Computer algebra methods for the stability analysis of differential systems with commensurate time-delays

Yacine Bouzidi, Adrien Poteaux, Alban Quadrat

Computer algebra methods for the stability analysis of differential systems with commensurate time-delays

Yacine Bouzidi[1], Adrien Poteaux[2] and Alban Quadrat[3]

***Abstract*— This paper is devoted to the study of the stability of linear differential systems with commensurate delays. Within the frequency-domain approach, it is well-known that the asymptotic stability of such systems is ensured by the condition that all the roots of the corresponding quasipolynomial have negative real parts. A classical approach for checking this condition consists in computing the set of critical zeros of the quasipolynomial, i.e., the roots (and the corresponding delays) of the quasipolynomial that lie on the imaginary axis, and then analyzing the variation of these roots with respect to the variation of the delay. Following this approach, based on solving algebraic systems techniques, we propose a certified and efficient symbolic-numeric algorithm for computing the set of critical roots of a quasipolynomial. Moreover, using recent algorithmic results developed by the computer algebra community, we present an efficient algorithm for the computation of Puiseux series at a critical zero which allows us to finely analyze the stability of the system with respect to the variation of the delay [1]. Explicit examples are given to illustrate our algorithms.**

I. INTRODUCTION

This paper aims at studying the stability of the class of *linear time-invariant (LTI) differential systems with commensurate time-delays* by means of computer algebra methods recently developed by the symbolic computation community.

An example of such a system is defined by the state-space representation $\dot{x}(t) = \sum_{k=0}^{m} A_k\, x(t - k\,\tau)$, where $\tau \in \mathbb{R}_+ := \{\tau \in \mathbb{R} \mid \tau \geq 0\}$, $m \in \mathbb{N}_+ := \{0, 1, \ldots\}$, $A_0, \ldots, A_m$, and $B_1, \ldots, B_m$ are constant matrices with entries in a field $\mathbb{K}$ (e.g., $\mathbb{K} = \mathbb{Q}$, $\mathbb{R}$). The *characteristic function* of this system is defined by a *quasi-polynomial* $f(s, \tau) = \sum_{j=0}^{mn} p_j(s)\, e^{-j\,\tau\, s}$, where the p_j's are polynomials in the complex variable s with coefficients in $\mathbb{K}$.

In this paper, we investigate the *asymptotic stability* of LTI differential commensurate time-delay systems whose dynamics are defined by quasipolynomials. Recall that such a system is said to be asymptotically stable if all the zeros of the quasi-polynomial $f(s, \tau)$ have negative real parts, i.e., $f(s, \tau) \neq 0$ for all s in the closed right half-plane $\overline{\mathbb{C}}_+ := \{s \in \mathbb{C} \mid \Re(s) \geq 0\}$. Our approach for analyzing the asymptotic stability of this class of systems is based on the so-called *critical pairs* of $f(s, \tau)$, that is the pairs $(\omega \in \mathbb{R},\, \tau \in \mathbb{R}_+)$ such that $f(i\,\omega, \tau) = 0$. For instance, see, e.g., [2], [3], [4], [5], [6] and the references therein. If such critical pairs exist, the stability is derived from the asymptotic behavior of the coordinates s of these pairs, called *critical imaginary roots* of $f(s, \tau)$, that is the way these critical imaginary roots behave under small variation of the time-delay τ. There are two independent steps to analyze the stability. First, the critical pairs of $f(s, \tau)$ are computed and then for each critical pair (s_0, τ_0), the asymptotic behavior of the root s_0 with respect to small variation of τ_0 has to be studied. See [2], [3], [4], [5], [6] and the references therein.

There exist several methods for computing the critical pairs. For the study of the asymptotic behavior of critical imaginary roots, several methods were developed for the case of simple imaginary roots (see [2], [3], [4], [5], [6] and the references therein). For *multiple imaginary roots*, a recent method based on *Puiseux series* was developed in [2], [1].

The contributions of this paper are twofold. We first present a new approach for the computation of the critical pairs. Applying some transformations, this approach roughly reduces the computation of the critical pairs of a quasipolynomial to the computation of the real solutions of a zero-dimensional polynomial system in two variables, i.e., a system admitting a finite number of complex solutions. This can be achieved using the so-called *Rational Univariate Representation* (RUR) [7] which is, roughly speaking, a one-to-one mapping between the solutions of the polynomial system and the roots of a univariate polynomial.

Our second contribution concerns the study of the asymptotic behavior of the quasipolynomial $f(s, \tau)$ at a critical pair (s_0, τ_0). More precisely, using recent advances obtained in the computer algebra community, we present an efficient algorithm that allows us to compute the different terms of Puiseux series. Hence, we can study the variation $\Delta\, s$ in terms of $\Delta\, \tau$ in a neighborhood of a critical pair (s_0, τ_0).

The paper is organized as follows. In Section II, we present two efficient and certified algorithms which compute the critical pairs. An algorithm which computes Puiseux series expansions is given in Section III.

II. COMPUTATION OF THE CRITICAL PAIRS

In this section, we focus on the certified computation of the critical pairs of the quasipolynomial $f(s, \tau)$, that is its zeros of the form $(i\,\omega,\, \tau)$, where $(\omega,\, \tau) \in \mathbb{R} \times \mathbb{R}_+$. This problem is equivalent to computing the set of real zeros of $f(i\,\omega, \tau)$. Due to the presence of transcendental terms, this quasipolynomial usually admits an infinite number of zeros. By means of particular transformations, this problem can be reduced to

[1]Yacine Bouzidi is with Inria Lille - Nord Europe, Non-A project, 40 Avenue Halley, Bat A - Park Plaza, 59650 Vileneuve d'Ascq, France. yacine.bouzidi@inria.fr

[2]Adrien Poteaux is with University of Lille I, CRIStAL - UMR CNRS 9189, CFHP, Batiment M3, 59655 Villeneuve d'Ascq, France. adrien.poteaux@univ-lille1.fr

[3]Alban Quadrat is with Inria Lille - Nord Europe, Non-A project, 40 Avenue Halley, Bat A - Park Plaza, 59650 Vileneuve d'Ascq, France. alban.quadrat@inria.fr

Supported by the ANR-13-BS03-0005.

the computation of the real solutions of a *zero-dimensional polynomial system*, i.e., a polynomial system admitting only a finite number of complex solutions. We present two kinds of transformations. The first one, the Rekasius transformation, appears in [8] and was used in a series of works [5], [6]. The second one is a Möbius transformation.

Rekasius transformation. This transformation consists in replacing the terms $e^{-\tau\, i\,\omega}$ in the quasipolynomial $f(i\,\omega, \tau)$ by the rational fraction $\frac{1-T\,i\,\omega}{1+T\,i\,\omega}$, where $T \in \mathbb{R}$. Cleaning the denominators, we obtain a polynomial of the form $\mathcal{R}(\omega, T) + i\,\mathcal{I}(\omega, T)$. Computing the critical pairs then amounts to studying the real zeros of the following polynomial system:

$$\mathcal{R}(\omega, T) = 0, \quad \mathcal{I}(\omega, T) = 0. \tag{1}$$

Given a solution $(\omega,\, T) \in \mathbb{R}^2$ of (1), the critical delays can be obtained by $\tau_k = \frac{2}{\omega}\,(\arctan(\omega\, T) + k\,\pi)$ for $k \in \mathbb{Z}$.

Möbius transformation. An alternative to Rekasius transformation consists in replacing the terms $e^{-\tau\, i\,\omega}$ by a new variable z. When the delay τ varies, $e^{-\tau\, i\,\omega}$ covers the complex torus $\mathbb{T} := \{z \in \mathbb{C} \mid |z| = 1\}$. The problem of studying the zeros of $f(i\,\omega, \tau)$ then amounts to studying the zeros of the polynomial $f(i\,\omega, z)$, where $\omega \in \mathbb{R}$ and $z \in \mathbb{T}$. For more details, see, e.g., [2] and the references therein.

A first approach is to consider $z = u + i\,v$, where $u,\, v \in \mathbb{R}$, so that $f(i\,\omega,\, u + i\,v) = \mathcal{R}(\omega,\, u,\, v) + i\,\mathcal{I}(\omega,\, u,\, v)$, where $\mathcal{R}$ and $\mathcal{I}$ are two polynomials with real coefficients. The problem then leads to the problem of computing the real solutions $(\omega,\, u,\, v)$ of the following polynomial system:

$$\mathcal{R}(\omega,\, u,\, v) = 0, \quad \mathcal{I}(\omega,\, u,\, v) = 0, \quad u^2 + v^2 - 1 = 0.$$

Generically, the above system is zero-dimensional, i.e., it only admits a finite number of complex solutions. Standard computer algebra methods such that the *Rational Univariate Representation* (RUR) [7] can then be used to obtain certified numerical approximations of the real solutions.

In order to reduce the number of indeterminates in the above equivalent polynomial system, we propose the Möbius transform $\frac{x-i}{x+i}$ which maps $\mathbb{R} \cup \{\infty\}$ to $\mathbb{T}$. Similarly as above, substituting $e^{-\tau\, i\,\omega}$ by $\frac{x-i}{x+i}$ into $f(i\,\omega, \tau)$ and cleaning the denominators, we obtain a polynomial of the form $\mathcal{R}(\omega, x) + i\,\mathcal{I}(\omega, x)$, where $\mathcal{R}$ and $\mathcal{I}$ are two polynomials with real coefficients. The critical pairs can then be computed by solving the following polynomial system

$$\mathcal{R}(\omega, x) = 0, \quad \mathcal{I}(\omega, x) = 0, \tag{2}$$

and then the critical delays are obtained as follows:

$$\tau_k = \frac{1}{\omega}\left(\arctan\left(\frac{2\,x}{x^2-1}\right) + k\,\pi\right), \quad k \in \mathbb{Z}. \tag{3}$$

Remark 1: We can observe that $z = 1$ is sent to ∞ in the above Möbius transformation. This case, which corresponds to the delays of the form $2\,k\,\pi$, where $k \in \mathbb{Z}$, can be independently studied. The corresponding quasipolynomial is then a polynomial $f(i\,\omega, 2\,k\,\pi)$ in ω whose real roots can be isolated using, e.g., classical *bisection algorithms* [9], [10].

Up to a sign, the Rekasius transformation is equivalent to the Möbius transformation by setting $x = -T\,\omega$.

For the above transformations, the computation of the critical pairs can be reduced to solving (1) or (2).

In what follows, we only consider the case of the Möbius transformation (the approach being similar for Rekasius transformation). Let us now consider the polynomial system (2) resulting from the above transformation. Without loss of generality, we can assume that (2) admits only a finite number of complex solutions, which means that the polynomials $\mathcal{R}$ and $\mathcal{I}$ do not have a non-trivial common factor. Our objective is to study the solutions of (2) by means of modern computer algebra methods.

A first important remark is that the ω-coordinates of the solutions of $f(i\,\omega, \tau) = 0$ (called the *imaginary roots* of f), are actually the roots of a *resultant* [1] of $\mathcal{R}$ and $\mathcal{I}$ with respect to the variable x [12].

Theorem 1: The roots of $\mathrm{Res}_x(\mathcal{R}, \mathcal{I})$ are the projection onto the ω-axis of the common solutions of $\mathcal{R}$ and $\mathcal{I}$ and the common roots of a_n and b_m.

If we want to check whether or not (2) admits imaginary roots, we can first compute $\mathrm{Res}_x(\mathcal{R}, \mathcal{I})$ and then check whether or not $\mathrm{Res}_x(\mathcal{R}, \mathcal{I})$ admits real roots. A method to achieve this is, for instance, to use the *Descartes rule of sign* [10]. If $f(i\,\omega, \tau)$ does not admit solutions with real ω-coordinates, then it means that the stability of the system is independent from the delay τ. In that case, a method for checking the stability consists in substituting, for instance, $\tau = 0$ in $f(s, \tau)$ and then applying the standard *Routh criterion* to the univariate polynomial $f(s, 0)$.

From the end-user point of view, if $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$, then a convenient way to express the solutions $V(\langle \mathcal{R},\, \mathcal{I}\rangle) := \big\{(\omega,\, x) \in \mathbb{K}^2 \mid \mathcal{R}(\omega, x) = \mathcal{I}(\omega, x) = 0\big\}$ of (2) is to use a RUR [7], that is a one-to-one mapping

$$\begin{array}{rcl} V(\langle \mathcal{R},\, \mathcal{I}\rangle) & \longrightarrow & V(\langle f_h\rangle) \\ (\omega,\, x) & \longmapsto & \xi, \\ \left(\dfrac{g_{h,\omega}(\xi)}{g_h(\xi)},\, \dfrac{g_{h,x}(\xi)}{g_h(\xi)}\right) & \longleftarrow\!\shortmid & \xi, \end{array} \tag{4}$$

between $V(\langle \mathcal{R},\, \mathcal{I}\rangle)$ and $V(\langle f_h\rangle) := \{t \in \mathbb{K} \mid f_h(t) = 0\}$ for a certain univariate polynomial f_h. In order to achieve the one-to-one condition, the representation (4) is computed with respect to a *separating linear form* $h := a_1\,\omega + a_2\,x \in \mathbb{Q}[\omega, x]$, for certain $a_1,\, a_2 \in \mathbb{Q}$, that takes different values when evaluated at the different points of $V(\langle \mathcal{R},\, \mathcal{I}\rangle)$. Using (4), the solutions of $V(\langle \mathcal{R},\, \mathcal{I}\rangle)$ are then defined by

$$f_h(t) = 0, \quad \omega = \frac{g_{h,\omega}(t)}{g_h(t)}, \quad x = \frac{g_{h,x}(t)}{g_h(t)}, \tag{5}$$

where $f_h,\, g_h,\, g_{h,\omega},\, g_{h,x} \in \mathbb{Q}[t]$ and f_h and g_h are coprime polynomials, i.e., $\gcd(f_h, g_h) = 1$.

Computing a RUR requires solving the two problems:

[1]Given an integral domain $\mathbb{A}$ and two polynomials $f = \sum_{i=0}^{n} a_i\, x^i$ and $g = \sum_{i=0}^{m} b_i\, x^i$ in $\mathbb{A}[x]$, the resultant of f and g w.r.t x, denoted by $\mathrm{Res}_x(f, g)$, is the determinant of the so-called *Sylvester matrix*. The latter belongs to $\mathbb{A}$.

- Find a separating linear form $h := a_1\,\omega + a_2\,x$.
- Given a linear form $h \in \mathbb{Q}[\omega, x]$, compute a RUR-candidate, that is the polynomials f_h, g_h, $g_{h,\omega}$ and $g_{h,x}$.

Computation of the RUR-candidate. Given a polynomial h in $\mathbb{Q}[\omega, x]$ (not necessary separating), the RUR-candidate with respect to h can be computed using the algorithm given in [7]. This algorithm requires the knowledge of a basis of the finite $\mathbb{Q}$-vector space $\mathcal{A} := \mathbb{Q}[\omega, x]/\langle \mathcal{R}, \mathcal{I} \rangle$ and a reduction algorithm which computes *normal forms* modulo the ideal $\langle \mathcal{R}, \mathcal{I} \rangle$. In order to explicitly characterize the polynomials appearing in the RUR-candidate, we first define the $\mathbb{Q}$-endomorphism defined by the multiplication by $h \in \mathbb{Q}[\omega, x]$ in $\mathcal{A}$, i.e., we consider $m_h : \mathcal{A} \longrightarrow \mathcal{A}$ defined by $m_h(\overline{p}) = \overline{h\,p}$, where $\overline{p}$ denotes the residue class of $p \in \mathbb{Q}[\omega, x]$ in $\mathcal{A}$ (i.e., modulo $\langle \mathcal{R}, \mathcal{I} \rangle$). A representative of $\overline{p}$ is the *normal form* of p with respect to the *Gröbner basis* of $\langle \mathcal{R}, \mathcal{I} \rangle$. For more details, see [11].

Given a basis $\{e_1, \ldots, e_n\}$ of $\mathcal{A}$ (which can be deduced from a Gröbner basis for any *monomial ordering* [11]), we can compute the $(n \times n)$-matrix M_h associated to m_h. The polynomial f_h of the RUR-candidate is defined as the characteristic polynomial of the matrix M_h. Moreover, if $\overline{f_h} := \frac{f_h}{\gcd\left(f_h, \frac{df_h}{dt}\right)} = \sum_{i=0}^{d} v_i\, t^{d-i} \in \mathbb{Q}[t]$ is the *square-free part* of f_h, and if we note $H_j := \sum_{i=0}^{j} v_i\, t^{j-i} \in \mathbb{Q}[t]$ for $j = 0, \ldots, d-1$, then, we can define:

$$\left\{ \begin{array}{l} g_h := \displaystyle\sum_{i=0}^{d-1} \mathrm{Trace}(M_h^i)\, H_{d-i-1}, \\ g_{h,\omega} := \displaystyle\sum_{i=0}^{d-1} \mathrm{Trace}(M_\omega\, M_h^i)\, H_{d-i-1}, \\ g_{h,x} := \displaystyle\sum_{i=0}^{d-1} \mathrm{Trace}(M_x\, M_h^i)\, H_{d-i-1}. \end{array} \right.$$

Finding of a separating linear form. For the computation of a separating polynomial of $V(\langle \mathcal{R}, \mathcal{I} \rangle)$, a critical remark is that the number of non separating elements is bounded by $m := n\,(n-1)/2$, where n denotes the cardinal of $V(\langle \mathcal{R}, \mathcal{I} \rangle)$, that is the number of lines passing by two distinct points of $V(\langle \mathcal{R}, \mathcal{I} \rangle)$. Thus, a separating form can always be found among the set $\{h := \omega + a\,x \mid a = 0, \ldots, m\}$. On the other hand, the *Bezout theorem* [11], [12], states that for polynomials $\mathcal{R}$ and $\mathcal{I}$ of total degree d, the cardinal of $V(\langle \mathcal{R}, \mathcal{I} \rangle)$ is bounded by d^2. Hence, a strategy for computing a separating element for $V(\langle \mathcal{R}, \mathcal{I} \rangle)$ is to loop over $m := d^2\,(d^2-1)/2 + 1$ different integers a, for each a, compute the number of distinct roots of the polynomial f_h (see above), i.e., the degree of its squarefree part $\overline{f_h}$, and finally select an a for which this number is maximal. This ensures that the degree of $\overline{f_h}$ is equal to the cardinal of $V(\langle \mathcal{R}, \mathcal{I} \rangle)$, and thus that the roots of f_h are in bijection with the points of $V(\langle \mathcal{R}, \mathcal{I} \rangle)$. This strategy is actually time-consuming since it requires the computation of m charateristic polynomials as well as their squarfree parts. In practice, noticing that an arbitrary chosen linear form is separating with high probability, one prefers a strategy that consists in choosing randomly a linear form and testing that the latter is separating a posteriori of the computation of a RUR-candidate. This test is based on the fact that a linear form h is separating for $V(\langle \mathcal{R}, \mathcal{I} \rangle)$ if and only if the polynomials $u_\omega := g_h\,\omega - g_{h,\omega}$ and $u_x := g_h\,x - g_{h,x}$ vanish on all the points of $V(\langle \mathcal{R}, \mathcal{I} \rangle)$ (i.e., belong to the radical of $\langle \mathcal{R}, \mathcal{I} \rangle$). In computational terms, this condition can be translated into $(\mathrm{Trace}(M_{u_\omega\, e_i}))_{i=1,..,n} = (0, \ldots, 0)$ and $(\mathrm{Trace}(M_{u_x\, e_i}))_{i=1,..,n} = (0, \ldots, 0)$ [7].

Efficient RUR algorithm. Actually, the above RUR algorithm is designed for general zero-dimensional systems and is thus not optimized for the specific case of two polynomials in two variables. The computation of a Gröbner basis of $\{\mathcal{R}, \mathcal{C}\}$, required for the computation of the RUR-candidate as well as for the separating form, is often time-consuming.

Alternatively, below, we propose a practically efficient method that computes a decomposition into univariate representations of the solutions of (2) and which avoids the computation of a Gröbner basis. This method consists in shearing the system, i.e., applying a change of variables and then using *resultants* and *subresultants polynomials* [12] to compute parametrizations of the solutions. Providing that the change of variables transforms the system into a generic position, the computed parametrization then encodes the solutions of the system in a one-to-one correspondence with the roots of a univariate polynomial.

Let us start with the following result which shows that there exist $h := \omega + a\,x$ such that, up to a factor in $\mathbb{Q}$, the polynomial f_h is equal to the resultant of two polynomials resulting from $\mathcal{R}$ and $\mathcal{I}$ after a change of variables.

Theorem 2 ([13]): Let $\mathcal{R}(\omega, x), \mathcal{I}(\omega, x) \in \mathbb{Q}[\omega, x]$. Define $\mathcal{R}'(t, x) := \mathcal{R}(t - a\,x, x)$ and $\mathcal{I}'(t, x) := \mathcal{I}(t - a\,x, x)$, where $a \in \mathbb{Z}$ is such that the leading coefficient of $\mathcal{R}'$ and $\mathcal{I}'$ with respect to x are coprime. Then, the resultant of $\mathcal{R}'$ and $\mathcal{I}'$ with respect to x, denoted $\mathrm{Res}_x(\mathcal{R}', \mathcal{I}')$, is equal to:

$$c \prod_{(\alpha_1,\, \alpha_2) \in V(\langle \mathcal{R}, \mathcal{I} \rangle)} (t - \alpha_1 - a\,\alpha_2)^{\mu(\alpha_1, \alpha_2)}, \quad c \in \mathbb{Q}.$$

Given a linear form $\omega + a\,x$, it is well-known that the latter is separating for $V(\langle \mathcal{R}, \mathcal{I} \rangle)$ if and only if the system $\{\mathcal{R}', \mathcal{I}'\}$ is in *generic position*, i.e., for each root α of $\mathrm{Res}_x(\mathcal{R}', \mathcal{I}')$ (where $\mathcal{R}'$ and $\mathcal{I}'$ are defined as in Theorem 2), the gcd of $\mathcal{R}'(\alpha, x)$ and $\mathcal{I}'(\alpha, x)$, denoted $\mathcal{G}(\alpha, x)$, has exactly one root (i.e., there exists only one intersection point above each $t = \alpha$).

To check the above genericity condition, we first perform a triangular decomposition of $\{\mathcal{R}', \mathcal{I}'\}$. More precisely, we compute a set of triangular systems of the form $\{r_k(t), \mathrm{Sres}_k(t, x)\}_{k=1,\ldots,l}$ where l is a positive integer bounded by the minimum degree in x of $\mathcal{R}'$ and $\mathcal{I}'$, and such that $\prod_{k=1}^{l} r_k(t) = \mathrm{Res}_x(\mathcal{R}', \mathcal{I}')$, and $\mathrm{Sres}_k(\alpha, x)$ of degree k in x is equal to $\mathcal{G}(\alpha, x)$ for any root α of $r_k(t)$. The polynomial $\mathrm{Sres}_k(t, x) = \sum_{i=0}^{k} sr_{k,i}\, x^i$ is known as the *k-th subresultant polynomial* of $\mathcal{R}'$ and $\mathcal{I}'$ with respect to x (see [12]). We refer to [13] for more details.

Once a triangular decomposition is computed, the genericity condition is equivalent to the fact that $\mathrm{Sres}_k(t, x)$ writes modulo r_k as $(a_k(t)\,x - b_k(t))^k$ with $\gcd(a_k, b_k) = 1$. To check the latter condition, we can use the following result of which the proof can be found in [14].

Theorem 3: Let $\mathcal{R}(\omega, x), \mathcal{I}(\omega, x) \in \mathbb{Q}[\omega, x]$. Define the polynomials $\mathcal{R}'(t, x)$, $\mathcal{I}'(t, x)$ as in Theorem 2, and let $\{r_k(t), \mathrm{Sres}_k(t, x)\}_{k=1,\dots,l}$ be the triangular decomposition of $\{\mathcal{R}', \mathcal{I}'\}$. Then, $\omega + a\,x$ is a separating element for $V(\langle \mathcal{R}, \mathcal{I}\rangle)$ if and only if we have

$$k\,(k-i)\,sr_{k,i}\,sr_{k,k} - (i+1)\,sr_{k,k-1}\,sr_{k,i+1} = 0 \bmod r_k,$$

for all $k \in \{1, \dots, l\}$ and for all $i \in \{0, \dots, k-1\}$.

Provided that the conditions of Theorem 3 are satisfied (i.e., the system is in generic position), each $\mathrm{Sres}_k(t, x)$ is then of the form $(a_k(t)\,x - b_k(t))^k$ with $\gcd(a_k, b_k) = 1$. Moreover, we can obtain the explicit expression of a_k and b_k by computing the k-th derivative polynomial of $\mathrm{Sres}_k(t, x)$, which yields that $a_k = k!\,sr_{k,k}$ and $b_k = (k-1)!\,sr_{k,k-1}$. Finally, the solutions of the system $\{\mathcal{R}, \mathcal{I}\}$ can be parametrized using the following set of parametrizations:

$$\begin{cases} r_k(t) = 0, \\ x = \dfrac{-sr_{k,k-1}(t)}{k\,sr_{k,k}(t)}, \\ \omega = t - a\,\dfrac{sr_{k,k-1}(t)}{k\,sr_{k,k}(t)}, \end{cases} \qquad k = 1, \dots, l. \tag{6}$$

Numerical approximations. Once a RUR of the solutions of (2) is computed, we can obtain certified numerical approximations of these solutions by first isolating the real roots of f_h by means of intervals using, for instance, the algorithm in [10], and then substituting these intervals into the rational fractions $\frac{g_{h,\omega}(t)}{g_h(t)}$ and $\frac{g_{h,x}(t)}{g_h(t)}$ in order to get isolating intervals for the coordinates ω and x of the solutions. Moreover, substituting these intervals into (3) yields intervals for the delays corresponding to each solution.

Efficient algorithms for computing the RUR (5) and the parametrization (6) are implemented in `Maple` (see the command `Rational Univariate Representation` of the package `Groebner` for the first one, and the external library `RS`[2] for the second). Moreover, the numerical approximations of the solutions, obtained by means of a RUR, can be computed using the `Maple` command `Isolate` of the package `RootFinding` with the option `Method="RS"` and `Output="interval"`.

Example 1: We consider the quasipolynomial $f(s, \tau) = e^{-3\,\tau\,s} - 3\,e^{-2\,\tau\,s} + 3\,e^{-\tau\,s} + s^4 + 2\,s^2$ [2, example 4.6].

To compute the critical pairs, we consider the following system, which results from the substitutions $s = i\,\omega$ and $e^{-\tau\,i\,\omega} = \frac{x-i}{x+i}$ into $f(s, \tau)$

$$\begin{cases} \mathcal{R}(\omega, x) = \left(x^3 - 3\,x\right)\omega^4 + \left(-2\,x^3 + 6\,x\right)\omega^2 + x^3 - 3\,x = 0, \\ \mathcal{I}(\omega, x) = \left(3\,x^2 - 1\right)\omega^4 + \left(-6\,x^2 + 2\right)\omega^2 + 3\,x^2 + 7 = 0, \end{cases}$$

and the $f(i\,\omega, 2\,k\,\pi) = (\omega^2 - 1)^2$ (see Remark 1).

The latter polynomial yields the critical pairs $(i, 2\,k\,\pi)$ and $(-i, 2\,k\,\pi)$ where $k \in \mathbb{Z}$. Note that the critical imaginary roots have multiplicity two.

[2]https://who.rocq.inria.fr/Fabrice.Rouillier/software.php

Computing a RUR of the former system with respect to the separating linear form $h = \omega + x$, we obtain:

$$\begin{cases} f_h = \left(t^4 - 2\,t^2 - 7\right)\left(t^8 - 16\,t^6 + 74\,t^4 - 32\,t^2 + 25\right), \\ g_h = t\left(t^{10} - 15\,t^8 + 66\,t^6 - 34\,t^4 - 143\,t^2 + 29\right), \\ g_{h,\omega} = t^{10} - 9\,t^8 - 26\,t^6 + 234\,t^4 + 53\,t^2 + 35, \\ g_{h,x} = 2\,\left(t^4 - 2\,t^2 - 7\right)\left(t^6 - 10\,t^4 + 17\,t^2 - 10\right). \end{cases}$$

Isolating the roots of f_h, we obtain the two real roots

$$t_1 \in [-\tfrac{268918098581}{137438953472}, -\tfrac{67229524645}{34359738368}], \quad t_1 \approx -1.956636687,$$

$$t_2 \in [\tfrac{67229524645}{34359738368}, \tfrac{268918098581}{137438953472}], \quad t_2 \approx 1.956636687,$$

which, after substitution into the rational fractions of the RUR, yields $x_1 = x_2 = 0$ and thus $\omega_j = t_j$ for $j = 1, 2$.

Note that the two ω-coordinates are roots of the polynomial $t^4 - 2\,t^2 - 7$ which can be solved symbolically. We then get $\omega_j = (-1)^j\sqrt{1 + 2\sqrt{2}}$ for $j = 1, 2$.

Finally, the critical delays $\tau_{j,k}$ can then be obtained by (3). In our case, the solutions (ω_1, x_1) and (ω_2, x_2) yields to $\tau_{j,k} = \frac{k\,\pi}{\omega_j}$, where $k \in \mathbb{Z}$ and $j = 1, 2$.

Alternatively, we can compute (6). Since all these solutions of the system are simple (i.e., their multiplicities are equal to 1), this yields the following single parametrization:

$$\begin{cases} r_1 = \left(t^4 - 2\,t^2 - 7\right)\left(t^8 - 16\,t^6 + 74\,t^4 - 32\,t^2 + 25\right), \\ sr_{1,0} = 65536\,t\left(3\,t^6 - 18\,t^4 + 7\,t^2 + 18\right), \\ sr_{1,1} = -327680\,t^6 + 1179648\,t^4 + 983040\,t^2 + 655360. \end{cases}$$

The two algorithms, based respectively on the computation of (5) and (6), were implemented in `Maple`. Below, we report their running times (on a laptop with an Intel i-7 processor and L8 cache) for randomly generated quasipolynomials of total degree deg and density dens (sparse/dense).

deg	dens	numsol	timing for (5)	timing for (6)
10	0.5	13	1.17	0.45
10	1	15	1.62	0.6
15	0.5	28	26.58	2.8
15	1	28	24.32	3.7
20	0.5	32	166.40	9.5
20	1	37	182.37	14.3
40	0.5	92	/	300

In the table, timings are the running times in CPU seconds and numsol is the average number of critical pairs (ω, x).

III. Computing Puiseux series

This section is organized as follows. We first provide definitions and algorithms for the case of polynomials. This situation is well-studied in the computer algebra literature. See [15], [16], [17] for instance. We then explain how these results can be adapted to the case of quasipolynomials. As shown in [2], [1], these results can be used to effectively decide the stability of a LTI differential system with commensurate delays, which are multiples of τ, when τ changes.

Definition (polynomial case). Let us consider a bivariate polynomial $F \in \mathbb{K}[X, Y]$, where $\mathbb{K}$ is a field of characteristic 0 (e.g., $\mathbb{K} = \mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$). Let $\overline{\mathbb{K}}$ denote the algebraic closure of $\mathbb{K}$. If we consider the roots of F viewed as a univariate polynomial in Y, then we have the *Puiseux theorem*.

Theorem 1: Let $F \in \mathbb{K}[X,Y]$ and d the degree of F in Y. Given $x_0 \in \overline{\mathbb{K}}$, there exist positive integers $e_1,\ldots,e_s$ with $\sum_{i=1}^{s} e_i = d$ and d distinct series

$$S_{ij}(X) = \sum_{k=n_i}^{\infty} \alpha_{ik}\, \zeta_{e_i}^{jk}\, (X-x_0)^{\frac{k}{e_i}}, \quad 1 \le i \le s,$$

where $0 \le j \le e_i - 1$, $n_i \in \mathbb{Z}$, ζ_{e_i} is the e_i-th root of unity, and $\alpha_{in_i} \neq 0$ if $S_{ij} \neq 0$, such that $F(X, S_{ij}(X)) = 0$.

Over most points x_0, the polynomial $F(x_0, Y)$ has d distinct roots. In such a case, the *Implicit Function Theorem* ensures that the series solutions are actually Taylor series. Such series can be quickly computed using, e.g., *quadratic Newton iterations* [18]. When dealing with multiple roots over a point x_0, this may not be the case anymore.

Computing the first term of the Puiseux series. We will compute the Puiseux series using a variant of the well-known *Newton-Puiseux algorithm* [19], [20], namely the *rational Newton-Puiseux algorithm* [15]. Let us explain the main idea of this algorithm by means of an example, beginning with the computation of the first term of each Puiseux series.

Example 2: Let us compute the Puiseux series at $x_0 = 0$ of $F := Y^6 + Y^5 X + 5\,Y^4 X^3 - 2\,Y^4 X + 4\,Y^2 X^2 + X^5 - 3\,X^4$.

Note that we can always be in this situation via a change of the variable $X \leftarrow X + x_0$. From Theorem 1, we know that the first term of any such series $S(X)$ is of the form $\alpha\, X^{\frac{m}{q}}$, $\alpha \in \mathbb{K}$ and $(m,q) \in \mathbb{N}^2$. We then have:

$$\begin{aligned} F(X, \alpha\, X^{\frac{m}{q}} + \cdots) &= \alpha^6\, X^{\frac{6\,m}{q}} + \alpha^5\, X^{\frac{5\,m}{q}+1} + 5\,\alpha^4\, X^{\frac{4\,m}{q}+3} \\ &\quad -2\,\alpha^4\, X^{\frac{4\,m}{q}+1} + 4\,\alpha^2\, X^{\frac{2\,m}{q}+2} + X^5 - 3\,X^4 + \cdots \end{aligned}$$

To get $F(X, Y(X)) = 0$, at least two terms of the above sum must cancel, i.e. (m,q) must be chosen so that two of the above exponents coincide.

This leads us to consider the *support* of a polynomial F.

Definition 1: Given $F(X,Y) = \sum_{i,j} a_{ij} X^j Y^i$, then $\text{Supp}(F) = \{(i,j) \in \mathbb{N}^2 \mid a_{ij} \neq 0\}$ is the *support of* F.

The condition on (m,q) can be translated as follows: two points of $\text{Supp}(F)$ belong to the same line $m\,i + q\,j = l$. In order to increase the X-order of the evaluation, there must be no point under this line.

Example 3: Considering again Example 2, we have

$$\text{Supp}(F) = \{(0,6), (1,5), (3,4), (1,4), (2,2), (5,0), (4,1)\},$$

and we have two such lines $i + 2\,j = 6$, and $i + j = 4$. They define the so-called *Newton polygon* of F.

Definition 2: The *Newton polygon* $\mathcal{N}(F)$ of $F \in \mathbb{K}[X,Y]$ is the lower part of the convex hull of its support.

Example 4: Let us consider again Examples 2 and 3. We consider α corresponding to $i + 2\,j = 6$ and we get $F(T^2, \alpha T) = (\alpha^6 - 2\,\alpha^4 + 4\,\alpha^2)\,T^6 - 3\,T^8 + \alpha^5\,T^7 + (5\,\alpha^4 + 1)\,T^{10} + \ldots$ meaning that α must be a root of $P = Z^6 - 2\,Z^4 + 4\,Z^2$. One can notice that $P \in \mathbb{K}[Z^2]$ and that 0 is a root of P. As we are not interested in such a root (we will get the first non-zero term of the corresponding Puiseux series by considering the other edge of the Newton polygon), we can see that it is more interesting to consider a root of the polynomial $\phi = T^2 - 2\,T + 4$.

By construction, the polynomial P will always belong to $\mathbb{K}[Z^q]$. The polynomial ϕ defined in Example 4 is a *characteristic polynomial.*

Definition 3: If $H = \sum a_{ij} X^j Y^i \mathbb{K}[X,Y]$, then the *characteristic polynomial* ϕ_Δ of $\Delta \in \mathcal{N}(H)$ is defined by $\phi_\Delta(T) = \sum_{(i,j)\in\Delta} a_{ij} T^{\frac{i-i_0}{q}}$, where i_0 is the smallest value such that (i_0, j_0) belongs to Δ for some j_0.

Computing more terms. Up to now, we have described a way to compute the first non zero term of each Puiseux series: we compute the Newton polygon, the characteristic polynomial of each of its edges, and compute the roots of these polynomials. Note first that we assume the use of symbolic computation techniques: we have to decide whether or not coefficients are 0 to be able to certify the correctness of our computations. At the end of this section, we will briefly discuss a symbolic-numeric strategy developed by the second author to overcome possibly costly symbolic computations.

In our context (assuming the generalisation in the next paragraph), if one of the roots of the characteristic polynomial leads to a purely imaginary coefficient for one of the Puiseux series (in the sequel, such a root is denoted by ξ), we then have to compute its next term to decide on the stability of the system. This can actually be done by applying the above strategy to $F\left(X, Y + \xi^{\frac{1}{q}}\, X^{\frac{m}{q}}\right) \in \mathbb{K}\left(\xi^{\frac{1}{q}}\right)\left[X^{\frac{1}{q}}, Y\right]$, or similarly to $F(X^q, Y + \xi^{\frac{1}{q}}\, X^m) \in \mathbb{K}\left(\xi^{\frac{1}{q}}\right)[X,Y]$. We can also avoid taking a q-th root of ξ by considering polynomial $F(\xi^v\, X^q, X^m\,(Y + \xi^u))/X^l \in \mathbb{K}(\xi)[X,Y]$, where $u, v \in \mathbb{Z}$ are such that $u\,q - m\,v = 1$ [15, section 4].

```
Factor($\mathbb{K}$, $\phi$)
Input: $\mathbb{K}$ a field, and $\phi \in \mathbb{K}[T]$.
Output: A set $\{(\phi_i, M_i)\}_i$, where $\phi_i \in \mathbb{K}[T]$ is monic and
irreducible, and $\phi = c \prod_i \phi_i^{M_i}$ for some $c \in \mathbb{K}$.
```

```
RNPuiseux($F$, $\mathbb{K}$)
Input: $\mathbb{K}$ a field, $F \in \mathbb{K}[X,Y]$ separable.
Output: Truncated Puiseux series of $F$.
1. $\mathcal{R} \leftarrow \{\}$
2. For $\Delta$ in $\mathcal{N}(F)$ do
3.    Compute $q$, $m$, $l$, and $\phi_\Delta$ associated to $\Delta$
4.    Compute $(u,v) \in \mathbb{Z}^2$ s.t. $u\,q - m\,v = 1$
5.    For $(\phi, M)$ in Factor($\mathbb{K}$, $\phi$) do
6.       If $M = 1$ then
7.          $\mathcal{R} \leftarrow \mathcal{R} \cup [\xi^v\, X^q, \xi^u\, X^m]$
8.       Else
9.          Let $\xi$ be any root of $\phi$
10.         $G \leftarrow F(\xi^v\, X^q,\, X^m\,(\xi^u + Y))/X^l$
11.         For $(P,Q)$ in RNPuiseux($\mathbb{K}(\xi)$, $G$) do
12.            $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\xi^v\, P^q,\, P^m\,(\xi^u + Q))\}$
13. Return $\mathcal{R}$.
```

The quasipolynomial case. As mentioned in [1], we can approximate the quasipolynomial by truncated power series.

In [16], [17], it is shown that truncation bounds exist to ensure the exactness of the Puiseux series. A first answer is therefore to compute such an approximation and then apply the standard algorithm explained above.

For the computation of the first term another way, described in [1], is to replace any coefficient access in the polynomial case by an evaluation of some derivatives of the quasipolynomial. For instance, if $f(s, \tau)$ is a quasi-polynomial, then $\frac{1}{2!\,1!}\frac{\partial^3 f}{\partial^2 s\,\partial\tau}(\tau_0, s_0)$ corresponds to the coefficient of the term $(s - s_0)^2\,(\tau - \tau_0)$.

For higher order terms, we can apply a similar idea: instead of evaluating the derivatives at a point, we evaluate it at the truncated Puiseux series already computed. Details about this strategy will be developed in a forthcoming work.

Finally, we have to change the condition in line 6 in `RNPuiseux`. The aim of `RNPuiseux` is to *desingularise* the curve $F(X, Y) = 0$, i.e. to stop when the multiplicity of the root ξ is 1. In our context, we might have to stop before such a condition is fulfilled or might have to consider more terms. We stop when the computed coefficient has a non-zero real part.

A symbolic-numeric strategy. In [21], [16], a symbolic-numeric strategy was developed to compute a numerical approximation of the Puiseux series coefficients with *correct* exponents. Roughly speaking, the idea is to first compute the Puiseux series modulo a well-chosen prime number, which gives the *structure* (exponents…) of the series, and then to use this structure to conduct purely numeric operations. Adapting this strategy to the quasi-polynomial case is an interesting challenge which will be studied in a future work.

Some examples. We implemented a `Maple` prototype to compute the Puiseux series of a quasipolynomial as explained above. We illustrate the results with examples and we show the logs of some computations (including the timing).

Example 5: We consider again the quasipolynomial defined in Example 1 and compute its Puiseux series at $(\tau_0,\, s_0) = (2\,\pi,\, i)$. We get the following:

```
Starting ; evaluation point is 2*Pi I
Newton polygon is   [[0, 3], [2, 0]]
xi is   1/4*I
Real part of the coefficient is 1/4*2^(1/2)
It took .88e-1 seconds
```

Hence, we obtain $\Delta s = (.3535 + .3535\,i)\;(\Delta\tau)^{\frac{3}{2}}$.

Example 6: We consider the quasipolynomial $F(\tau, s) = \sum_{k=0}^{4} a_k(s)\,e^{-k\,s\,\tau}$ where $a_4(s) = s^3 + 2\,s^2 + 2\,s + 1$, $a_3(s) = 3\,s^3 + 9\,s^2 + 9\,s + 4$, $a_2(s) = 5/4\,\pi\,s^5 + 11/4\,\pi\,s^4 + 3\,s^3 - \pi\,s^3 + 1/2\,s^2\pi + 13\,s^2 + 15\,s - 9/4\,s\,\pi + 6 - 9/4\,\pi$, $a_1(s) = 5/4\,\pi\,s^5 + 11/2\,\pi\,s^4 + s^3 + 7/2\,\pi\,s^3 + s^2\,\pi + 7\,s^2 + 11\,s + 9/4\,s\,\pi + 4 - 9/2\,\pi$, and $a_0(s) = 1 + 3\,s + 9/2\,\pi\,s^3 + s^2 - 45/8\,\pi^2 + 9/2\,s\,\pi - 15/8\,s^4\,\pi^2 + 1/2\,s^2\,\pi - 75/8\,s^2\,\pi^2 - 9/4\,\pi + 11/4\,\pi\,s^4 + 15/8\,\pi^2\,s^6$ (this polynomial is given in `http://web1.lss.supelec.fr/delsys2015/Site/Program_files/Wed7_LI.pdf`).

Computing the Puiseux series at the point $(5\,\pi, i)$, we get:

```
Newton polygon is  [[0 2] [1 1] [4 0]]
Edge 1, xi is  -((1/2)*I)/Pi
Real part is 0; Going for a recursive call
   Newton polygon is [[0 1] [1 0]]
   xi is (-3/208+(1/104)*I)*(117*Pi+2-3*I)/Pi^2
   Real part is -(27/16)/Pi,  We're done.
Edge 2, xi is ((6/5)*I)/(50*Pi^3-30*Pi^2-3*Pi)
Real part is 0; Going for a recursive call
   Newton polygon is [[0 1] [1 0]]
   xi is (see below)
   real part is non zero. We're done.
 It took 3.596 seconds
```

This leads to the two following Puiseux series:

$$\Delta s = -0.1591\,i\,\Delta\tau - (0.5371 - 0.3644\,i)\,(\Delta\tau)^2\,,$$

$$\Delta s = -0.0987\,i\;(\Delta\tau)^{1/3} - (0.03557 - 0.0028\,i)\;(\Delta\tau)^{2/3}\,.$$

In this last example, the coefficient growth is important. For instance, the root ξ that is not written above is:

$$\frac{3}{25\pi^2}\;\frac{200\,i\pi^2 - 180\,i\pi - 750\,\pi^3 - 21\,i + 600\,\pi^2}{125000\,\pi^6 - 225000\,\pi^5 + 112500\,\pi^4 - 6750\,\pi^2 - 810\,\pi - 27}.$$

This explains why the running time is longer on this example. This advocates for the development of an efficient symbolic-numeric strategy.

References

[1] X.-G. Li, S.-I. Niculescu, A. Çela, H.-H. Wang, T.-Y. Cai, "On computing puiseux series for multiple imaginary characteristic roots of LTI systems with commensurate delays," *IEEE TAC*, vol. 58, no. 5, pp. 1338–1343, 2013.

[2] X.-G. Li, S.-I. Niculescu, A. Çela, *Analytic Curve Frequency-Sweeping Stability Tests for Systems with Commensurate Delays*. Springer, 2015.

[3] K. Gu, V. L. Kharitonov, J. Chen, *Stability of Time-Delay Systems*. Birkhäuser, 2003.

[4] J. E. Marshall, H. Gorecki, A. Korytowski, K. Walton, *Time-Delay. Systems: Stability and Performance Criteria with Applications*. Ellis Horwood, 1992.

[5] S.-I. Niculescu, *Delay Effects on Stability: A Robust Control Approach*, LNCIS 269, Springer, 2001.

[6] N. Olgac, R. Sipahi, "An exact method for the stability analysis of time-delayed linear time-invariant (LTI) systems," *IEEE TAC*, vol. 47, no. 5, pp. 793–796, 2002.

[7] F. Rouillier, "Solving zero-dimensional systems through the rational univariate representation," *AAECC* , vol. 9, no. 5, pp. 433–461, 1999.

[8] Z. V. Rekasius, "A stability test for systems with delays," in *Proc. Joint Automatic Control Conf.*, 1980.

[9] G. E. Collins, A. G. Akritas, "Polynomial real root isolation using descarte's rule of signs," in *Proceedings of the third ACM symposium on Symbolic and algebraic computation*. ACM, 1976, pp. 272–275.

[10] F. Rouillier, P. Zimmermann, "Efficient isolation of polynomial real roots," *J. Comput. Appl. Math.*, vol. 162, no. 1, pp. 33–50, 2003.

[11] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms*, 3rd ed. Springer, 2007.

[12] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, 2nd ed. Springer, 2006.

[13] Y. Bouzidi, S. Lazard, M. Pouget, and F. Rouillier, "Separating linear forms and rational univariate representations of bivariate systems," *J. Symb. Comput.*, vol. 68, pp. 84–119, 2015.

[14] D. N. Daouda, B. Mourrain, and O. Ruatta, "On the computation of the topology of a non-reduced implicit space curve," in *ISSAC'08*. ACM, 2008, pp. 47–54.

[15] D. Duval, "Rational Puiseux expansions," *Compositio Math.*, vol. 70, no. 2, pp. 119–154, 1989.

[16] A. Poteaux, "Calcul de développements de Puiseux et application au calcul de groupe de monodromie d'une courbe algébrique plane," Ph.D. dissertation, University of Limoges, 2008.

[17] A. Poteaux, M. Rybowicz, "Complexity bounds for the rational newton-puiseux algorithm over finite fields," *AAECC*, vol. 22, pp. 187–217, 2011.

[18] H. T. Kung, J. F. Traub, "All algebraic functions can be computed fast," *J. ACM*, vol. 25, no. 2, pp. 245–260, 1978.

[19] R. J. Walker, *Algebraic Curves*. Springer-Verlag, 1950.

[20] E. Brieskorn, H. Knörrer, *Plane Algebraic Curves*. Birkhaüser, 1986.

[21] A. Poteaux, "Computing Monodromy Groups Defined by Plane Algebraic Curves," in *Proceedings of the 2007 International Workshop on Symbolic-Numeric Computation*. ACM, 2007, pp. 36–45.