 3. Highlighted results of Algorithm 1—Inside region R_2 .

Mapping	$\bar{\tau}$	s_0	Error	RT
Eq (9)	(5.00643, 9.68377)e-2	-3.578e-4 + 3.847336i	2.03e-4	(-1.980057–3.902248i, -1.776911–3.712380i)
	(6.00732, 8.57107)e-2	-3.592e-4 + 3.849544i	2.02e-4	(-1.979616–3.901380i, -1.784657–3.703965i)
	(7.00747, 7.46430)e-2	-3.604e-4 + 3.851478i	1.99e-4	(-1.979793–3.901079i, -1.793172–3.696423i)
	(8.00698, 6.36342)e-2	-3.609e-4 + 3.853141i	1.98e-4	(-1.980603–3.901336i, -1.802458–3.689744i)
	(9.00595, 5.26837)e-2	-3.611e-4 + 3.854537i	3.47e-4	(-1.982059–3.902144i, -1.812522–3.683919i)
Eq (12)	(5.00665, 9.66366)e-2	-4.9e-6 + 3.848074i	2.4e-6	(-1.977402–3.901955i, -1.774661–3.711958i)
	(6.00741, 8.55114)e-2	-5.5e-6 + 3.850279i	3.0e-6	(-1.976951–3.901107i, -1.782401–3.703575i)
	(7.00743, 7.44455)e-2	-5.6e-6 + 3.852210i	2.1e-6	(-1.977117–3.900824i, -1.790908–3.696065i)
	(8.00682, 6.34384)e-2	-5.1e-4 + 3.853870i	2.7e-6	(-1.977919–3.901097i, -1.800189–3.689416i)
	(9.00568, 5.24897)e-2	-4.3e-4 + 3.855262i	3.3e-6	(-1.979368–3.901922i, -1.810246–3.683621i)

The results of Algorithm 1 within region $R_2 = [0.05, 0.1] \times [0.05, 0.1]$; Root tendency (RT) was defined in Eq (3). The rest of the quantities have already been defined in Table 2.

<https://doi.org/10.1371/journal.pone.0178950.t003>

domain. Apparently, delay values from the region determined as stable (namely, the nominal ones) yield a stable step response (5A); while the ones selected from Table 2 as switching delay result in a steady periodic output of a skater’s position angle deviation (see 5B), and the third chosen delays give unstable response (5C). Let us recall that—amazingly, smaller delay values deteriorate stability in this case. Note that S1 File includes simulation MATLAB® (Simulink) scheme of the control loop.

Computational complexity

Although it is definitely not a difficult or crucial task in the case of the DDS algorithm introduced here, the (asymptotical) computational complexity should be discussed anyway. Parameterized statements are presented first, followed by practically measured data from the above numerical bio-cybernetic example. The basic general facts about Algorithm 1 are going to be presented, as well as the complexity of the QPmR.

The course of Algorithm 1 can be divided into two parts: a cyclic and a non-cyclic one. The non-cyclic part of the program implementation has the time consumption ϑ_{nc} ; consisting of the initialization and the CQP definition. The cyclic one has two subdivisions: with, or without, the RF, $\vartheta_{c,nRF}$, $\vartheta_{c,RF}$ respectively. The former one mainly includes the iterative polynomial approximation and the leading zero calculation, whereas the latter one expresses time for the successive RF interpolation when the imaginary axis is crossed. The following statement can be formulated:

Proposition 2: The computational complexity of Algorithm 1 reads:

$$C(N, L) = \vartheta_{nc} + (N + 1)^L \vartheta_{c,nRF} + c_1(N + 1)^{L-1} \vartheta_{c,RF} = \vartheta_{nc} + (c_2(2L - 1) + N + 1)(N + 1)^{L-1} \vartheta_{c,nRF} \quad (24)$$

$c_1, c_2 \in \mathbb{R}_+ \setminus \infty$

The proof of Proposition 2 is given in the Appendix.

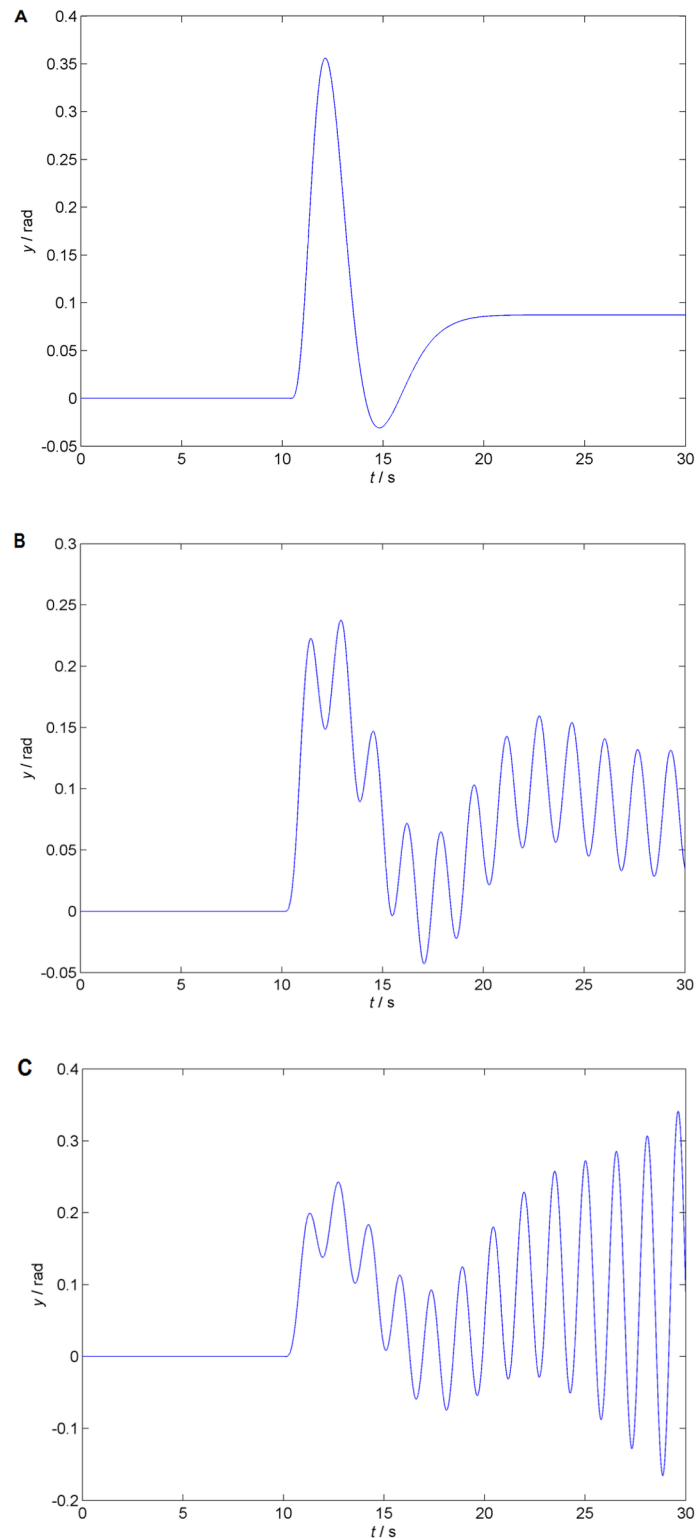


Fig 5. Time domain results verification. The three plots of the system's step reference time responses for three different delay vectors are displayed. Whereas the nominal setting $\tau = (0.3, 0.1)$ (A) gives a stable response asymptotically tracing the reference signal, the second option $\tau = (0.0700747, 0.74643)$ (B) indicates the stability border by steady oscillations, and the last one: $\tau = (0.04, 0.04)$ (C), gives unstable oscillations with a rising amplitude.

<https://doi.org/10.1371/journal.pone.0178950.g005>

As mentioned above, the QPmR can be used in Algorithm 1 instead of steps 6 and 7 to avoid the computation of the ACP; or within the direct procedure, in which for a sufficiently fine grid of discretized delays, the CQP zeros inside a defined region are computed. Regarding the latter case, its complexity can be expressed simply as:

$$C_{QPmR}(N_{QPmR}, L) = \vartheta_{QPmR,nc} + (N_{QPmR} + 1)^L \vartheta_{QPmR,c} \tag{25}$$

where N_{QPmR} stands for the number of delay discretization steps for the QPmR; $\vartheta_{QPmR,nc}$ represents the time that the initialization of the QPmR takes outside the cycle, and $\vartheta_{QPmR,c}$ means the time period of the QPmR course over a grid node itself. Note that the duration $\vartheta_{QPmR,c}$ significantly depends on the selected region for computation of the zeros, its density, and the desired eventual accuracy.

Real computation time efficiency. Finally, let us provide the reader with the concise real computation-time requirements of Algorithm 1, compared to the use of the QPmR for direct leading pole computation inside a defined region. The particular variables and symbols were defined in Eqs (24) and (25). Computational tests were performed by means of a laptop equipped with 32b Windows® 7 Professional, running on an Intel® Core™2 DuoCPU P8700@2.53GHz, 4GB memory, in the Matlab® 7.11.0.584 (2010b) environment.

The whole course of the final computing test introduced above of Algorithm 1 took 2595 s, which gives the following mean durations of particular operations: $\vartheta_{nc} = 0.54s$, $\vartheta_{c,nRF} = 0.36s$, $\vartheta_{c,RF} = 1.64s$ for found $c_1 = 1.75$; hence, it can be calculated that $c_2 = 2.66$.

Now, let us consider the QPmR and two possible search rectangles in the complex plane and their corresponding computed precisions. The first is chosen as a region of the size $[10\text{Re}\{\hat{s}_{0,(i)} - \hat{s}_{0,(i-1)}\} \times 10\text{Im}\{\hat{s}_{0,(i)} - \hat{s}_{0,(i-1)}\}]$ with its center in $\hat{s}_{0,(i)}$, which expresses the currently found leading pole estimation (in the i th iteration). For such a case, we have measured the mean values $\vartheta_{QPmR,nc} = 0.66s$ and $\vartheta_{QPmR,c} = 0.62s$ (for the selected precision of 10^{-8}). The second rectangle has the fixed size of $[1 \times 1]$ centered in $\hat{s}_{0,(i)}$; again, for which we have obtained $\vartheta_{QPmR,nc} = 0.68s$ and $\vartheta_{QPmR,c} = 0.47s$ (for the reduced precision of 10^{-7}). These results imply that the use of the QPmR for seeking the leading pole instead of the ACP roots computing introduced in Algorithm 1 does not yield a shorter computation time. Moreover, the direct trial-and-error use of the QPmR is hardly applicable; for instance, if the latter region is considered with the discretization step of $\Delta\tau = 0.001$ (for a sufficiently smooth result), the overall computation time would approximately be $3 \cdot 10^5$ s; see Eq (25).

Conclusions

The formulation of a novel gridding multiple stability switching delay search algorithm and its verification by a numerical bio-cybernetic example of a human being on a remotely controlled swaying bow were the main objectives of this contribution. The proposed DDS algorithm can be fitted into a group of frequency-domain direct methods, based on an effort to find all characteristic roots (poles) located on the stability border, i.e. on the imaginary axis. Within these methods, it is usually only the upper and lower bounds on the imaginary parts of such poles that can be found by the elimination of delays from the characteristic quasipolynomial [28,33,34]. If the delays are commensurate, a procedure can be relatively fast; however, in the general case of non-commensurate delays, a computationally-lengthy iterative reduction procedure ought to be performed [31]. The algorithm presented herein can deal with non-commensurate delays by more effectively omitting a complex mathematical apparatus. The main advantages of the algorithm reside in the fact that it utilizes basic mathematical operations and standard software tools solely; thus, it is simple to implement by engineers and practitioners.

The effectiveness, simplicity, rapidity and accuracy of the gridding algorithm have been tested by the numerical example. The simple control feedback loop with delays system—applied to a skater on a swaying bow, is controlled by a conventional controller. For such a configuration, the marginal stability values of the delay vector were found by the proposed DDS algorithm; these results can be utilized practically, e.g. in biomechanics. The example presented here, proved that the procedure is computationally more effective than the direct use of the QPmR algorithm [43].

The formulation of the Algorithm 1 introduced above, represents its basic version. There are some natural possibilities regarding how to suggest a modification—for instance, the delays’ discretization can be performed with diverse settings $N_{k_1} \neq N_{k_2}$ for some $k_1 \neq k_2$ rather than a constant N , or with a non-constant $\Delta\tau_{k,j}$ for various k,j . In addition, it is not necessary to take $\tau_{k,0} = 0$; however, in such a case, a sufficiently accurate estimation of the primal pole $\hat{s}_{0,\dots,0}$ has to be made, e.g. by means of the QPmR. These topics might be solved in future research. Its applicability is limited to retarded and strongly stable neutral delay systems—which also yields further natural possible research directions. In addition, the numerical example has shown that the crucial step consists in the determination of a suitable sampling period that influences the accuracy and the convergence rate.

Appendix

Proof of Theorem 1: Assume the linear interpolation first. Then, this can be written:

$$z^{-(n+m)} = z^{-(n+1)} z^{1-m} \tag{26}$$

where the left-hand factor has the integer exponent $1 - m \in (0,1)$. Let the intention be to find: $a_0, a_1 \in \mathbb{C}$ such that $z^{1-m} \approx a_0 + a_1 z$. Now employ the Taylor series expansion at any z_0 :

$$z^{1-m} = [z^{1-m}]_{z=z_0} + [(1 - m)z^{-m}]_{z=z_0} (z - z_0) + c_1 (z - z_0)^2 \tag{27}$$

for some $c_1 \in \mathbb{C} \setminus \infty$ emerging from the remainder. If $|z - z_0|$ is sufficiently small, the last term can be neglected. Then, by substituting Eqs (27) into (26) and applying simple algebra on Eq 26 the eventual formula—as in Eq (16) is obtained.

Now consider a proof of Eq (17) and use an analogous idea. Hence:

$$z^{-(n+m)} = z^{-(n+2)} z^{2-m} \tag{28}$$

where $2 - m \in (1,2)$. The Taylor series expansion of the latter term can be given as:

$$z^{2-m} = [z^{2-m}]_{z=z_0} + [(2 - m)z^{1-m}]_{z=z_0} (z - z_0) + 0.5[(2 - m)(1 - m)z^{-m}]_{z=z_0} (z - z_0)^2 + c_2 (z - z_0)^3 \tag{29}$$

for some $c_2 \in \mathbb{C} \setminus \infty$. Again, the cubic term can be cancelled from Eq (29) for some z sufficiently close to z_0 , and by combining Eqs (29) and (28)—these yield Eq (17), finally.

Proof of Proposition 2: The cyclic part must act in every grid node, the overall number of which is $(N + 1)^L$. Only if the imaginary axis is crossed by the leading pole, is the RT part applied in addition. This happens approximately once or a few times ($c_1 < N$) during the inner-most loop course—and this loop is entered $(N + 1)^{L-1}$ times.

The right-hand side of Eq (24) can be obtained by the supposition that the polynomial approximation is the most time consumptive operation. Whereas, outside the RF, there is only one approximation—the number of $2(L - 1) + 1 = 2L - 1$ such operations inside the RF is performed, i.e. $\vartheta_{c,RF} = \tilde{c}_2(2L - 1)\vartheta_{c,nRF}$ where $\tilde{c}_2 > 0$ assumes other low-time consuming operations within the cycle. Substituting this and $c_2 = c_1 \tilde{c}_2$ into the left-hand side of

Eq (24), the right-hand value is obtained. Note that for a particular N, L one can also write $C(N, L) = \vartheta_{nc} + c_3(2L + N)(N + 1)^{L-1}\vartheta_{c,nRF}$, $c_3 > 0$.

Supporting information

S1 Table. The eventual approximated leading pole loci for Table 1. Leading pole estimations of the ACP received from steps 6 and 7 of Algorithm 1, which gives rise to errors introduced in Table 1.

(XLSX)

S2 Table. Results of the numerical experiment according to Algorithm 1. Signs of the right-most quasipolynomial roots from the QPmR for Fig 3 are given in part A: The data table includes signs of the CQP roots calculated via the QPmR that give rise to the stable and unstable regions in Fig 3. Computed leading roots of the ACP along the stability border line of Fig 3 are provided in part B: The table displays the computed leading roots of the ACP for delay values in the vicinity of the stability border and also their real parts for some other selected delays, as in Fig 3. The pre-warping strategy is used here. Computed switching delays and poles for Fig 3 are introduced in part C: In the table, the reader can find values sets of and for both the strategies (with or without pre-warping) to support data in Fig 3.

(XLSX)

S1 File. A simulation scheme of the skater stabilization. This *.mdl file from MATLAB® enables to test the feedback response for various τ_1 and τ_2 that can be set inside the "Skater model" block.

(MDL)

Author Contributions

Conceptualization: LP.

Data curation: LP.

Formal analysis: LP.

Investigation: LP.

Methodology: LP.

Project administration: LP RM RP.

Resources: LP RM RP.

Software: LP.

Supervision: RP.

Validation: LP RM.

Visualization: LP RM.

Writing – original draft: LP RM RP.

Writing – review & editing: LP.

References

1. Abusaksaka AB and Partington JR. BIBO stability of some classes of delay systems and fractional systems. Syst Control Lett. 2014; 64: 43–46. <https://doi.org/10.1016/j.sysconle.2013.11.009>

2. Bonnet C, Fioravanti AR and Partington JR. Stability of neutral systems with commensurate delays and poles asymptotic to the imaginary axis. *SIAM J Control Optim.* 2011(2); 49: 498–516. <https://doi.org/10.1137/090776275>
3. Gu K, Kharitonov VL, Chen J. *Stability of time-delay systems.* 2nd ed. Boston: Birkhäuser; 2003.
4. Michiels W and Niculescu SI. *Stability and stabilization of time-delay Systems.* 1st ed. Philadelphia: SIAM; 2007.
5. Chiasson J, Loiseau JJ, editors. *Applications of time delay systems.* 1st ed. New York: Springer; 2007.
6. Sipahi R, Vyhlídal T, Niculescu SI, Pepe P, editors. *Time delay systems: Methods, applications and new trends.* 1st ed. New York: Springer; 2012.
7. Zhang W, Branicky MS, Phillips SM. Stability of networked control systems. *IEEE Contr Syst Mag.* 2001; 21(1): 84–99. <https://doi.org/10.1109/37.898794>
8. Wibrál M, Pampu M, Priesemann V, Siebenhünher F, Siewert H, Linder M, et al. Measuring information-transfer delays. *PLoS ONE.* 2013; 8(2): e55809. <https://doi.org/10.1371/journal.pone.0055809> PMID: 23468850
9. Saha DJ, Morasso P. Stabilization strategies for unstable dynamics. *PLoS ONE.* 2012; 7(1): e30301. <https://doi.org/10.1371/journal.pone.0030301> PMID: 22279580
10. Suzuki Y, Nomura T, Casadio M, Morasso P, Intermittent control with ankle, hip, and mixed strategies during quietstanding: A theoretical proposal based on a double inverted pendulum model. *J Theor Biol.* 2012; 310: 55–79. <https://doi.org/10.1016/j.jtbi.2012.06.019> PMID: 22732276
11. Zenzeri J, De Santis D, Morasso P. Strategy switching in the stabilization of unstable dynamics. *PLoS ONE.* 2014; 9(6): e99087. <https://doi.org/10.1371/journal.pone.0099087> PMID: 24921254
12. Orosz G, Wilson RE, Stepan G. Traffic jams: dynamics and control. *Phil Trans R Soc A.* 2010; 368: 4455–4479. <https://doi.org/10.1098/rsta.2010.0205> PMID: 20819817
13. Asai Y, Tasaka Y, Nomura K, Nomura T, Casadio M, Morasso P, et al. A model of postural control in quiet standing: Robust compensation of delay-induced instability using intermittent activation of feedback control. *PLoS ONE.* 2009; 4(7): e6169. <https://doi.org/10.1371/journal.pone.0006169> PMID: 19584944
14. Li X, De Souza CE. Delay-dependent robust stability and stabilization of uncertain linear delay systems: A linear matrix inequality approach. *IEEE T Automat Contr.* 1997; 42(8): 1144–1148. <https://doi.org/10.1109/9.618244>
15. Pepe P, Jiang ZP. A Lyapunov—Krasovskii methodology for ISS and iISS of time-delay systems. *Syst Control Lett.* 2006; 55(12): 1006–1014. <https://doi.org/10.1016/j.sysconle.2006.06.013>
16. Cao J. Improved delay-dependent exponential stability criteria for time-delay system. *J Franklin Inst.* 2013; 350: 790–801.
17. Talebi S, Ataei M, Ekramian M. Guaranteed interval of delay for a class of non-linear control systems. *T I Meas Control* 2015; 38(3): 365–369. <https://doi.org/10.1177/0142331215576989>
18. Zhu XL, Yang GH. Jensen inequality approach to stability analysis of discrete-time systems with time-varying delay. In: *Proceedings of American Control Conference*; 2008 June 11–13; Seattle, WA. p. 1644–1649.
19. Moon YS, Park P, Kwon WH, Lee YS. Delay dependent robust stabilization of uncertain state-delayed systems. *Int J Control.* 2001; 74(14): 1447–1455. <https://doi.org/10.1080/00207170110067116>
20. Fridman E, Shaked U. A descriptor system approach to H1 control of linear time-delay systems. *IEEE T Automat Contr.* 2002; 47(2): 253–270. <https://doi.org/10.1109/9.983353>
21. Xu S, Lam J, Zou Y. Simplified descriptor system approach to delay-dependent stability and performance analyses for time-delay systems. *IEE P-Contr Theor Ap.* 2005; 152(2): 147–151. <https://doi.org/10.1049/ip-cta:20045023>
22. Kwon OM, Park MJ, Park JH, Lee SM, Cha EJ. Improved robust stability criteria for uncertain discrete-time systems with interval time-varying delays via new zero equalities. *IET Control Theory A.* 2012; 6(16): 2567–2575. <https://doi.org/10.1049/iet-cta.2012.0257>
23. Zhang Q, Niu Y, Wang L, Shen L, Zhu H. Average consensus seeking of high-order continuous-time multi-agent systems with multiple time-varying communication delays. *Int J Control Autom Syst.* 2011 9: 1209–1218. <https://doi.org/10.1007/s12555-011-0623-3>
24. Yu H, Xia X, Zhang T. A less conservative method for average consensus with multiple time-varying delays. In: *Proceedings of IEEE Africon*; 2011 September 13–15; Livingstone, Zambia; 2011. p. 423–429.

25. Shang Y. Continuous-time average consensus under dynamically changing topologies and multiple time-varying delays. *Appl Math Comput.* 2014; 244: 1745–1757. <https://doi.org/10.1016/j.amc.2014.07.019>
26. Shang Y. Average consensus in multi-agent systems with uncertain topologies and multiple time-varying delays. *Linear Algebra Appl.* 2014; 459: 411–429. <https://doi.org/10.1016/j.laa.2014.07.019>
27. Gahinet P, Nemirovski A, Laub AJ, Chilali M. *LMI Control Toolbox User's Guide*, The Math Works, Natick, MA, 1995.
28. Sönmez S, Ayasun S, Nwankpa CO. An exact method for computing delay margin for stability of load frequency control systems with constant communication delays. *IEEE T Power Syst.* 2015; 31(1): 370–377. <https://doi.org/10.1109/TPWRS.2015.2403865>
29. Rekasius ZV. A stability test for systems with delays. In: *Proceedings of the Joint Automatic Control Conference; 1980 December; San Francisco, CA; Amer. Inst. of Chemical Engineers; 1980. Paper no: TP9-A.*
30. Olgac N, Sipahi R. A practical method for analyzing the stability of neutral type LTI-time delayed systems. *Automatica.* 2004; 40(5): 847–853. <https://doi.org/10.1016/j.automatica.2003.12.010>
31. Delice II, Sipahi R. Delay-independent stability test for systems with multiple-delays. *IEEE T Automat Contr.* 2012, 57(4): 963–972. <https://doi.org/10.1109/TAC.2011.2168992>
32. Xu Q, Stepan G, Wang Z. Delay-dependent stability analysis by using delay-independent integral evaluation. *Automatica* 2016; 70(C): 153–157. <https://doi.org/10.1016/j.automatica.2016.03.028>
33. Chen J, Gu G, Nett CN. A new method for computing delay margins for stability of linear delay systems. *Syst Control Lett.* 1995; 26(2): 107–117. [https://doi.org/10.1016/0167-6911\(94\)00111-8](https://doi.org/10.1016/0167-6911(94)00111-8)
34. Walton KE, Marshall JE. Direct method for TDS stability analysis. *IEE P-Contr Theor Ap.* 1987; 134(2): 101–107.
35. Ayasun S, Gelen A. Stability analysis of a generator excitation control system with time delays. *Electr Eng.* 2010; 91(6): 347–355. <https://doi.org/10.1007/s00202-009-0142-x>
36. Ji JC. Stability and bifurcation in an electromechanical system with time delays. *Mech Res Commun.* 2003; 30(3): 217–225. [https://doi.org/10.1016/S0093-6413\(03\)00006-5](https://doi.org/10.1016/S0093-6413(03)00006-5)
37. Song Y, Peng Y. Stability and bifurcation analysis on a logistic model with discrete and distributed delays. *Appl Math Comput.* 2006; 181(2): 1745–1757. <https://doi.org/10.1016/j.amc.2006.03.025>
38. Oppenheim A. *Discrete time signal processing*. Upper Saddle River, NJ: Pearson Higher Education; 2010.
39. Zítek P, Kučera V, Vyhlídal T. Meromorphic observer-based pole assignment in time delay systems. *Kybernetika.* 2008; 44(5): 633–648.
40. Varszegi B, Takacs D, Stepan G, Hogan SJ. Stabilizing skateboard speed-wobble with reflex delay, *J R Soc Interface.* 2016; 13(121): 20160345. <https://doi.org/10.1098/rsif.2016.0345> PMID: 27534701
41. Vyhlídal T, Zítek P. Quasipolynomial mapping algorithm rootfinder for analysis of time delay systems. In: *Proceedings of the IFAC Workshop on Time-Delay Systems (TDS 2003); 2003 September 8–10; Rocquencourt, France. Rocquencourt: INRIA; 2003. p. 1–6.*
42. Vyhlídal T, Zítek P. Mapping based algorithm for large-scale computation of quasipolynomial zeros. *IEEE T Automat Contr.* 2009; 54(1): 171–177. <https://doi.org/10.1109/TAC.2008.2008345>
43. Vyhlídal T, Zítek P. QPmR—Quasi-polynomial root-finder: Algorithm update and examples. In: Vyhlídal T, Lafay JF, Sipahi R, editors. *Delay systems: From theory to numerics and applications*. New York: Springer; 2014. p. 299–312. https://doi.org/10.1007/978-3-319-01695-5_22
44. Hale JK, Verduyn Lunel SM. *Introduction to Functional Differential Eqs.* 1st ed. New York: Springer; 1993.
45. Vanbiervliet T, Verheyden K, Michiels W, Vandewalle S. A nonsmooth optimization approach for the stabilization of time-delay systems. *ESAIM Contr Optim Ca.* 2008; 14(3): 478–493. <https://doi.org/10.1051/cocv:2007060>
46. Burke J, Lewis AS, Overton ML. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM J Optim.* 2005; 15(3): 751–779. <https://doi.org/10.1137/030601296>
47. Walton KE, Marshall JE. Direct method for TDS stability analysis. *IEE Proc-D.* 1987 134(2): 101–107.
48. Pekař L, Prokop R. Algebraic optimal control in RMS ring: A case study. *Int J Math Comput Simulat.* 2013; 7(1): 59–68.
49. Balátě J. [Automatic control]. 1st ed. Prague: BEN Publishing; 2004. Czech.
50. Bellmann R, Cooke KL. *Differential-difference Equations*. New York: Academic Press; 1969.