



A symbolic computation approach to the asymptotic stability analysis of differential systems with commensurate delays

Yacine Bouzidi, Adrien Poteaux, Alban Quadrat

► To cite this version:

Yacine Bouzidi, Adrien Poteaux, Alban Quadrat. A symbolic computation approach to the asymptotic stability analysis of differential systems with commensurate delays. [Research Report] RR-9044, Inria Lille - Nord Europe; Université de Lille 1, Sciences et Technologies; CRISTAL UMR 9189. 2017, pp.20.

HAL Id: hal-01485536

<https://hal.inria.fr/hal-01485536>

Submitted on 9 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A symbolic computation approach to the asymptotic stability analysis of differential systems with commensurate delays

Yacine Bouzidi, Adrien Poteaux, A. Quadrat

**RESEARCH
REPORT**

N° 9044

March 2017

Project-Teams Non-A



A symbolic computation approach to the asymptotic stability analysis of differential systems with commensurate delays

Yacine Bouzidi*, Adrien Poteaux†, A. Quadrat‡

Project-Teams Non-A

Research Report n° 9044 — March 2017 — 17 pages

Abstract: This paper aims at studying the asymptotic stability of retarded type linear differential systems with commensurate delays. Within the frequency-domain approach, it is well-known that the asymptotic stability of such a system 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 [16]. 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 [15]. Explicit examples are given to illustrate our algorithms.

Key-words: Linear differential systems with commensurate delays, quasipolynomials, asymptotic stability, critical pairs, systems of two polynomial equations in two unknowns, Rational Univariate Representation, subresultants, Puiseux series, computer algebra, implementation.

This work was supported by the ANR-13-BS03-0005 (MSDOS).

* Inria Lille - Nord Europe, Non-A project, Parc Scientifique de la Haute Borne, 40 Avenue Halley, Bat. A - Park Plaza, 59650 Villeneuve d'Ascq, France. Email: yacine.bouzidi@inria.fr.

† University of Lille, CRISTAL - UMR CNRS 9189, CFHP, Batiment M3, 59655 Villeneuve d'Ascq, France. Email: adrien.poteaux@univ-lille1.fr.

‡ Inria Lille - Nord Europe, Non-A project, Parc Scientifique de la Haute Borne, 40 Avenue Halley, Bat. A - Park Plaza, 59650 Villeneuve d'Ascq, France. Email: alban.quadrat@inria.fr.

**RESEARCH CENTRE
LILLE – NORD EUROPE**

Parc scientifique de la Haute-Borne
40 avenue Halley - Bât A - Park Plaza
59650 Villeneuve d'Ascq

Une approche par le calcul formel de la stabilité asymptotique des systèmes linéaires différentiels à retards commensurables

Résumé : Ce papier a pour but l'étude de la stabilité asymptotique des systèmes différentiels linéaires à retards commensurables de type retardé. Dans l'approche fréquentielle, il est bien connu que la stabilité asymptotique d'un tel système est assurée par la condition que les racines du quasipolynôme correspondant ont des parties réelles négatives. Une approche classique pour tester cette condition consiste à calculer l'ensemble des zéros critiques du quasipolynôme, c'est-à-dire les racines (et les retards correspondants) du quasipolynôme qui sont sur l'axe imaginaire, et d'analyser la variation de ces racines par rapport à la variation du retard [16]. Suivant cette approche, et en nous basant sur des techniques de résolution de systèmes algébriques, nous proposons un algorithme symbolique-numérique efficace et certifié pour le calcul de l'ensemble des racines critiques d'un quasipolynôme. De plus, en utilisant des résultats algorithmiques récents développés par la communauté du calcul formel, nous présentons un algorithme efficace pour le calcul des séries de Puiseux en une racine critique nous permettant d'analyser finement la stabilité du système par rapport à la variation du retard [15]. Nous donnons des exemples explicites qui illustrent nos algorithmes.

Mots-clés : Systèmes linéaires différentiels à retards commensurables, quasipolynômes, stabilité asymptotique, paires critiques, systèmes de deux équations polynomiales en deux variables, Représentation Rationnelle Univariée, sous-résultants, séries de Puiseux, calcul formel, implantation.

1 Introduction

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

An example of such a system is defined by the state-space representation

$$\dot{x}(t) = \sum_{k=0}^m A_k x(t - k\tau), \quad (1)$$

where $\tau \in \mathbb{R}_+ := \{\tau \in \mathbb{R} \mid \tau \geq 0\}$, $m \in \mathbb{Z}_{\geq 0} := \{0, 1, \dots\}$ and $A_0, \dots, A_m \in \mathbb{K}^{n \times n}$ with \mathbb{K} a field (e.g., $\mathbb{K} = \mathbb{Q}, \mathbb{R}$). The *characteristic function* of (1) is then a *quasi-polynomial* of the form

$$f(s, \tau) = \det \left(sI_n - \sum_{k=0}^m A_k e^{-k\tau s} \right) = \sum_{j=0}^l p_j(s) e^{-j\tau s},$$

where the p_j 's are polynomials in the complex variable s with coefficients in \mathbb{K} .

In this paper, we consider *retarded type* linear time-invariant differential commensurate time-delay systems, namely systems whose characteristic functions are defined by quasipolynomials of the form of

$$f(s, \tau) = \sum_{j=0}^l p_j(s) e^{-j\tau s},$$

where $\deg p_0 \geq \max\{\deg p_1, \dots, \deg p_l\}$ and $\deg p_j$ stands for the degree of the polynomial p_j in s (e.g., (1)). We investigate the *asymptotic stability* of this class of systems by means of the location of the complex solutions s of $f(s, \tau) = 0$. Recall that a retarded type differential commensurate time-delay system is *asymptotically stable* [2, 12] if and only if all the complex solutions of its characteristic function $f(s, \tau) = 0$ have negative real parts, i.e., $f(s, \tau) \neq 0$ for all s in the closed right half-plane

$$\overline{\mathbb{C}}_+ := \{s \in \mathbb{C} \mid \Re(s) \geq 0\}$$

(see, e.g., [2, 12, 16]). While considering the asymptotic stability of a quasipolynomial $f(s, \tau)$, three different problems can be studied: the first one consists in checking the stability for a fixed value of τ , the second one is the study of the so-called *delay independent stability* property which guarantees the stability for all values of τ in $\mathbb{R}_{\geq 0}$, and the last one, *the stability analysis depending on the time-delay* τ , which considers τ as a parameter of the system and aims at computing the values of τ for which the system is asymptotically stable. In this paper, we mainly focus on the last problem. Our approach for analyzing the asymptotic stability of this class of systems is based on the computation of the so-called *critical pairs* of $f(s, \tau)$, that is the pairs $(\omega, \tau) \in \mathbb{R} \times \mathbb{R}_+$ satisfying $f(i\omega, \tau) = 0$. For more details, see [16, 11, 18, 19, 20] and the references therein. If such critical pairs exist, the asymptotic stability can then be derived from the way the first component of these pairs, called *critical imaginary roots* of $f(s, \tau)$, behaves under a small variation $\Delta\tau$ of the time-delay τ with respect to $\overline{\mathbb{C}}_+$. Thus, the asymptotic stability analysis is divided into two distinct problems. First, the critical pairs of $f(s, \tau)$ must be computed. Then, for each critical pair (ω_0, τ_0) , the behavior of the critical imaginary root ω_0 , particularly its real part, must be studied with respect to a small variation of τ_0 . For more details, see [16, 11, 18, 19, 20].

There exist several methods for computing the critical pairs of a general quasipolynomial. On the other hand, the study of the asymptotic behavior of critical imaginary roots has been addressed for the case of simple imaginary roots (see [11, 16, 18, 19, 20] and the references therein). For *multiple imaginary roots*, a recent method based on the so-called *Puiseux series* [29] was developed in [15, 16].

The contributions of the paper are twofold. We first present a new approach for the efficient computation of the critical pairs of a general quasipolynomial. After a *Möbius transformation*, the problem reduces to the computation of the real solutions of a system formed by two polynomial equations in two variables. An efficient method to solve this last problem is to compute the so-called *Rational Univariate Representation* (RUR) [25, 27] of the polynomial system which is a one-to-one mapping between the solutions of the polynomial system and of the roots of a univariate polynomial. The complex/real solutions of the polynomial system can then be obtained in a *certified manner* by numerical isolation of the complex/real roots of the univariate polynomial [13, 26]¹. The RUR of a polynomial system admitting a finite number of complex solutions can be obtained by means of the command `RationalUnivariateRepresentation` of the Maple package `Groebner`. Motivated by applications in *computational geometry* (computation of the topology of algebraic curves), an extremely efficient algorithm for the RUR computation was recently obtained in [4, 5] for systems formed by two polynomial equations in two variables. This algorithm avoids a *Gröbner basis computation* [8].

Moreover, based on recent advances in the direction of Newton-Puiseux series for algebraic curves developed in [21, 22, 23], the second contribution of the paper is to present an efficient algorithm which efficiently computes terms of Newton-Puiseux series of a quasipolynomial $f(s, \tau)$ at a critical pair (ω_0, τ_0) . Thus, we can use these Newton-Puiseux series to study the variation Δs in terms of $\Delta \tau$ in a neighborhood of the critical pair (ω_0, τ_0) , which allows us to conclude about the asymptotic stability of the corresponding retarded type differential time-delay system.

This paper is based on the congress paper [6].

The paper is organized as follows. In Section 2, we present an efficient algorithm which computes, via a Rational Univariate Representation, certified numerical approximations of the critical pairs of a quasipolynomial. Then, an algorithm which computes Newton-Puiseux series on these critical pairs is presented in Section 3.

2 An efficient algorithm for the computation of the critical pairs

In this section, we focus on the computation of the *critical pairs* of a quasipolynomial

$$f(s, \tau) = \sum_{j=0}^l p_j(s) e^{-j\tau s},$$

namely the set $\{(\omega, \tau) \in \mathbb{R} \times \mathbb{R}_+ \mid f(i\omega, \tau) = 0\}$. Due to the presence of transcendental terms $e^{-j\tau s}$ in $f(s, \tau)$, $f(i\omega, \tau) = 0$ usually admits an infinite number of zeros $(i\omega, \tau)$. Using the so-called *Möbius transformation* (or the *Rekasius transformation* [19, 20]) introduced below, we show that this problem is reduced to the computation of the real solutions of a system defined by two polynomial equations in two variables. An efficient computer algebra method [5] and its implementation in the library RS [28] can be used to solve the latter problem, and thus to compute critical pairs of quasipolynomials in a certified manner.

When the delay τ (resp., ω) varies in $\mathbb{R}_{\geq 0}$ (resp., \mathbb{R}), $e^{-\tau i\omega}$ covers the complex torus:

$$\mathbb{T} := \{z \in \mathbb{C} \mid |z| = 1\}.$$

¹In this paper, the term “certified” means that each real solution of a polynomial system in one variable (resp., n variables) is represented by an interval (resp., a cartesian product of n intervals) with rational endpoints which isolates it from the others. Using this representation, we can obtain numerical approximations of the solutions up to an arbitrary precision, test the vanishing of given polynomials on these solutions, etc.

The problem of studying the zeros of $f(i\omega, \tau)$ then amounts to studying the zeros of the bivariate polynomial $f(i\omega, z) = \sum_{j=0}^l p_j(i\omega) z^j$, where $\omega \in \mathbb{R}$ and $z \in \mathbb{T}$.

A first approach consists in considering $z = u + iv$, where $u, v \in \mathbb{R}$, which yields $f(i\omega, u + iv) = \mathcal{R}(\omega, u, v) + i\mathcal{I}(\omega, u, v)$, where \mathcal{R} and \mathcal{I} are two polynomials with real coefficients. Thus, we have to compute the real solutions (ω, u, v) of the following polynomial system:

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

Generically, the above system is zero-dimensional, i.e., it only admits a finite number of complex solutions. Standard computer algebra methods, particularly the so-called *Rational Univariate Representation* (RUR) [25], can be used to obtain certified numerical approximations of the real solutions of (2).

For an efficiency issue, it is important to reduce the number of the indeterminates $(\omega, u$ and $v)$ of (2). To do that, we use the so-called *Möbius transform* defined by:

$$\begin{aligned} \mathcal{M} : \mathbb{R} \cup \{\infty\} &\mapsto \mathbb{T} \\ x &\mapsto z = \frac{x-i}{x+i} = \frac{x^2-1}{x^2+1} - i \frac{2x}{x^2+1}, \\ \infty &\mapsto 1. \end{aligned}$$

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 in ω and x with real coefficients. The critical pairs can then be computed by first solving the following system formed by two polynomial equations in two variables:

$$\mathcal{R}(\omega, x) = 0, \quad \mathcal{I}(\omega, x) = 0. \quad (3)$$

Accordingly, the critical delays can easily be obtained as follows:

$$\tau_k = \frac{1}{\omega} \left(\arctan \left(\frac{2x}{x^2-1} \right) + k\pi \right), \quad k \in \mathbb{Z}. \quad (4)$$

Thus, the computation of the critical pairs is reduced to searching for the real solutions of the polynomial system (3).

Remark 1 Since $z = 1$ is sent to ∞ by the above Möbius transformation, this case, which corresponds to $\tau_k \omega = 2k\pi$, where $k \in \mathbb{Z}$, can be independently studied. The quasipolynomial f then becomes the pure polynomial $f(i\omega, 1) = \sum_{j=0}^l p_j(i\omega)$ in ω whose real roots can be isolated using, e.g., classical *bisection algorithms* [7, 26].

Remark 2 The *Rekasius transformation* is defined by mapping $e^{-\tau i \omega}$ to $\frac{1-iT\omega}{1+iT\omega}$, where $T \in \mathbb{R}$ (see [19, 20] and the references therein). Note that the Rekasius transformation can be seen as a particular Möbius transform.

In what follows, we consider the case of the Möbius transformation (the approach being similar for the Rekasius transformation). Let us now consider the polynomial system (3). Without loss of generality, we can assume that (3) 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 (see Theorem 1 below for the computation of $\gcd(\mathcal{R}, \mathcal{I})$). Our objective is to develop an efficient approach for the certified computation of the real solutions of (3) by means of modern computer algebra methods.

We note that the ω -coordinate of the solutions of $f(i\omega, \tau) = 0$ (called the *imaginary roots* of f) is a root of the *resultant* $\text{Res}_x(\mathcal{R}, \mathcal{I})$ of \mathcal{R} and \mathcal{I} with respect to the variable x . Let us recall the definition of

a (sub)resultant. Let $A = \mathbb{K}[\omega]$, $\mathcal{R} = \sum_{i=0}^p a_i(\omega) x^i \in A[x]$ and $\mathcal{I} = \sum_{j=0}^q b_j(\omega) x^j \in A[x]$, i.e., the a_i 's and b_j 's belong to \mathcal{A} . Let us suppose that $a_p \neq 0$ and $b_q \neq 0$ so that $\deg_x \mathcal{R} = p$ and $\deg_x \mathcal{I} = q$, and $p \geq q$. Let $A[x]_n = \{P \in \mathcal{A}[x] \mid \deg_x P \leq n\}$ be the set of polynomials with degree at most n and $\{x^i\}_{i=0,\dots,n}$ the standard basis of the free \mathcal{A} -module $A[x]_n$ of rank n . We set $A[x]_n = 0$ for negative integer n . For $0 \leq k \leq q$, we can consider the following homomorphism of free \mathcal{A} -modules:

$$\begin{aligned} \varphi_k : A[x]_{q-k-1} \times A[x]_{p-k-1} &\longrightarrow A[x]_{p+q-k-1} \\ (\mathcal{U}, \mathcal{V}) &\longmapsto \mathcal{U} \mathcal{R} + \mathcal{V} \mathcal{I}. \end{aligned}$$

Using the standard basis of $A[x]_{q-k-1}$ (resp., $A[x]_{p-k-1}$, $A[x]_{p+q-k-1}$) and identifying the polynomial $\sum_{i=0}^{q-k-1} u_i x^i \in A[x]_{q-k-1}$ with the row vector $(u_0, \dots, u_{q-k-1}) \in A^{1 \times (p-k)}$, we obtain that

$$\varphi_k(u_0, \dots, u_{q-k-1}, v_0, \dots, v_{p-k-1}) = (u_0, \dots, u_{q-k-1}, v_0, \dots, v_{p-k-1}) S_k,$$

where the matrix S_k is the matrix defined by:

$$\begin{aligned} S_k &= \begin{pmatrix} U_k \\ V_k \end{pmatrix} \in A^{(q-k+p-k) \times (p+q-k)}, \\ U_k &= \begin{pmatrix} a_0 & a_1 & \dots & a_{q-k} & \dots & a_p & 0 & \dots & 0 \\ 0 & a_0 & \dots & a_{q-k-1} & \dots & a_{p-1} & a_p & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_0 & \dots & \dots & \dots & \dots & a_p \end{pmatrix} \in A^{(q-k) \times (p+q-k)}, \\ V_k &= \begin{pmatrix} b_0 & b_1 & \dots & b_{p-k} & \dots & b_q & 0 & \dots & 0 \\ 0 & b_0 & \dots & b_{q-k-1} & \dots & b_{q-1} & b_q & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & b_0 & \dots & \dots & \dots & \dots & b_q \end{pmatrix} \in A^{(p-k) \times (p+q-k)}. \end{aligned}$$

To be coherent with the degree of polynomials, we attach index $i-1$ to the i^{th} column of S_k so that the index of the columns goes from 0 to $p+q-k-1$.

Definition 1 For $0 \leq j \leq p+q-k-1$ and $0 \leq k \leq q$, let $\text{sr}_{k,j}$ be the determinant of the submatrix of S_k formed by the last $p+q-2k-1$ columns, the column of index j and all the $p+q-2k$ rows. The polynomial $\text{Sres}_k(\mathcal{R}, \mathcal{I}) = \text{sr}_{k,k} x^k + \dots + \text{sr}_{k,0}$ is the k^{th} subresultant of \mathcal{R} and \mathcal{I} , and its leading term $\text{sr}_{k,k}$ is the k^{th} principal subresultant of \mathcal{R} and \mathcal{I} . The matrix $S_0 \in A^{(p+q) \times (p+q)}$ is the Sylvester matrix associated with \mathcal{R} and \mathcal{I} and $\text{Res}_x(\mathcal{R}, \mathcal{I}) = \det S_0$ is the resultant of \mathcal{R} and \mathcal{I} .

Remark 3 For $k < j \leq p+q-2k-1$, we note that $\text{sr}_{k,j} = 0$ since $\text{sr}_{k,j}$ is the determinant of a matrix having twice the same column. Moreover, we can check that we have:

$$\text{sr}_{q,q} = b_q^{p-q}, \quad \forall q < p, \quad \text{Sres}_q(\mathcal{R}, \mathcal{I}) = b_q^{p-q-1} \mathcal{I}.$$

Since $A = \mathbb{K}[\omega]$ is an integral domain, we can consider its field of fractions $Q(A) = \mathbb{K}(\omega)$ and the Euclidean domain $B = Q(A)[x]$. Since $\mathcal{R}, \mathcal{I} \in B$, we can define the greatest common factor $\gcd(\mathcal{R}, \mathcal{I})$, which is defined up to a non-zero element of $Q(A)$, so that we can suppose that $\gcd(\mathcal{R}, \mathcal{I}) \in A$.

Theorem 1 ([1]) The first $\text{Sres}_k(\mathcal{R}, \mathcal{I})$ such that $\text{sr}_{k,k} \neq 0$ is equal to $\gcd(\mathcal{R}, \mathcal{I})$.

Theorem 2 ([1]) The roots of $\text{Res}_x(\mathcal{R}, \mathcal{I})$ are the projection onto the ω -axis of the common solutions of \mathcal{R} and \mathcal{I} and the common roots of a_p and b_q .

If we want to check whether or not (3) admits imaginary roots, we can first compute $\text{Res}_x(\mathcal{R}, \mathcal{J})$ and then check whether or not $\text{Res}_x(\mathcal{R}, \mathcal{J})$ admits real roots. A method to do that is, for instance, to use the *Descartes rule of sign* [26]. If $f(i\omega, \tau)$ does not admit solutions with real ω -coordinates, then it means that the stability of the system is independent of τ . We can then set, for instance, $\tau = 0$ in $f(s, \tau)$ and then apply the *Routh-Hurwitz criterion* to the univariate polynomial $\sum_{j=0}^l p_j(s)$.

Example 1 Let us consider $f(s, \tau) = s + 2 + e^{-\tau s}$. Applying the Möbius transformation to

$$f(i\omega, \tau) = i\omega + 2 + e^{-i\tau\omega},$$

we obtain $\mathcal{R} = -\omega + 3x$ and $\mathcal{J} = \omega x + 1$. Then, we get $\text{Res}_x(\mathcal{R}, \mathcal{J}) = \omega^2 + 3$ whose solutions are complex. Considering $\tau = 0$, $f(s, 0) = s + 3$ is stable which proves that f is stable independently of the delay.

If $\mathbb{K} = \mathbb{Q}, \mathbb{R}$ or \mathbb{C} , then a convenient way to express the solutions

$$V(\langle \mathcal{R}, \mathcal{J} \rangle) := \{(\omega, x) \in \mathbb{K}^2 \mid \mathcal{R}(\omega, x) = \mathcal{J}(\omega, x) = 0\}$$

of the polynomial system (3) is to use the so-called *Rational Univariate Representation* (RUR) [25, 27], that is the following one-to-one mapping

$$\begin{aligned} V(\langle \mathcal{R}, \mathcal{J} \rangle) &\longrightarrow V(\langle f_a \rangle) \\ (\omega, x) &\longmapsto \xi, \\ \left(\frac{g_{a,\omega}(\xi)}{g_a(\xi)}, \frac{g_{a,x}(\xi)}{g_a(\xi)} \right) &\longleftarrow \xi, \end{aligned} \tag{5}$$

between the solutions $V(\langle \mathcal{R}, \mathcal{J} \rangle)$ of (3) and the roots $V(\langle f_a \rangle) := \{t \in \mathbb{K} \mid f_a(t) = 0\}$ of a certain univariate polynomial f_a . In order to achieve the one-to-one condition, the representation (5) is computed with respect to a so-called *separating linear form* $t = a_1 \omega + a_2 x \in \mathbb{Q}[\omega, x]$ for certain $a = (a_1, a_2) \in \mathbb{Q}^2$, that takes different values when evaluated at the different points of $V(\langle \mathcal{R}, \mathcal{J} \rangle)$. Using (5), the solutions of $V(\langle \mathcal{R}, \mathcal{J} \rangle)$ are then defined by the following rational parametrization

$$f_a(t) = 0, \quad \omega = \frac{g_{a,\omega}(t)}{g_a(t)}, \quad x = \frac{g_{a,x}(t)}{g_a(t)}, \tag{6}$$

where $f_a, g_a, g_{a,\omega}, g_{a,x} \in \mathbb{Q}[t]$ and f_a and g_a satisfy $\gcd(f_a, g_a) = 1$.

Computing a RUR requires solving the following two problems:

- Find a separating linear form $t = a_1 \omega + a_2 x$ for $V(\langle \mathcal{R}, \mathcal{J} \rangle)$.
- Given a linear form $t = a_1 \omega + a_2 x \in \mathbb{Q}[\omega, x]$, compute a RUR-candidate, that is to say the polynomials $f_a, g_a, g_{a,\omega}$ and $g_{a,x}$.

Computation of the RUR-candidate. Given a polynomial h in $\mathbb{Q}[\omega, x]$ (not necessarily separating), a RUR-candidate with respect to h can be computed using an algorithm given in [25, 27]. This algorithm requires the knowledge of a \mathbb{K} -basis of the finite \mathbb{K} -vector space $\mathcal{A} := \mathbb{K}[\omega, x] / \langle \mathcal{R}, \mathcal{J} \rangle$ and a *reduction algorithm* which computes *normal forms* modulo the ideal $\langle \mathcal{R}, \mathcal{J} \rangle$ [8]. 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 $m_h : \mathcal{A} \rightarrow \mathcal{A}$ defined by $m_h(\bar{p}) = \overline{hp}$, where \bar{p} denotes the residue class of $p \in \mathbb{Q}[\omega, x]$ in \mathcal{A} (i.e., modulo $\langle \mathcal{R}, \mathcal{J} \rangle$). A representative of \bar{p} is the *normal form* of p with respect to the *Gröbner basis* of $\langle \mathcal{R}, \mathcal{J} \rangle$. For more details, see [8].

Given a \mathbb{K} -basis $\{e_1, \dots, e_n\}$ of \mathcal{A} – which can be deduced from a Gröbner basis computation of $\langle \mathcal{R}, \mathcal{J} \rangle$ for the *graded reverse lexicographic order* [8] – we can compute the $(n \times n)$ -matrix $M_{a_1 \omega + a_2 x}$ associated to $m_{a_1 \omega + a_2 x}$. The polynomial f_a of the RUR-candidate is defined as the characteristic polynomial of the matrix $M_{a_1 \omega + a_2 x}$. If $\bar{f}_a := \frac{f_a}{\gcd(f_a, \frac{df_a}{dt})} = \sum_{i=0}^d v_i t^{d-i} \in \mathbb{Q}[t]$ is the *square-free part* of f_a of degree d , and if we note $H_j = \sum_{i=0}^j v_i t^{j-i} \in \mathbb{Q}[t]$ for $j = 0, \dots, d-1$, then we have:

$$\begin{cases} g_a = \sum_{i=0}^{d-1} \text{Trace}(M_{a_1 \omega + a_2 x}^i) H_{d-i-1}, \\ g_{a, \omega} = \sum_{i=0}^{d-1} \text{Trace}(M_{\omega} M_{a_1 \omega + a_2 x}^i) H_{d-i-1}, \\ g_{a, x} = \sum_{i=0}^{d-1} \text{Trace}(M_x M_{a_1 \omega + a_2 x}^i) H_{d-i-1}. \end{cases}$$

Finding of a separating linear form. For the computation of a separating polynomial of $V(\langle \mathcal{R}, \mathcal{J} \rangle)$, 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 $V(\langle \mathcal{R}, \mathcal{J} \rangle)$, that is the number of lines passing by two distinct points of $V(\langle \mathcal{R}, \mathcal{J} \rangle)$. Thus, a separating form can always be found among the set $\{t = \omega + ax \mid a = 0, \dots, m\}$. On the other hand, the *Bezout theorem* [1, 8] states that for polynomials \mathcal{R} and \mathcal{J} of total degree respectively d_1 and d_2 , the cardinal of $V(\langle \mathcal{R}, \mathcal{J} \rangle)$ is bounded by $d_1 d_2$. Hence, a strategy for computing a separating element for $V(\langle \mathcal{R}, \mathcal{J} \rangle)$ is to consider $m := d_1 d_2 (d_1 d_2 - 1)/2 + 1$ distinct integers a and for each a , compute the number of distinct roots of the polynomial f_a (see above), i.e., the degree of its squarefree part \bar{f}_a , and finally select an a for which this number is maximal. This ensures that the degree of \bar{f}_a is equal to the cardinal of $V(\langle \mathcal{R}, \mathcal{J} \rangle)$, and thus that the roots of f_a are in bijection with the points of $V(\langle \mathcal{R}, \mathcal{J} \rangle)$. This strategy is actually time-consuming since it requires the computation of m characteristic polynomials as well as their squarefree 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 it separates the points of $V(\langle \mathcal{R}, \mathcal{J} \rangle)$ a posteriori of the computation of a RUR-candidate. This test is based on the fact that a linear form t is separating for $V(\langle \mathcal{R}, \mathcal{J} \rangle)$ if and only if the polynomials $u_{\omega} := g_a \omega - g_{a, \omega}$ and $u_x := g_a x - g_{a, x}$ vanish on all the points of $V(\langle \mathcal{R}, \mathcal{J} \rangle)$, i.e., belong to the *radical* of $\langle \mathcal{R}, \mathcal{J} \rangle$ [8]. In computational terms, this condition can be translated into $(\text{Trace}(M_{u_{\omega} e_i}))_{i=1, \dots, n} = (0, \dots, 0)$ and $(\text{Trace}(M_{u_x e_i}))_{i=1, \dots, n} = (0, \dots, 0)$ [25, 27].

Efficient RUR algorithm. The above algorithm for the computation of the RUR is designed for general zero-dimensional systems and is thus not optimized for the specific system (3) formed by two polynomials in two variables (i.e., ω, x). The computation of a Gröbner basis of $\langle \mathcal{R}, \mathcal{C} \rangle$, required for the computation of the RUR-candidate as well as for the separating form, is usually time-consuming.

Alternatively, we propose below an efficient method, developed in [4], that computes a decomposition of the solutions of (3) into *univariate representations*. This method avoids the computation of a Gröbner basis. This method first consists in applying a change of variables on the variables (ω, x) and then using resultants and subresultants polynomials to compute parametrizations of the solutions of (3). These parametrizations then encode the solutions of the system in a one-to-one correspondence with the roots of a univariate polynomial provided that the change of variables shears the system into *generic position*. Let us introduce this last definition.

Definition 2 Let $\mathcal{R}(\omega, x), \mathcal{J}(\omega, x) \in \mathbb{Q}[\omega, x]$. If $\#S$ denotes the cardinality of a finite set S , then the system $\{\mathcal{R}, \mathcal{J}\}$ is said to be in *generic position* if we have:

$$\forall \alpha \in \mathbb{C}, \quad \#\{\beta \in \mathbb{C} \mid \mathcal{R}(\alpha, \beta) = \mathcal{J}(\alpha, \beta) = 0\} \leq 1.$$

Let us start with the following theorem which shows that there exists $t = \omega + ax$ such that, up to a factor in \mathbb{Q} , the polynomial f_a is equal to the resultant of two polynomials obtained from \mathcal{R} and \mathcal{S} after a change of variables.

Theorem 3 ([4]) Let $\mathcal{R}(\omega, x), \mathcal{S}(\omega, x) \in \mathbb{Q}[\omega, x]$. Define

$$\mathcal{R}'(t, x) = \mathcal{R}(t - ax, x), \quad \mathcal{S}'(t, x) = \mathcal{S}(t - ax, x),$$

where $a \in \mathbb{Z}$ is such that the leading coefficient of \mathcal{R}' and \mathcal{S}' with respect to x are coprime. Then, the resultant of \mathcal{R}' and \mathcal{S}' with respect to x is then equal to

$$\text{Res}_x(\mathcal{R}', \mathcal{S}') = c \prod_{(\alpha_1, \alpha_2) \in V(\langle \mathcal{R}, \mathcal{S} \rangle)} (t - \alpha_1 - a\alpha_2)^{\mu(\alpha_1, \alpha_2)} = c f_a(t),$$

where $c \in \mathbb{Q}$ and $\mu(\alpha_1, \alpha_2)$ denotes the multiplicity of $(\alpha_1, \alpha_2) \in V(\langle \mathcal{R}, \mathcal{S} \rangle)$ [8].

Given a linear form $t = \omega + ax$, it can be shown that it is separating for $V(\langle \mathcal{R}, \mathcal{S} \rangle)$ if and only if the system $\{\mathcal{R}', \mathcal{S}'\}$ is in generic position (see Definition 2). Algebraically, this means that for each root α of $\text{Res}_x(\mathcal{R}', \mathcal{S}')$ (where \mathcal{R}' and \mathcal{S}' are defined as in Theorem 3), the gcd of $\mathcal{R}'(\alpha, x)$ and $\mathcal{S}'(\alpha, x)$, denoted $\mathcal{G}(\alpha, x)$, has exactly one distinct root.

To check the above genericity condition, we consider a *triangular description* of the solutions of $\{\mathcal{R}', \mathcal{S}'\}$ given by a finite union of *triangular systems*:

$$V(\langle \mathcal{R}', \mathcal{S}' \rangle) = \bigcup_{k=1}^l \{(\alpha, \beta) \in \mathbb{C}^2 \mid r_k(\alpha) = \mathcal{G}_k(\alpha, \beta) = 0\}.$$

Such a triangular description can be obtained via a *triangular decomposition algorithm* based on the resultant and subresultant polynomials (see Algorithm 1 of [4] for more details). Such a triangular decomposition yields a set of triangular systems of the form $\{r_k(t), \text{Sres}_k(t, x)\}_{k=1, \dots, l}$, where $l = \min\{\deg_x \mathcal{R}', \deg_x \mathcal{S}'\}$, $\text{Res}_x(\mathcal{R}', \mathcal{S}') = \prod_{k=1}^l r_k(t)$, $r_k \in \mathbb{K}[t]$ is the factor of $\text{Res}_x(\mathcal{R}', \mathcal{S}')$ (possibly equal to 1) whose roots α 's satisfy the property that the degree of $\mathcal{G}(\alpha, x)$ (i.e., the gcd of $\mathcal{R}'(\alpha, x)$ and $\mathcal{S}'(\alpha, x)$) in x is equal to k and $\text{Sres}_k(t, x) = \sum_{i=0}^k \text{sr}_{k,i}(t) x^i$ is the k^{th} subresultant of \mathcal{R}' and \mathcal{S}' . Once a triangular decomposition $\{r_k(t), \text{Sres}_k(t, x)\}_{k=1, \dots, l}$ of $\{\mathcal{R}', \mathcal{S}'\}$ is computed, the genericity condition is equivalent to the fact that $\text{Sres}_k(t, x)$ can be written as $(a_k(t)x - b_k(t))^k$ modulo $r_k(t)$, with $\gcd(a_k, b_k) = 1$. The next theorem checks this last condition.

Theorem 4 ([9]) Let $\mathcal{R}(\omega, x), \mathcal{S}(\omega, x) \in \mathbb{Q}[\omega, x]$. Define the polynomials $\mathcal{R}'(t, x), \mathcal{S}'(t, x)$ as in Theorem 3, and let $\{r_k(t), \text{Sres}_k(t, x)\}_{k=1, \dots, l}$ be the triangular decomposition of $\{\mathcal{R}', \mathcal{S}'\}$. Then, $\{\mathcal{R}', \mathcal{S}'\}$ is in generic position if and only if we have

$$k(k-i) \text{sr}_{k,i} \text{sr}_{k,k} - (i+1) \text{sr}_{k,k-1} \text{sr}_{k,i+1} = 0 \pmod{r_k}, \quad (7)$$

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

If $\{\mathcal{R}', \mathcal{S}'\}$ is in generic position, i.e., if (7) are satisfied, then we obtain that

$$\text{Sres}_k(t, x) = \sum_{i=0}^k \text{sr}_{k,i}(t) x^i = (a_k(t)x - b_k(t))^k \pmod{r_k(t)},$$

with $\gcd(a_k, b_k) = 1$. An explicit expression for x can be obtained by differentiating $(k-1)$ -times $\text{Sres}_k(t, x)$ with respect to x , which yields $x = -\frac{b_k(t)}{a_k(t)} = -\frac{\text{sr}_{k,k-1}(t)}{k \text{sr}_{k,k}(t)}$. Finally, the solutions of the system $\{\mathcal{R}, \mathcal{I}\}$ can be parametrized as follows:

$$r_k(t) = 0, \quad x = -\frac{\text{sr}_{k,k-1}(t)}{k \text{sr}_{k,k}(t)}, \quad \omega = t - a \frac{\text{sr}_{k,k-1}(t)}{k \text{sr}_{k,k}(t)}, \quad k = 1, \dots, l. \quad (8)$$

Numerical approximations. Once a RUR of the solutions of (3) is computed (respectively a parametrization by means of the subresultants (8)), we can obtain certified numerical approximations of these solutions by first isolating the real roots of f_a (resp., r_k) by means of intervals using, for instance, the algorithm in [26], and then substituting these intervals into the rational fractions $g_{a,\omega}/g_a$ and $g_{a,x}/g_a$ (resp., $t - a \text{sr}_{k,k-1}(t)/k \text{sr}_{k,k}(t)$ and $-\text{sr}_{k,k-1}(t)/k \text{sr}_{k,k}(t)$) in order to get isolating intervals for the coordinates ω and x of the solutions of (3). Moreover, substituting these intervals into (4) yields intervals for the delays corresponding to each solution.

An efficient algorithm for the RUR computation is implemented in the Maple command `RationalUnivariateRepresentation` of the Groebner package. The computation of the parametrization (8) is done in the external library RS [28]. 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 2 We consider $f(s, \tau) = e^{-3\tau s} - 3e^{-2\tau s} + 3e^{-\tau s} + s^4 + 2s^2$ studied in [16]. To compute the critical pairs of f , we consider the polynomial system

$$\begin{cases} \mathcal{R}(\omega, x) = (x^3 - 3x) \omega^4 + (-2x^3 + 6x) \omega^2 + x^3 - 3x = 0, \\ \mathcal{I}(\omega, x) = (3x^2 - 1) \omega^4 + (-6x^2 + 2) \omega^2 + 3x^2 + 7 = 0, \end{cases} \quad (9)$$

obtained by substituting $s = i\omega$ and $e^{-\tau i\omega} = \frac{x-i}{x+i}$ into $f(s, \tau)$ and separating the real and imaginary parts of the numerator of the result, as well as the univariate polynomial $f(i\omega, 2k\pi) = (\omega^2 - 1)^2$ (see Remark 1), whose solutions are $\{-1, -1, 1, 1\}$. The latter polynomial yields the critical pairs $(i, 2k\pi)$ and $(-i, 2k\pi)$ where $k \in \mathbb{Z}$. Computing the RUR of (9) with respect to the separating form $h = \omega + x$, we obtain:

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

Then, isolating the real roots of f_a , we obtain the two real roots

$$\begin{aligned} t_1 &\in \left[-\frac{268918098581}{137438953472}, -\frac{67229524645}{34359738368} \right], & t_1 &\approx -1.956636687, \\ t_2 &\in \left[\frac{67229524645}{34359738368}, \frac{268918098581}{137438953472} \right], & t_2 &\approx 1.956636687, \end{aligned}$$

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 $t^4 - 2t^2 - 7$ 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 (4). 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 the parametrization (8) of (9). Since all the solutions of (9) are simple (i.e., their multiplicities are equal to 1), this yields the following single parametrization:

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

The two algorithms, based respectively on the computation of (6) and of (8), are implemented in `Maple`. We report below their running times (on a laptop with an Intel *i-7* processor and L8 cache) for randomly generated quasipolynomials of total degree \deg (i.e., the maximum of the total degree of the monomials in s and $z = e^{-\tau s}$ in the quasipolynomial $f(s, \tau)$) and density dens (sparse/dense).

\deg	dens	numsol	timing for (6)	timing for (8)
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, the timings are the running times in CPU seconds and numsol stands for the average number of critical pairs (ω, x) .

3 Computing Puiseux series and asymptotic stability analysis

In this section, we first provide basic definitions and algorithms for the computation of *Puiseux series* in the case of polynomials. This problem is well-studied in the computer algebra literature [10, 22, 23]. We then explain how these results can be adapted to the case of quasipolynomials. As shown in [16, 15], Puiseux series can be used to effectively decide the stability of a linear time-invariant differential system with commensurate delays (multiples of $\tau \in \mathbb{R}_{>0}$) when τ changes.

Let $F \in \mathbb{K}[X, Y]$ be a bivariate polynomial, where \mathbb{K} is a field of characteristic 0 (e.g. $\mathbb{K} = \mathbb{Q}, \mathbb{R}$ or \mathbb{C}). In what follows, we shall assume that the polynomial F is *square-free* and *monic* in Y , i.e., as a polynomial in Y with coefficients in $\mathbb{K}[X]$, F has no factor of multiplicity 2 and its leading coefficient is 1. Let $\overline{\mathbb{K}}$ denote an *algebraic closure* of \mathbb{K} . With these notations, we have the following *Puiseux theorem*.

Theorem 5 *Let $d = \deg_Y F$ be the degree of $F \in \mathbb{K}[X, Y]$ in Y and ζ_e the e^{th} root of unity where $e \in \mathbb{Z}_{>0}$. Given $x_0 \in \overline{\mathbb{K}}$, there exist positive integers e_1, \dots, e_s with $\sum_{i=1}^s e_i = d$ and d distinct series*

$$S_{ij}(X) = \sum_{k=0}^{\infty} \alpha_{ik} \zeta_{e_i}^{jk} (X - x_0)^{\frac{k}{e_i}}$$

such that $F(X, S_{ij}(X)) = 0$ for $1 \leq i \leq s$ and $0 \leq j \leq e_i - 1$.

Over most points x_0 , the polynomial $F(x_0, Y)$ has d distinct roots. In this case, the *Implicit Function Theorem* ensures that the series solutions are Taylor series. Such series can be quickly computed using, e.g., *quadratic Newton iterations* [14]. When dealing with multiple roots over a point x_0 , this may not be the case anymore.

In what follows, we shall compute the Puiseux series using a variant of the well-known *Newton-Puiseux algorithm* [29, 3], namely the *rational Newton-Puiseux algorithm* [10]. Let us explain the main idea of this algorithm by means of an example.

Example 3 Let $H = Y^6 + XY^5 + 5X^3Y^4 - 2XY^4 + 4X^2Y^2 + X^5 - 3X^4$ and let us compute the Puiseux series at $x_0 = 0$. Note that we can always be in this situation via a change of the variable $X \leftarrow X + x_0$. From Theorem 5, we know that the first term of any series $Y(X)$ is of the form $\alpha X^{\frac{m}{q}}$, $\alpha \in \mathbb{K}$ and $(m, q) \in \mathbb{N}^2$. We then get:

$$H\left(X, \alpha X^{\frac{m}{q}} + \dots\right) = \alpha^6 X^{\frac{6m}{q}} + \alpha^5 X^{\frac{5m}{q}+1} + 5\alpha^4 X^{\frac{4m}{q}+3} - 2\alpha^4 X^{\frac{4m}{q}+1} \\ + 4\alpha^2 X^{\frac{2m}{q}+2} + X^5 - 3X^4 + \dots$$

To get $H(X, Y(X)) = 0$, 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.

Definition 3 The *support* of a polynomial $F(X, Y) = \sum_{i,j} a_{ij} X^j Y^i$ is defined by:

$$\text{Supp}(F) = \{(i, j) \in \mathbb{N}^2 \mid a_{ij} \neq 0\}.$$

Example 3 (continued) $\text{Supp}(H) = \{(6, 0), (5, 1), (4, 3), (4, 1), (2, 2), (0, 5), (0, 4)\}$.

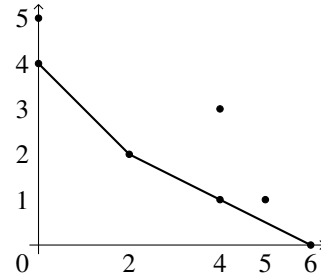
The condition on the pairs (m, q) can be translated as follows: two points of $\text{Supp}(F)$ must lie on the same line $mi + qj = l$ for a certain $l \in \mathbb{Z}$. To increase the X -order of the valuation, there must be no point under this line. This leads us to the following definition.

Definition 4 The *Newton polygon* $\mathcal{N}(F)$ of $F \in \mathbb{K}[X, Y]$ is the lower part of the *convex hull* of its support $\text{Supp}(F)$.

Example 3 (continued) As illustrated in the figure below, $\mathcal{N}(H)$ is given by two edges: $\Delta_1 = ((6, 0), (2, 2))$ – corresponding to the line $i + 2j = 6$ (i.e. $m = 1$ and $q = 2$) and $\Delta_2 = ((2, 2), (0, 4))$. The points of $\text{Supp}(H)$ belonging to Δ_1 are $(6, 0)$, $(2, 2)$ and $(4, 1)$. We have

$$H(T^q, \alpha T^m) = H(T^2, \alpha T) \\ = (\alpha^6 - 2\alpha^4 + 4\alpha^2)T^6 + \alpha^5 T^7 + (5\alpha^4 + 1)T^{10} - 3T^8 + \dots$$

Since we want to increase the T -order of $H(T^2, \alpha T)$, we must set α equal to a root of $P = Z^6 - 2Z^4 + 4Z^2$. We first note that $P \in \mathbb{K}[Z^2]$. Indeed, by construction, the polynomial P always belongs to $\mathbb{K}[Z^q]$. Note also that 0 is a root of P and we are not interested in this root (we shall get the first non-zero term of the associated Puiseux series by considering the edge Δ_2). For these reasons, only the roots of the polynomial $\phi = T^2 - 2T + 4$ are interesting. This polynomial is called the *characteristic polynomial* of Δ_1 .



The Newton polygon $\mathcal{N}(H)$

Definition 5 If $F = \sum a_{ij} X^j Y^i \in \mathbb{K}[X, Y]$, then the *characteristic polynomial* ϕ_Δ of $\Delta \in \mathcal{N}(F)$ is defined by $\phi_\Delta(T) = \sum_{(i,j) \in \Delta} a_{ij} T^{\frac{i-j_0}{q}} \in \mathbb{K}[T]$, where i_0 is the smallest value such that (i_0, j_0) belongs to Δ for some j_0 .

Computing more terms. Let us sum up the computation of the first non zero term of the Puiseux series of F at 0: we first compute the Newton polygon $\mathcal{N}(F)$ of F , then the characteristic polynomial ϕ_Δ for all the edges Δ of $\mathcal{N}(F)$, and finally compute the roots of these polynomials. Note that we assume that we can decide whether or not a coefficient of F is 0 to be able to certify the correctness of our computations. At the end of this section, we shall briefly discuss a symbolic-numeric strategy developed in [21, 22] to overcome possibly costly symbolic computations.

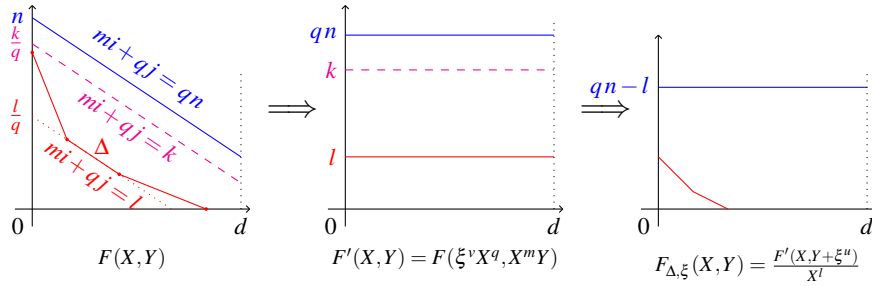


Figure 1: Geometry of a change of variables: each diagonal becomes an horizontal line.

Assuming the generalization of the above results for quasipolynomials (see 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 differential time-delay system. This can actually be done by applying the above strategy to $F(X, Y + \xi^{\frac{1}{q}} X^{\frac{m}{q}}) \in \mathbb{K}(\xi^{\frac{1}{q}})[X^{\frac{1}{q}}, Y]$, or similarly to $F(X^q, Y + \xi^{\frac{1}{q}} X^m) \in \mathbb{K}(\xi^{\frac{1}{q}})[X, Y]$. To avoid taking a q^{th} root of ξ , following [10, section 4], we can consider the following polynomial

$$F_{\Delta, \xi}(X, Y) = \frac{F(\xi^v X^q, X^m(Y + \xi^u))}{X^l} \in \mathbb{K}(\xi)[X, Y],$$

where $u, v \in \mathbb{Z}$ are such that $uq - mv = 1$ and $mi + qj = l$ with $(i, j) \in \Delta$. Figure 1 illustrates such a transformation.

Let us now introduce the concept of a *rational Puiseux expansion*.

Definition 6 A *rational Puiseux expansion* over \mathbb{K} of F above 0 is a pair of non-constant formal power series $(X(T), Y(T)) = (\lambda T^e, \sum_{j=0}^{\infty} \beta_j T^j)$ such that:

1. $e > 1$ and $\lambda \neq 0$.
2. $(X(T), Y(T))$ is a *parametrization* of F , i.e., $F(X(T), Y(T)) = 0$.
3. The parametrization $(X(T), Y(T))$ of F is *irreducible*, namely e is minimal:

$$\forall u > 1, \quad Y(T) \notin \mathbb{K}[[T^u]].$$

From a rational Puiseux expansion of F , we can deduce e Puiseux series:

$$Y_k(X) = Y \left(\zeta_e^k (\lambda^{-1} X)^{\frac{1}{e}} \right) = \sum_{j=0}^{\infty} \beta_j \lambda^{-j} \zeta_e^{kj} X^{\frac{j}{e}}, \quad 0 \leq k \leq e-1.$$

Next page, the *rational Newton-Puiseux algorithm* [10, 22], which computes truncated rational Puiseux expansions of $F(X, Y) = 0$ at $x_0 = 0$, is given. It uses the subroutine `Factor`(\mathbb{K}, ϕ), that given a field \mathbb{K} and $\phi \in \mathbb{K}[T]$, returns a finite set $\{(\phi_i, M_i)\}_{i \in I}$ of its irreducible factors and the associated multiplicities, i.e. $\phi_i \in \mathbb{K}[T]$ is a monic and irreducible polynomial and

$$\phi = c \prod_{i \in I} \phi_i^{M_i},$$

```

Input:  $F \in \mathbb{K}[X, Y]$ , monic and squarefree with  $\mathbb{K}$  a field
          $\pi$  the result of previous computations ( $\pi = (X, Y)$  for the initial call)
Output: Truncated rational Puiseux expansions of  $F$ 
1  $\mathcal{R} \leftarrow \{\}$ ; // results of the algorithm will be grouped in  $\mathcal{R}$ 
2 foreach  $\Delta \in \mathcal{N}(F)$  do
3   Compute  $m, q, l \in \mathbb{N}$  s.t.  $q$  and  $m$  are coprime and  $mi + qj = l$  for all  $(i, j) \in \Delta$ ;
4   Compute  $(u, v) \in \mathbb{Z}^2$  such that  $uq - mv = 1$  and  $\phi_\Delta$  associated to  $\Delta$  (see Definition 5);
5   foreach  $(\phi, M)$  in  $\text{Factor}(\mathbb{K}, \phi_\Delta)$  do
6     Consider a new symbol  $\xi$  satisfying  $\phi(\xi) = 0$ ;
7      $\pi_1 = \pi(\xi^v X^q, X^m(Y + \xi^u))$ ;
8     if  $M = 1$  then  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(\pi_1(T, 0), \mathbb{K}(\xi))\}$ ;
9     else
10       $G(X, Y) \leftarrow F(\xi^v X^q, X^m(Y + \xi^u))/X^l$ ; // Puiseux transformation
11       $\mathcal{R} \leftarrow \mathcal{R} \cup \text{RNPuiseux}(G, \mathbb{K}(\xi), \pi_1)$ ;
12 return  $\mathcal{R}$ ;

```

where $c \in \mathbb{K}$ and $M_i \in \mathbb{Z}_{>0}$ for $i \in I$.

The quasipolynomial case. As mentioned in [15], we can approximate the quasipolynomial by truncated power series. In [22, 23], 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, we can replace any coefficient access in the polynomial case by an evaluation of some derivatives of the quasipolynomial [15]. For instance, if $f(s, \tau)$ is a quasipolynomial, then $\left(\frac{1}{2!1!} \frac{\partial^3 f}{\partial s^2 \partial \tau}\right)(\tau_0, s_0)$ corresponds to the coefficient of the term $(s - s_0)^2 (\tau - \tau_0)$.

For higher order terms, instead of evaluating the derivatives of f at a point, we can evaluate f at the truncated Puiseux series already computed (e.g., via the parameter π of Algorithm `RNPuiseux`). It would also be interesting to develop such a strategy and compare the two approaches.

Finally, the condition on line 8 in Algorithm `RNPuiseux` has to be changed. Indeed, the aim of this algorithm is to *desingularize* the curve $F(X, Y) = 0$, i.e., to stop when the multiplicity of the root ξ is 1. In our context, we stop when the computed coefficient has a non-zero real part. This can happen before or after the condition $M = 1$ is reached.

A symbolic-numeric strategy. In [21, 22], 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* of the series (exponents, etc.), and then to use it to conduct purely numeric operations. Adapting this strategy to quasi-polynomials is an interesting challenge which will be studied.

Examples. We implemented a `Maple` prototype to compute the Puiseux series of a quasipolynomial. Note that this is a symbolic algorithm. We illustrate the results with examples and we show the logs of some computations (including the timing).

Example 4 We consider again the quasipolynomial defined in Example 2 and compute its Puiseux series at $(\tau_0, s_0) = (2\pi, i)$. We get the following:

```
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
```

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

Example 5 Let us consider the quasipolynomial $F(\tau, s) = \sum_{k=0}^4 a_k(s) e^{-ks\tau}$, where

$$\begin{cases} a_4(s) = s^3 + 2s^2 + 2s + 1, \\ a_3(s) = 3s^3 + 9s^2 + 9s + 4, \\ a_2(s) = \frac{5}{4}\pi s^5 + \frac{11}{4}\pi s^4 + 3s^3 - \pi s^3 + \frac{1}{2}s^2\pi + 13s^2 + 15s - \frac{9}{4}s\pi + 6 - \frac{9}{4}\pi, \\ a_1(s) = \frac{5}{4}\pi s^5 + \frac{11}{2}\pi s^4 + s^3 + \frac{7}{2}\pi s^3 + s^2\pi + 7s^2 + 11s + \frac{9}{4}s\pi + 4 - \frac{9}{2}\pi, \\ a_0(s) = 1 + 3s + \frac{9}{2}\pi s^3 + s^2 - \frac{45}{8}\pi^2 + \frac{9}{2}s\pi - \frac{15}{8}s^4\pi^2 + \frac{1}{2}s^2\pi - \frac{75}{8}s^2\pi^2 \\ \quad - \frac{9}{4}\pi + \frac{11}{4}\pi s^4 + \frac{15}{8}\pi^2 s^6. \end{cases}$$

considered in [17]. Computing the Puiseux series at the point $(5\pi, i)$, we obtain:

```
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
```

We obtain the following two Puiseux series:

$$\begin{cases} \Delta s = -0.1591i\Delta\tau - (0.5371 - 0.3644i) \\ \Delta s = -0.0987i(\Delta\tau)^{1/3} - (0.03557 - 0.0028i)(\Delta\tau)^{2/3}. \end{cases}$$

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

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

This explains that the running time is longer than for Example 4. This fact advocates for the development of an efficient symbolic-numeric strategy.

References

- [1] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, 2nd ed., ser. Algorithms and Computation in Mathematics. Springer-Verlag, 2006, vol. 10.
- [2] R. E. Bellman, K. L. Cooke, *Differential-Difference Equations*, The Rand Corporation, 1963.
- [3] E. Brieskorn and H. Knörrer, *Plane Algebraic Curves*. Birkhäuser, 1986.

- [4] 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.
- [5] Y. Bouzidi, S. Lazard, G. Moroz, M. Pouget, F. Rouillier, M. Sagraloff, “Solving bivariate systems using Rational Univariate Representations”, *Journal of Complexity*, vol. 37, 2016, pp. 34–75.
- [6] Y. Bouzidi, A. Poteaux, A. Quadrat, “Computer algebra methods for the stability analysis of differential systems with commensurate time-delays”, *Proceedings of the 13th IFAC Workshop on Time Delay Systems*, Istanbul (Turkey), 22-24/06/2016.
- [7] G. E. Collins and A. G. Akritas, “Polynomial real root isolation using descartes’s rule of signs,” in *Proceedings of the third ACM symposium on Symbolic and algebraic computation*. ACM, 1976, pp. 272–275.
- [8] D. Cox, J. Little, and D. O’Shea, *Ideals, Varieties, and Algorithms*, 3rd ed., ser. Undergraduate Texts in Mathematics. Springer-Verlag, 2007.
- [9] D. N. Daouda, B. Mourrain, and O. Ruatta, “On the computation of the topology of a non-reduced implicit space curve,” in *ISSAC’08: Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*. New York, ACM, 2008, pp. 47–54.
- [10] D. Duval, “Rational Puiseux expansions,” *Compositio Math.*, vol. 70, no. 2, pp. 119–154, 1989.
- [11] K. Gu, V. L. Kharitonov, and J. Chen, *Stability of Time-Delay Systems*. Birkhäuser, 2003.
- [12] J. K. Hale, *Theory of Functional Differential Equations*, Springer, 1977.
- [13] A. Kobel, F. Rouillier, M. Sagraloff, “Computing Real Roots of Real Polynomials ... and now for real!”, *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 2016.
- [14] H. T. Kung and J. F. Traub, “All algebraic functions can be computed fast,” *J. ACM*, vol. 25, no. 2, pp. 245–260, 1978.
- [15] X.-G. Li, S.-I. Niculescu, A. Çela, H.-H. Wang, and T.-Y. Cai, “On computing Puiseux series for multiple imaginary characteristic roots of LTI systems with commensurate delays,” *Automatic Control, IEEE Transactions on*, vol. 58, no. 5, pp. 1338–1343, 2013.
- [16] X.-G. Li, S.-I. Niculescu, and A. Çela, *Analytic Curve Frequency-Sweeping Stability Tests for Systems with Commensurate Delays*. Springer, 2015.
- [17] X.-G. Li, S.-I. Niculescu, A. Çela, and L. Zhang, “Characterizing invariance property of uniformly distributed delay systems.” http://web1.lss.supelec.fr/delsys2015/Site/Program_files/Wed7_LI.pdf.
- [18] J. E. Marshall, H. Gorecki, A. Korytowski, and K. Walton, *Time-Delay. Systems: Stability and Performance Criteria with Applications*. London: Ellis Horwood, 1992.
- [19] S.-I. Niculescu, *Delay Effects on Stability: A Robust Control Approach*, ser. Lecture Notes in Control and Information Sciences. Springer, 2001, no. 269.
- [20] N. Olgac and R. Sipahi, “An exact method for the stability analysis of time-delayed linear time-invariant (LTI) systems,” *IEEE Transactions on Automatic Control*, vol. 47, no. 5, pp. 793–796, 2002.
- [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.

- [22] A. Poteaux, “Calcul de développements de puiseux et application au calcul de groupe de monodromie d’une courbe algébrique plane,” Ph.D. dissertation, Université de Limoges, 2008.
- [23] A. Poteaux and M. Rybowicz, “Complexity bounds for the rational Newton-Puiseux algorithm over finite fields,” *Applicable Algebra in Engineering, Communication and Computing*, vol. 22, pp. 187–217, 2011.
- [24] Z. V. Rekasius, “A stability test for systems with delays,” in *Proc. Joint Automatic Control Conf.*, 1980.
- [25] F. Rouillier, “Solving zero-dimensional systems through the rational univariate representation,” *J. of Applicable Algebra in Engineering, Communication and Computing*, vol. 9, no. 5, pp. 433–461, 1999.
- [26] F. Rouillier and P. Zimmermann, “Efficient isolation of polynomial real roots,” *J. of Computational and Applied Mathematics*, vol. 162, no. 1, pp. 33–50, 2003.
- [27] F. Rouillier, “Algorithmes pour l’étude des solutions réelles des systèmes polynomiaux”, Habilitation Thesis, University Paris 6, <https://hal.inria.fr/tel-01435142>.
- [28] F. Rouillier, RS: Real Roots isolation for algebraic systems with rational coefficients with a finite number of complex roots, <https://who.rocq.inria.fr/Fabrice.Rouillier/software.php>.
- [29] R. J. Walker, *Algebraic Curves*. Springer-Verlag, 1950.



**RESEARCH CENTRE
LILLE – NORD EUROPE**

Parc scientifique de la Haute-Borne
40 avenue Halley - Bât A - Park Plaza
59650 Villeneuve d'Ascq

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399