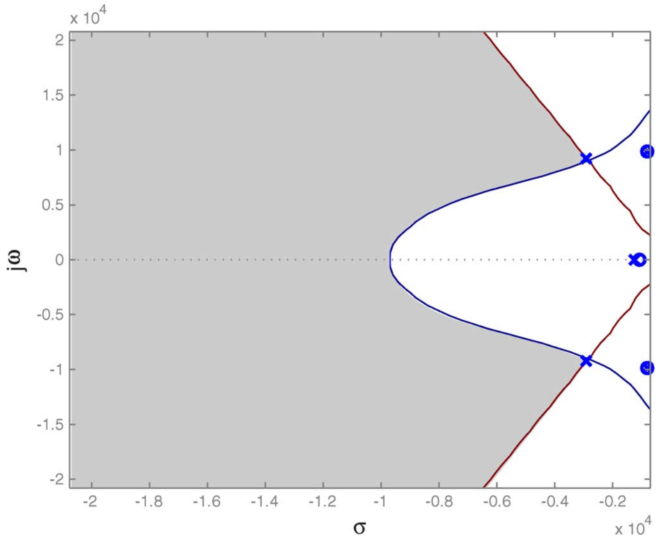

Fig. 5. Admissible domain for the hard disk drive and pole-zero map of the closed loop system.

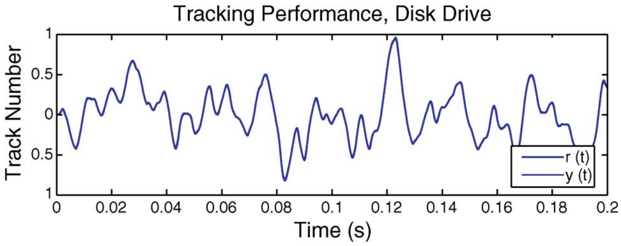

Fig. 6. Tracking performance of the hard disk drive system.

where

$$K = 5.7214 \times 10^5. \tag{27}$$

Applying this controller to $P_D(s)$, results in the closed loop pole-zero configuration shown in Fig. 5. Note that this controller leads to a number of approximate stable pole-zero cancellations and, consequently, yields a pair of dominant poles at the desired locations. The resulting tracking error is $\sigma_e = 0.0425\sigma_r$, which satisfies the design specifications given in (24). Fig. 6 illustrates the quality of tracking in the time domain, where the output completely overlays the reference signal.

It is interesting to compare $C(s)$ to the controller presented in [6]. In particular, in [6], a fifth order controller was designed using an $\mathcal{H}\infty$ method, achieving an error variance of $\sigma_e = 0.056\sigma_r$. Clearly, we are able to exceed this performance with a fourth order controller using simple pole placement in an appropriately calculated admissible domain.

VI. Conclusion

While it is clear that admissible pole domains for tracking band-limited signals are larger than those for tracking steps, this understanding must be quantified to be useful for controller design. Such a quantification is carried out in this note. In particular, it is shown that level curves of the tracking quality indicators can be viewed as boundaries of admissible domains in the same manner as rise time and overshoot are utilized for tracking steps. Based on the technique developed, we are able to design controllers using classical techniques with demonstrably good performance.

References

[1] B. Kuo and F. Golnaraghi, *Automatic Control Systems*, 8th ed. New York: Wiley, 2003.

[2] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2002.

[3] E. J. Ohlmeyer, "Root-mean-square miss distance of proportional navigation missile against sinusoidal target," *J. Guid., Control Dyn.*, vol. 19, no. 3, pp. 563–568, 1996.

[4] H. S. Bae and J. C. Gerdes, "Command Modification Using Input Shaping for Automated Highway Systems With Heavy Trucks," California PATH Research Rep. UCB-ITS-PRR-2004-48, 2004.

[5] T. B. Goh, Z. Li, and B. M. Chen, "Design and implementation of a hard disk servo system using robust and perfect tracking approach," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 2, pp. 221–233, Mar. 2001.

[6] Y. Eun, P. T. Kabamba, and S. M. Meerkov, "Tracking of random references: Random sensitivity function and tracking quality indicators," *IEEE Trans. Automat. Control*, vol. 48, no. 9, pp. 1666–1671, Sep. 2003.

Mapping Based Algorithm for Large-Scale Computation of Quasi-Polynomial Zeros

Tomáš Vyhlídal and Pavel Zítek

***Abstract*—A method for computing all zeros of a retarded quasi-polynomial that are located in a large region of the complex plane is presented. The method is based on mapping the quasi-polynomial and on utilizing asymptotic properties of the chains of zeros. First, the asymptotic exponentials of the chains are determined based on the distribution diagram of the quasi-polynomial. Secondly, large regions free of zeros are defined. Finally, the zeros are located as the intersection points of the zero-level curves of the real and imaginary parts of the quasi-polynomial, which are evaluated over the areas of the region outside those free of zeros.**

***Index Terms*—Function zeros, quasi-polynomial, retarded system, root-finding techniques, spectrum distribution.**

I. Introduction

In this technical note, we describe an algorithm for efficient computation of all zeros of a retarded quasi-polynomial that are located in a large region $\boldsymbol{D}$ of the complex plane. Even though the spectrum of the quasi-polynomial is infinite, the zeros of large modulus are distributed in a finite number of asymptotic chains [1]. Based on the asymptotic features of these chains, large areas of $\boldsymbol{D}$ free of zeros can be determined. In the algorithm presented, these areas are omitted from the scanning procedure, which increases considerably the efficiency of the algorithm.

Manuscript received November 28, 2006; revised October 16, 2007. Current version published January 14, 2009. This work was supported by the Ministry of Education of the Czech Republic under the Project 1M0567. Recommended by Associate Editor G. Feng.

The authors are with the Center for Applied Cybernetics, Department of Instrumentation and Control Engineering, Faculty of Mechanical Engineering, Czech Technical University in Prague, Praha 6, Czech Republic (e-mail: tomas.vyhlidal@fs.cvut.cz; pavel.zitek@fs.cvut.cz).

Digital Object Identifier 10.1109/TAC.2008.2008345

The retarded quasi-polynomial is considered in the following form:

$$h(s) = \sum_{j=0}^{N} p_j(s) \mathrm{e}^{-s\alpha_j} \tag{1}$$

where $\alpha_0 > \alpha_1 > \ldots > \alpha_{N-1} > \alpha_N = 0$ and $p_j(s) = \sum_{k=0}^{m_j} p_{j,k} s^k$ are polynomials in s of degree m_j at most $n-1$ for $p_j(s)$, $j = 0, 1, \ldots N-1$, where n is degree of $p_N(s)$. Notice that the quasi-polynomial of the form (1) can be considered as numerator $b(s)$ and denominator $a(s)$ of the following meromorphic function

$$G(s) = \frac{b(s)}{a(s)} d(s) \tag{2}$$

where $d(s)$ is zero free exponential function. The function (2) can represent a transfer function of a retarded time delay system. Thus, using the root-finding algorithm described in this technical note, both the poles and zeros of (2) can be computed. Let us remark that the efficient and large scale computation of the spectra of time delay systems has been the primary motivation for development of the root-finding algorithm. Even though the stability of the system is determined by the finite number of rightmost roots, the knowledge of the spectrum distribution on a large scale may provide a deeper understanding of the system dynamics, see e.g. [2].

A. *On Computing the Spectrum of Quasi-Polynomial Zeros*

Obviously, the zeros of (1) are located at the same positions as the roots of the system

$$\mathbf{x}'(t) = \sum_{j=0}^{N} \mathbf{A}_j \mathbf{x}(t - \alpha_j) \tag{3}$$

where $\mathbf{x} \in \mathrm{R}^n$ and $\mathbf{A}_j \in \mathrm{R}^{n \times n}$, $j = 0 \ldots N$ are the constant coefficient matrices. These are given as follows:

$$\mathbf{A}_N = \begin{bmatrix} 0 & 0 & \ldots & 0 & -\widetilde{p}_{N,0} \\ 1 & 0 & \ldots & & -\widetilde{p}_{N,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -\widetilde{p}_{N,n} \end{bmatrix},$$

$$\mathbf{A}_j = \begin{bmatrix} 0 & 0 & \ldots & 0 & -\widetilde{p}_{j,0} \\ 0 & 0 & \ldots & & -\widetilde{p}_{j,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & -\widetilde{p}_{j,n} \end{bmatrix}, \quad j = 0, 1, \ldots, N-1$$

where $\widetilde{p}_{j,k}$ are the coefficients of the polynomials $\widetilde{p}_j(s) = p_j(s)/p_{N,n}$. The problem of computing the rightmost roots of the retarded system of form (3) has been widely studied. The first group of methods derived for approximating the stability determining right-most roots are based on a discretization of the system solution operator, using LMS methods [3], [4] or Runge-Kutta methods [5]. The second group of methods are based on a discretization of the system infinitesimal generator, using, e.g., Runge–Kutta methods [6] or pseudospectral techniques [7]. Useful tools for computing the roots of TDS are DDE-Biftool Matlab package [8] and also Matlab package Trace-DDE [9].

Alternatively, the zeros of (1) can be computed by numerical algorithms developed for computing zeros of general analytic functions, [10], [11]. Traditional methods for computing the zeros of analytic functions are based on iterative schemes, e.g., the well known Newton's method. On the one hand, these methods have very good convergence properties, however an accurate initial guess of the positions of zeros needs to be provided. Therefore, the iterative methods are often used for the final calculation of zeros if their approximate positions are computed using a different method. For example, the starting positions for the iteration methods can be obtained by bisection-based algorithms. These are used for computing all zeros located in a selected region $\boldsymbol{D}$ of the complex plane, see e.g. [12] or [13]. The problem of computing all zeros of an analytic function in $\boldsymbol{D}$ can also be solved using the quadrature methods [10], [14].

In their earlier work, the authors designed the quasi-polynomial mapping based root-finder (QPmR), [15] for computing the zeros of quasi-polynomials. This algorithm will be described briefly in the next subsection.

B. *Quasi-Polynomial Mapping Based Root-Finder*

The QPmR algorithm has been designed to compute all zeros of a quasi-polynomial in region $\boldsymbol{D} = [\beta_{\min}, \beta_{\max}] \times [\omega_{\min}, \omega_{\max}]$ of the complex plane. The main idea of the algorithm is given as follows: Considering $s = \beta + \mathrm{j}\omega$, the characteristic quasi-polynomial $h(s)$ can be split into $R(\beta, \omega) = \mathrm{Re}(h(s))$ and $I(\beta, \omega) = \mathrm{Im}(h(s))$. Consequently the characteristic equation $h(s) = 0$ can be split into

$$R(\beta, \omega) = 0 \tag{4}$$
$$I(\beta, \omega) = 0. \tag{5}$$

Notice that as a geometric representation of these equations, the intersection contours of the surfaces $\boldsymbol{R} = R(\beta, \omega)$ and $\boldsymbol{I} = I(\beta, \omega)$, respectively, with the s-plane, can be considered. Consequently, zeros of (1) are given as the intersection points of these contours described implicitly by (4) and (5). Obviously, an analytical expression of these contours $R(\beta, \omega) = 0$ and $I(\beta, \omega) = 0$ can be obtained only for the most simple quasi-polynomials. In general, a numerical contour plotting algorithm needs to be used to map the contours (such an algorithm is available in most mathematical software: *contour* in Matlab, *contourplot* in Mathematica and Maple, etc., a description of the algorithm can be found e.g. in [16], see also the references therein).

In the practical implementation of the QPmR algorithm, first, the region of interest $\boldsymbol{D}$ is covered by a regular mesh grid

$$\boldsymbol{\Gamma} = \begin{bmatrix} \beta_0 + \mathrm{j}\omega_0 & \cdots & \beta_{k_{\max}} + \mathrm{j}\omega_0 \\ \vdots & \beta_k + \mathrm{j}\omega_l & \vdots \\ \beta_0 + \mathrm{j}\omega_{l_{\max}} & \cdots & \beta_{k_{\max}} + \mathrm{j}\omega_{l_{\max}} \end{bmatrix}$$
$$\beta_k = \beta_{\min} + k\Delta_g, \quad k = 0, 1, \ldots, k_{\max}$$
$$\omega_l = \omega_{\min} + l\Delta_g, \quad l = 0, 1, \ldots, l_{\max} \tag{6}$$

with a grid step Δ_g. Secondly, the functions $h(s)$ is evaluated at each grid point of (6) and split into $R(\beta, \omega)$ and $I(\beta, \omega)$. Next, the contours $R(\beta, \omega) = 0$ and $I(\beta, \omega) = 0$ are mapped by a contour plotting algorithm. Then, the intersection points of the contours $R(\beta, \omega) = 0$ and $I(\beta, \omega) = 0$ are determined as the zeros of the function $\widetilde{I}(\beta, \omega) = \mathrm{Im}(h(s))$ evaluated over the points on the curve $R(\beta, \omega) = 0$, as described in [15]. Let us remark that using this approach, all the zeros are computed within the same accuracy determined by Δ_g. Finally, Newton's iteration method is applied to increase the accuracy of each zero.

A considerable drawback of QPmR is the need to evaluate the functions $h(s)$ at each point of the grid over the region $\boldsymbol{D}$. Thus, if the scanned region $\boldsymbol{D}$ is large and Δ_g small, computation time can be very large (on the other hand, if Δ_g is too large, the contours may not be mapped well, which may result in failure of the algorithm). In order to reduce computation time of QPmR, an advanced version of the algorithm (aQPmR) is introduced in this technical note. In the aQPmR, the features of the retarded system spectrum are utilized in order to reduce the area which is to be scanned for the zeros.

Remark 1: It should be mentioned that the accuracy of the QPmR algorithm may be reduced if the quasi-polynomial (1) is ill-conditioned, see [15]. For example, an ill-conditioned problem may arise if (1) is derived as the characteristic function of the system in the state space description, given e.g. by (3). The rounding errors associated with evaluation of the determinant in $h(s) = \det(s\mathrm{I} - \sum_{j=0}^{N} \mathbf{A}_j \exp(-s\alpha_j))$ may result in considerable displacement of the zeros of $h(s)$ from the roots of the system. For computing roots of such a system, it is more convenient to use one of the discretization based methods mentioned above.

II. Asymptotic Distribution of Quasi-Polynomial Zeros

In this section we adopt the results of [1] concerning the distribution of retarded system roots of large magnitudes. Consider the function

$$g(s) = h(s)\mathrm{e}^{s\alpha_0} = \sum_{j=0}^{N} p_{j,m_j} s^{m_j} \left(1 + \varepsilon_j(s)\right) \mathrm{e}^{s\vartheta_j} \tag{7}$$

where $\vartheta_j = \alpha_0 - \alpha_j$, $0 = \vartheta_0 < \vartheta_1 < \ldots < \vartheta_{N-1} < \vartheta_N$, $p_{j,m_j} \neq 0$ $(j = 0, 1, \ldots, N)$ and the functions $\varepsilon_j(s)$ have the property $\lim_{|s|\to\infty} |\varepsilon_j(s)| = 0$. Obviously, (7) has the same distribution of zeros as (1). As it has been shown in [1], with the points $P_j = (\vartheta_j, m_j)$, we can define a *spectrum distribution diagram* (also called *potential diagram*) of (7), which determines the distribution properties of the root chains.

Proposition 1: Spectrum distribution diagram. Let a polygonal line L be constructed over the points $P_j = (\vartheta_j, m_j)$ satisfying the following features: joins P_0 with P_N, has vertices only at points of the set P_j, is convex upward and is such that no points P_j lie above it. Let the successive segments of L be denoted by $L_1, L_2, \ldots. L_M$, numbered from left to right, and let μ_r denote the slope of L_r. Based on the distribution diagram, for $c_1 \in (0, \infty)$ the complex plane can be divided into a number of regions $V_1, \ldots\ldots, V_M, U_0, \ldots\ldots, U_M$, defined by the inequalities

$$\begin{aligned}
&V_r : -c_1 < \mathrm{Re}(s + \mu_r \ln s) < c_1, \quad r = 1 \ldots M\\
&U_r : c_1 < \mathrm{Re}(s + \mu_r \ln s),\ \mathrm{Re}(s + \mu_{r+1} \ln s) < -c_1\\
&\qquad r = 1 \ldots M - 1\\
&U_0 : \mathrm{Re}(s + \mu_1 \ln s) < -c_1\\
&U_M : \mathrm{Re}(s + \mu_M \ln s) > c_1.
\end{aligned} \tag{8}$$

It has been proved in [1] that outside a sufficiently large circle, these V and U regions do not intersect each other, and the following theorem about the distribution of the zeros of $g(s)$ and its bounds holds.

Theorem 1: Let $g(s)$ be a quasi-polynomial of the form (7). Let the s-plane be divided into regions $V_1, V_2, \ldots, V_M$ and $U_0, U_1, U_2, \ldots, U_M$ in the manner described above. Outside a certain large circle $|s| = c$, the following statements apply:

a) There are no zeros of $g(s)$ in U_r, $(r = 0, 1, \ldots M)$. If $s \in U_r$, the term of $g(s)$ corresponding to the point of the spectrum distribution diagram at the right end of the segment L_r is of predominant order of magnitude.

b) The zeros of $g(s)$ in V_r are asymptotic to those of the comparison function

$$g_r(s) = \sum_{L_r} p_{j,m_j} s^{m_j} \mathrm{e}^{s\vartheta_j} \tag{9}$$

where the notation means that the sum is taken over all terms corresponding to the points on L_r.

c) In any strip V_r, the zeros $g(s)$ lie asymptotically along a finite number of curves

$$\mathrm{Re}(s + \mu_r \ln s) = \mathrm{const.} \tag{10}$$

Proof: See proofs of Theorem 12.8, Theorem 12.9 and Theorem 12.10 in [1]. □

Based on Theorem 1, the following Lemma can be stated

Lemma 1: In each V_r corresponding to the segment of the spectrum distribution diagram L_r, for large magnitudes of $s = \beta + \mathrm{j}\omega$, the asymptotic curves of the chains of zeros of (7) given by (10) can be approximated by the asymptotic exponentials

$$\beta = \mu_r \left(\ln |w_{rk}| - \ln |\omega|\right) \tag{11}$$

where w_{r_k} is a zero of the polynomial

$$f_r(w) = \sum_{j=0}^{N_r} \overline{p}_j w^{\widetilde{m}_j} \tag{12}$$

where $\widetilde{m}_j = \overline{m}_j - \overline{m}_0$, $\overline{m}_j$ and $\overline{p}_j$ correspond to those points $P_j = (\vartheta_j, m_j)$ defined for (7) that lie on the particular segment L_r, i.e. they correspond to those in (9), $\overline{m}_0$ is the exponent of s corresponding to the left end of the segment L_r and $N_r + 1$ is the number of points $(\overline{\vartheta}_j, \overline{m}_j)$ on the segment.

Proof: The result is a direct consequence of Theorem 1: consider the function

$$\widetilde{g}_r(s) = s^{-\overline{m}_0} \mathrm{e}^{-s\overline{\vartheta}_0} g_r(s) = \sum_{j=0}^{N_r} \overline{p}_j s^{\widetilde{m}_j} \mathrm{e}^{s\widetilde{\vartheta}_j} \tag{13}$$

$\widetilde{m}_j = \overline{m}_j - \overline{m}_0$, $\widetilde{\vartheta}_j = \overline{\vartheta}_j - \overline{\vartheta}_0$, where $(\overline{\vartheta}_j, \overline{m}_j)$, $j = 0, 1, \ldots N_r$ are the points on the segment L_r and $\overline{p}_j$ are the corresponding coefficients. Obviously, the function (13) has the same distribution of zeros as (9) except the multiple zero $s_i = 0, i = 1 \ldots \overline{m}_0$. Consider $z = s + \mu_r \ln s$ and $\widetilde{m}_j = \mu_r \widetilde{\vartheta}_j$, from (13), we obtain

$$f_r(z) = \sum_{j=0}^{N_r} \overline{p}_j \mathrm{e}^{\widetilde{\vartheta}_j z} = \sum_{j=0}^{N_r} \overline{p}_j \left(\mathrm{e}^{z/\mu_r}\right)^{\widetilde{m}_j}. \tag{14}$$

Obviously, $f_r(z)$ is a polynomial in $w_r = \mathrm{e}^{z/\mu_r}$ of degree $\widetilde{m}_{N_r}$. To each zero w_{r_k}, $k = 1, \ldots, \widetilde{m}_{N_r}$ of this polynomial, there corresponds a chain of zeros determined by

$$z = \mu_r \ln |w| + \mathrm{j}\mu_r (\arg w + 2k\pi). \tag{15}$$

Taking into consideration that according to Lemma 12.3 of [1], the curve $\mathrm{Re}(z) = \mathrm{Re}(s + \mu_r \ln s) = const.$ is asymptotic to the curve $\beta + \mu_r \ln |\omega| = const.$, the formula (11) results from (15). □

III. Features of the Quasi-Polynomial Maps

In order to determine the positions of the zeros, the contours of $g(s)$ can be mapped instead of $h(s)$. This is convenient, because it allows us to utilize the features of the function $g(s)$ for large magnitudes of s. First of all, for $|s| \geq c$ (c is large) the contours $R(\beta, \omega) = 0$ and $I(\beta, \omega) = 0$, where $R(\beta, \omega) = \mathrm{Re}(g(s))$ and $I(\beta, \omega) = \mathrm{Im}(g(s))$, can intersect each other only in the regions V_r. There are no intersections of the contours in the regions U_r and the following Lemma holds.

Lemma 2: For $|s| \geq c$ (c is large) in the regions U_r, the contours $R(\beta,\omega) = 0$ and $I(\beta,\omega) = 0$, where $R(\beta,\omega) = \mathrm{Re}(g(s))$ and $I(\beta,\omega) = \mathrm{Im}(g(s))$, tend to match the curves described by

$$\widetilde{R}(\beta,\omega) = \cos\left(\omega\overline{\vartheta}_{N_r} + \overline{m}_{N_r}\phi\right) = 0 \tag{16}$$

$$\widetilde{I}(\beta,\omega) = \sin\left(\omega\overline{\vartheta}_{N_r} + \overline{m}_{N_r}\phi\right) = 0 \tag{17}$$

where $\phi = \arg(s)$ and $(\overline{\vartheta}_{N_r}, \overline{m}_{N_r})$ is the rightmost point of the segment L_r.

Proof: Due to a) of Theorem 1 and due to the fact that $\lim_{|s|\to\infty} |\varepsilon_r(s)| = 0$, in the regions U_r, the function (7) can be approximated by the function

$$\begin{aligned}\widetilde{g}_r(s) &= \overline{p}_{N_r} s^{\overline{m}_{N_r}} \mathrm{e}^{s\overline{\vartheta}_{N_r}} = \overline{p}_{N_r} \mathrm{e}^{s\overline{\vartheta}_{N_r} + \overline{m}_{N_r}\ln(s)} \\ &= \overline{p}_{N_r} \mathrm{e}^{\mathrm{Re}(s)\overline{\vartheta}_{N_r} + \overline{m}_{N_r}\ln|s|} \mathrm{e}^{\mathrm{j}\left(\mathrm{Im}(s)\overline{\vartheta}_{N_r} + \overline{m}_{N_r}\arg(s)\right)}.\end{aligned} \tag{18}$$

Thus, the zero-level contours of (18), i.e. the functions $\widetilde{R}(\beta,\omega) = 0$ and $\widetilde{I}(\beta,\omega) = 0$ are determined by the second exponential term in (18), which can be split into the functions (16) and (17) respectively. □

Obviously, for β fixed and $|s|$ large, the shortest distances of points on the neighboring curves (16) and (17) are $\Delta\omega_{\widetilde{R}\widetilde{R}} = \Delta\omega_{\widetilde{I}\widetilde{I}} \approx \pi/\vartheta_r$. Thus, the highest density of the curves is in the region U_M, where $\Delta\omega_{\widetilde{R}\widetilde{R}} = \Delta\omega_{\widetilde{I}\widetilde{I}} \approx \pi/\vartheta_N = \pi/\alpha_0$. Therefore, the value of π/α_0 is an important scale parameter for mapping the zero-level contours in the region $\boldsymbol{D}$ where $|s|$ is large. Choosing Δ_g according to

$$\Delta_g = \frac{1}{N_{gp}}\frac{\pi}{\alpha_0} \tag{19}$$

implies that the mutual distances of the curves $\Delta\omega_{\widetilde{R}\widetilde{R}}$ and $\Delta\omega_{\widetilde{I}\widetilde{I}}$ in the region U_M will be covered by N_{gp} grid points (recommended value of $N_{gp} : 5 < N_{gp} < 10$). However, let us remark that such Δ_g is not proper for mapping the contours in the area of $\boldsymbol{D}$ where $|s|$ is small if the mutual distances of the zeros in this region are smaller than π/α_0. For example, if the delay α_0 is close to zero, e.g. $\alpha_0 = 0.1$ and some zeros are distributed in the circle $|s| = 1$, running QPmR with Δ_g chosen according to (19) would result in omitting some of these zeros. Therefore, after running QPmR, the maps always need to be checked in the region of low magnitudes of s. If the contours are not mapped smoothly, Δ_g should be reduced and QPmR should be run again over this region.

IV. Locating Areas Free of Zeros

Due to the root distribution properties addressed above, large areas of $\boldsymbol{D}$ are free of zeros. Omitting these areas in the scanning procedure of advanced QPmR (aQPmR) will decrease considerably the computational power needed in the algorithm. The procedure for selecting regions free of zeros in $\boldsymbol{D}$ is described in the following algorithm:

Algorithm 1 Defining the regions free of zeros.

Let the region which is to be scanned be given by $\boldsymbol{D} = [\beta_{\min}\beta_{\max}] \times [0\ \omega_{\max}]$, where $\beta_{\max}$ is the real upper bound of the spectrum and $\pi/\alpha_0 \ll \omega_{\max}$.

1) Determine the asymptotic exponentials S_p given by (11) as described in Lemma 1 and order them so that $\mu_p \geq \mu_{p+1}$, $p = 1, 2, \ldots, p_{\max}$
2) Determine δ_S, such that $\pi/\alpha_0 < \delta_S$, which determines the width of the strips where the chains of zeros are to be located.

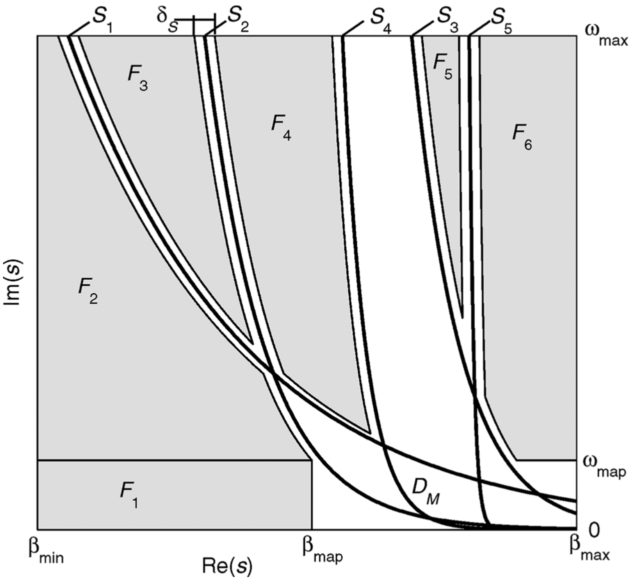

Fig. 1. Illustration of the regions F_q, determined in Algorithm 1, which are likely to be free of zeros.

3) For each S_p, within $\boldsymbol{D}$, define a strip D_p of width δ_S with centre line S_p and left/right boundaries S_{L_p} and S_{R_p} respectively, parallel with S_p.
4) Determine ω_{map} such that $\pi/\alpha_0 \ll \omega_{\mathrm{map}} < \omega_{\max}$, which bounds from above the region of small $|s|$ and where it is likely that the asymptotic features of the chains of zeros are not apparent.
5) Define the region $D_M = [\beta_{\mathrm{map}}\beta_{\max}] \times [0\ \omega_{\mathrm{map}}]$ which is to be mapped whole, where β_{map} is given as the intersection point of $\omega = \omega_{\mathrm{map}}$ and the leftmost S_p at this level.
6) Let the regions F_q suspected to be free of zeros be defined as the areas of $\boldsymbol{D}$ not covered by any D_p or D_M.
7) Discard all regions F_q which:
    i) do not have a common boundary with $\boldsymbol{D}$
    ii) have boundaries S_j and S_k for which $\beta_{S_j} < \beta_{S_k}$ and $\mu_j < \mu_k$, where β_{S_j} and β_{S_k} are the intersection points of S_j, and S_k, respectively, with the upper boundary $\omega = \omega_{\max}$

    (in the discarded regions, the zeros are not likely to match the asymptotic exponentials since some of their intersection points lie above the regions, i.e., the corresponding chains are not likely to be separated yet in the regions)
8) Over the boundary of each suspect region F_q apply the argument principle [1]

$$N_{F_q} = \frac{1}{2\pi}\Delta \arg_{\varphi_q} h(s) = \frac{1}{2\pi\mathrm{j}}\int_{\varphi_q} \frac{h'(s)}{h(s)} ds \tag{20}$$

    where φ_q is a boundary of the particular region F_q, and N_{F_q} is the number of zeros located in the region F_q.
    i) If $N_{F_q} = 0$ for $q = 1, 2, \ldots, q_{\max}$, either accept the defined boundaries of F_q or decrease δ_S and ω_{map} and return to step 2.
    ii) If $N_{F_q} > 0$ for any $q = 1, 2, \ldots, q_{\max}$, either discard the region or increase δ_S or ω_{map} and return to step 2.

An illustration example resulting from this procedure is shown in Fig. 1.

Remark 2: For the stability and dynamics analysis of time delay systems with transfer function (2), it is necessary to determine the rightmost roots. Therefore, for this purpose, the right boundary $\beta_{\max}$ of $\boldsymbol{D}$ has to be selected as the upper bound of the spectrum of (1). The problem of determining the positive upper bound of the spectrum has been addressed in Proposition 1.10 of [17]. Let us also remark that the presence of no roots with real parts larger than $\beta_{\max}$ can simply be checked by applying the argument principle. This provides the condition

$$\Delta \arg h(\beta_{\max} + \mathrm{j}\omega) = n\frac{\pi}{2}, \quad \omega \in [0, \infty)$$

which is an analogous condition to the Mikhaylov criterion.

Remark 3: Formula (20) can be evaluated using a method of numerical integration, such as in [10], [13]. An alternative to this approach, used in aQPmR, is a geometric evaluation of $\Delta \arg_{\varphi} h(s)$ [18]. Evaluating the argument on the boundary of φ using the formula $\arg h(s) = \tan^{-1}(\mathrm{Im}h(s)/\mathrm{Re}h(s))$ and taking into account that the result of the inverse tangent is limited on $[-\pi/2, \pi/2]$, $\Delta \arg_{\varphi} h(s)$ is relevant to the number of the sign changes of $\arg h(s)$ along φ.

V. Algorithm Summary

Based on the results presented above, the algorithm aQPmR can be summarized as follows.

Algorithm 2 Summary of the advanced quasi-polynomial mapping based root-finder (aQPmR).

Let the characteristic function $h(s)$ given in (1) be transformed to $g(s)$, given in (7) and let $R(\beta, \omega) = \mathrm{Re}(g(\beta + \mathrm{j}\omega))$ and $I(\beta, \omega) = \mathrm{Im}(g(\beta + \mathrm{j}\omega))$

1) According to Proposition 1, construct the spectrum distribution diagram for function (7) and determine the asymptotic exponentials as described in Lemma 1
2) Locate the areas free of zeros F_q as described in Algorithm 1
3) Map the contours $R(\beta, \omega) = 0$ and $I(\beta, \omega) = 0$ in $\boldsymbol{D}$, except areas F_q, and locate the intersection points $s_{i,0}$ of the contours, determining approximate positions of the zeros.
4) In order to increase accuracy, for each zero $s_{i,0}$ obtained, use the Newton's iteration method

$$s_{i,k+1} = s_{i,k} - h(s_{i,k})/h'(s_{i,k}). \tag{21}$$

which stops when the required accuracy ε_{N} is achieved.

Remark 4: Let us mention that the convergence features of the Newton's method has to be taken into consideration [19]. First of all the initial positions of the zeros need to be close to their exact positions, which is achieved by using a sufficiently small grid step Δ_g. Difficulties may arise when there are multiple zeros in the spectrum or zeros located close to each other. In such cases the result of the Newton's iteration need to be compared with the results of the quasi-polynomial maps. Notice that the multiplicity of the zero is given by the number of the curves $R(\beta, \omega) = 0$ and $I(\beta, \omega) = 0$ intersecting each other in the zero location [15].

Remark 5: The aQPmR given in Algorithm 2 has been implemented in Matlab.[1] In the implementation, the function *contour* is used for mapping the contours $R(\beta, \omega) = 0$ and $I(\beta, \omega) = 0$. The mesh grid Γ is defined as in (6) but the functions $R(\beta, \omega)$ and $I(\beta, \omega)$ are evaluated only at the points outside the suspect regions F_q. The remaining points of Γ are set to zero.

TABLE I
Polynomials and Delays of the Quasi-Polynomial (1)

j	α_j	$p_j(s)$
0	24.99	51.7
1	23.35	$0.03s^3 + 0.04s^2 - 0.1s + 1.5$
2	19.9	$0.5s^3$
3	18.52	$0.15s^5 + 0.2s^4 - 0.9s^3 + 25.2s$
4	13.52	$0.8s^6 + 0.1s^4 - 1.4s + 7.2$
5	10.33	$-8.7s^4 + 2.1s^2 + 19.3s$
6	8.52	$s^7 - 1.1s^5 + 6.7s$
7	4.61	29.1
8	0	$0.2s^8 + 1.7s^7 - 12.8s^5 + 0.01s^2 - 1.8s$

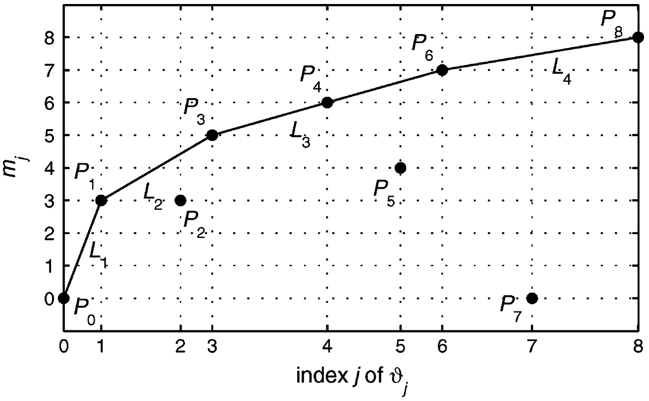

Fig. 2. Spectrum distribution diagram of the quasi-polynomial given by (1) and Table I.

VI. Application Example

The aQPmR algorithm is applied to quasi-polynomial (1), with details given in Table I. The region $\boldsymbol{D} = [\beta_{\min}\beta_{\max}] \times [0\ \omega_{\max}]$ with $\beta_{\min} = -4.5$, $\beta_{\max} = 3\ \omega_{\max} = 100$ is selected. In the first step, the distribution diagram is defined as shown in Fig. 2. The slopes of the diagram segments L_r, $r = 1, 2 \ldots 4$, polynomials of the form (12) and their zeros are given in Table II, determining five asymptotic exponentials given by (11). In step 2, according to the scale factor $\pi/\alpha_0 = 0.126$, the parameters $\Delta_g = 0.0157 (N_g = 8)$, $\delta_s = 0.38$ and $\omega_{\mathrm{map}} = 10$ (which implies $\beta_{\mathrm{map}} = -1.4$) are selected. The suspect regions free of zeros are defined and checked as described in Algorithm 1. In the third step, the contours $R(\beta, \omega) = 0$ and $I(\beta, \omega) = 0$ are mapped, see Fig. 3, and their intersection points $s_{i,0}$ are located. In the last step, Newton's method is used to increase the accuracy of each zero to $\varepsilon_{\mathrm{N}} = 1e - 6$. The final results of the aQPmR algorithm are shown in Fig. 4.

In order to evaluate the impact of omitting empty regions on the computation time, the algorithm aQPmR is compared with QPmR in Table III. As can be seen, larger discarded regions resulted in faster computation times as compared to the standard QPmR. However, the reduction of the computation time is less than the ratio between the mapped areas in both algorithms ρ_D. This is due to fact that in QPmR Matlab implementation the function is evaluated over the mesh grid Γ using only a few matrix based evaluations, whereas in aQPmR a large number of such evaluations are done over the area strips. In Table III the computation time of DDE-Biftool is also shown, which was used to compute the roots of system (3) associated to (1) determined by Table I. As can be seen, DDE-Biftool is more efficient if only the stability determining right-most zeros are to be computed (notice that this is the

[1] Matlab implementations of both QPMR and aQPmR are available at http://www.cak.fs.cvut.cz/QPmR

TABLE II
SLOPES OF THE DIAGRAM SEGMENTS L_r OF THE QUASI-POLYNOMIAL GIVEN BY (1) AND TABLE I, CORRESPONDING POLYNOMIALS OF THE FORM (12) AND THEIR ZEROS

	L_1	L_2	L_3	L_4
μ_r	1.8293	0.4141	0.2	0.1174
$f_r(w)$	$0.03w^3 + 51.7$	$0.15w^2 + 0.03$	$w^2 + 0.8w + 0.15$	$0.2w + 1$
w_1	-11.989	0.447j	-0.5	-5
w_2	5.995+10.383j	- 0.447j	-0.3	
w_3	5.995-10.383j			
$\|w_k\|$	11.989	0.447	0.5, 0.3	5

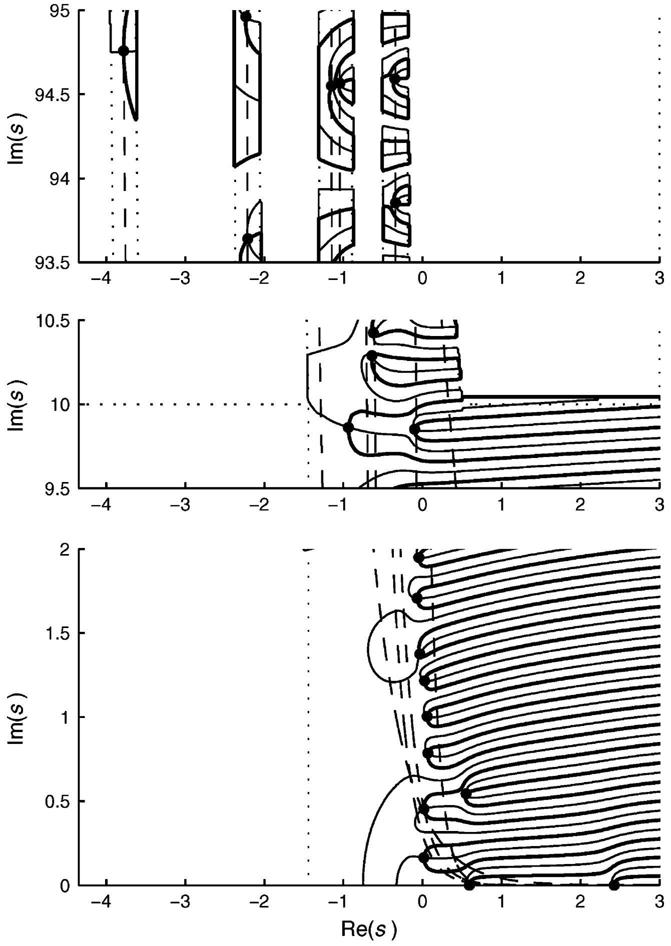

Fig. 3 Contours $R(\beta, \omega) = 0$ (thick line) and $I(\beta, \omega) = 0$ (thin line) of the quasi-polynomial given by (1) and Table I mapped by aQPmR over the regions defined in Algorithm 1. Black circles—zeros, dashed lines—asymptotic exponentials, dotted lines—boundaries of the zero free regions.

primary purpose of DDE-Biftool), while aQPmR is more efficient for scanning large areas for the quasi-polynomial zeros.

VII. CONCLUSION

The advanced quasi-polynomial mapping based root-finder (aQPmR) presented in the technical note is an effective tool for computing the spectrum of quasi-polynomial (1) over large regions of the complex plane. The efficiency of the root-finder is increased by eliminating large areas free of zeros from the scanning procedure. The areas are located between the asymptotic exponentials of the chains of zeros, resulting from the distribution diagram of the quasi-polynomial. Furthermore, features of the quasi-polynomial maps are studied and the problem of proper definition of the mesh grid for the mapping is discussed. An application example is presented, and the efficiency of the algorithm is compared with both an older version of the algorithm and DDE-Biftool.

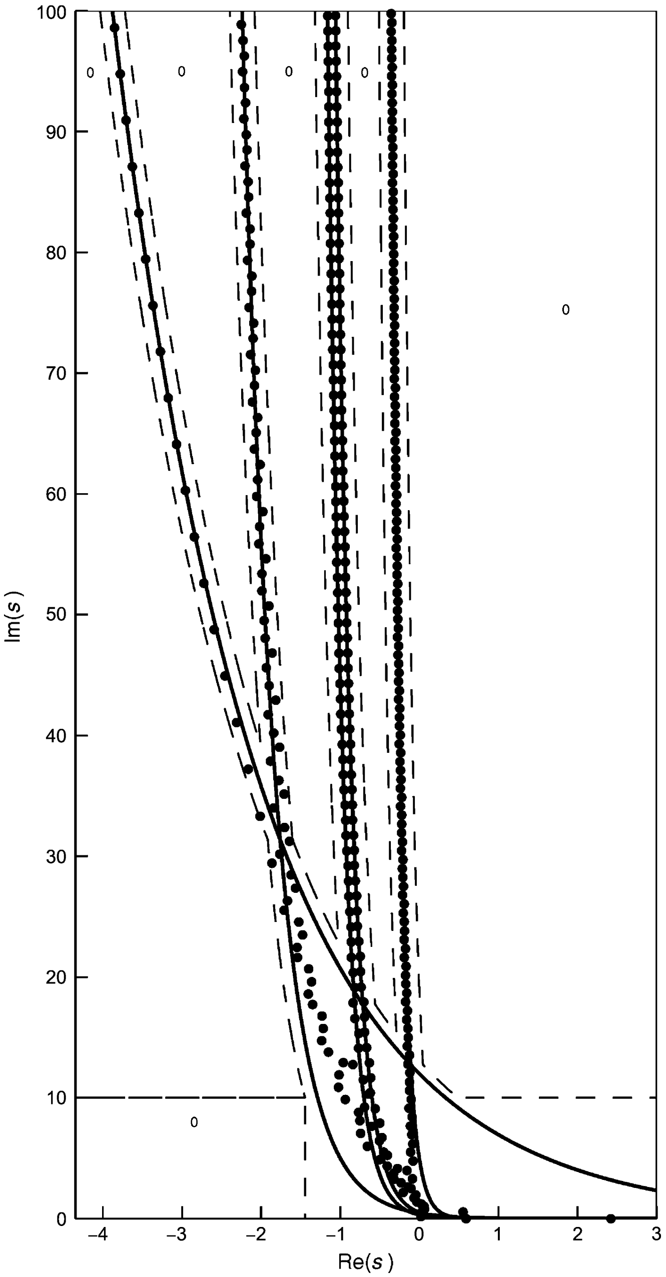

Fig. 4 Spectrum of quasi-polynomial given by (1) and Table I., computed by aQPmR, black circles—zeros, solid lines—asymptotic exponentials, dashed lines—boundaries of the zero free regions.

REFERENCES

[1] R. Bellman and K. L. Cooke, *Differential-Difference Equation*. New York: Academic Press, 1963.

[2] P. Zítek and T. Vyhlídal, "A residue based evaluation of pole significance in time delay systems," in *Proc 5th IFAC Workshop, Time Delay Syst. (TDS'04)*, Leuven, Belgium, 2004, pp. 29–34.

TABLE III
COMPARISON OF THE ADVANCED aQPmR WITH THE STANDARD QPmR AND DDE-BIFTOOL USED TO COMPUTE THE ZEROS OF THE QUASI-POLYNOMIAL GIVEN BY (1) AND TABLE I

$\omega_{\max}$	$\beta_{\min}$	N_D	QPmR	aQPmR		ρ_D	DDE-Biftool	
			T_C	T_C	T_P		T_C	Δt
10	-1.5	43	3.2				1.15	0.15
20	-2.1	82	7.2	11.7	3.9	1.8	7.1	0.075
40	-2.8	161	15.6	17.2	4.3	2.6	48.6	0.038
100	-4.5	401	47.7	33.2	6.3	4.0		
200	-5.7	797	125	61.7	9.8	5		
300	-6.5	1196	334	97	12	5.6		

T_C [sec.]—overall time of computation (Matlab 7.4, PC—Core Duo, 1.66 GHz, 1 GB RAM), T_P [sec.]—time of the preparatory stage of aQPmR (defining the suspect regions free of zeros, applying the argument principle). ρ_D—ratio between the mapped areas of QPmR and aQPmR. N_D—number of zeros in $\boldsymbol{D}$. Parameters of aQPmR (QPmR): $\boldsymbol{D} = [\beta_{\min}\beta_{\max}] \times [0\ \omega_{\max}]$, $\beta_{\max} = 3, \omega_{\text{map}} = 10(\beta_{\text{map}} = -1.4), \Delta_g = 0.0157(N_g = 8)$, $\delta_s = 0.38$, accuracy $\varepsilon_{\text{N}} = 1\text{e} - 6$. Parameters of DDE-Biftool: Milne-Simpson 4th order method, accuracy 1e-6, Δt—discretization step.

[3] K. Engelborghs and D. Roose, "Numerical computation of stability and detection of Hopf bifurcation of steady state solutions of delay differential equations," *Adv. Comput. Math.*, vol. 10, no. 3–4, pp. 271–289, 1999.

[4] K. Verheyden, T. Luzyanina, and D. Roose, "Efficient computation of characteristic roots of delay differential equations using LMS methods," *J. Comput. Appl. Math.*, vol. 214, no. 1, pp. 209–226, 2008.

[5] D. Breda, "Solution operator approximation for characteristic roots of delay differential equations," *Appl. Numer. Math.*, vol. 56, pp. 305–331, 2006.

[6] D. Breda, S. Maset, and R. Vermiglio, "Computing the characteristic roots for delay differential equations," *IMA J. Numer. Anal.*, vol. 24, no. 1, pp. 1–19, 2004.

[7] D. Breda, S. Maset, and R. Vermiglio, "Pseudospectral differencing methods for characteristic roots of delay differential equations," *SIAM J. Sci. Comput.*, vol. 27, no. 2, pp. 482–495, 2005.

[8] K. Engelborghs, T. Luzyanina, and D. Roose, "Numerical bifurcation analysis of delay differential equations using DDE-Biftool," *ACM Trans. Math. Software*, vol. 28, no. 1, pp. 1–21, 2002.

[9] D. Breda, S. Maset, and R. Vermiglio, "TRACE-DDE: A tool for robust analysis and characteristic equations for delay differential equations," in *Topics in Time-Delay Systems: Analysis, Algorithms and Control*, ser. LNCIS, J. J. Loiseau, W. Michiels, S. I. Niculescu, and R. Sipahi, Eds. Berlin, Germany: Springer, 2009.

[10] P. Kravanja and M. Van Barel, *Computing the Zeros of Analytic Functions*. Berlin, Germany: Springer, 2000, vol. 1727.

[11] A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*. New York: McGraw-Hill, 1970.

[12] M. N. Vrahatis, "An efficient method for locating and computing periodic orbits of nonlinear mappings," *J. Comput. Phys.*, vol. 119, pp. 105–119, 1995.

[13] M. Dellnitz, O. Schütze, and Q. Zheng, "Locating all the zeros of an analytic function in one complex variable," *J. Comput. Appl. Math.*, vol. 138, pp. 325–333, 2002.

[14] L. M. Delves and J. N. Lyness, "A numerical method for locating the zeros of analytic functions," *Math. Comput.*, vol. 21, pp. 543–560, 1967.

[15] T. Vyhlídal and P. Zítek, "Quasipolynomial mapping based rootfinder for analysis of time delay systems," in *Proc 4th IFAC Workshop, Time Delay Syst. (TDS'03)*, Rocquencourt, France, 2003, pp. 227–232.

[16] A. D. Gouveia, "An aid to two-dimensional contouring using nonuniform orthogonal grids—A fortran algorithm," *Comput. Geosci.*, vol. 19, no. 8, pp. 1071–1076, 1992.

[17] W. Michiels and S. Niculescu, *Stability and Stabilization of Time-Delay Systems. An Eigenvalue Based Approach*, ser. Advances in Design and Control 12. Philadelphia, PA: SIAM Publications, 2007.

[18] T. Luzyanina and D. Roose, "Numerical stability analysis and computation of Hopf bifurcation points for delay differential equations," *J. Comput. Appl. Math.*, vol. 72, pp. 379–392, 1996.

[19] P. Deuflhard, *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Berlin, Germany: Springer, 2004, vol. 35.

A Simplified Design for Strict Lyapunov Functions Under Matrosov Conditions

Frederic Mazenc, Michael Malisoff, and Olivier Bernard

***Abstract*—We construct strict Lyapunov functions for broad classes of nonlinear systems satisfying Matrosov type conditions. Our new constructions are simpler than the designs available in the literature. We illustrate the practical interest of our designs using a globally asymptotically stable biological model.**

***Index Terms*—Biological systems, Lyapunov function, Matrosov theorem.**

I. INTRODUCTION

Lyapunov functions play an essential role in modern nonlinear systems analysis and controller design. Oftentimes, *non*-strict Lyapunov functions are readily available. However, *strict* (i.e., *strong*) Lyapunov functions are preferable since they can be used to quantify the effects of disturbances; see the precise definitions below. Strict Lyapunov functions have been used in several biological contexts e.g. to quantify the effects of actuator noise and other uncertainty on the steady state concentrations of competing species in chemostats [13], but their explicit construction can be challenging. For some large classes of systems, there are mechanisms for transforming non-strict Lyapunov functions into the required strict Lyapunov functions e.g. [4], [11], [12], [14], [15].

For systems satisfying conditions of Matrosov's type [8], [10], strict Lyapunov functions were constructed in [15], under very general conditions. However, the generality of the assumptions in [15] makes its constructions complicated and therefore difficult to apply. Moreover, the Lyapunov functions provided by [15] are not locally bounded from below by positive definite quadratic functions, even for asymptotically stable linear systems, which admit a quadratic strict Lyapunov function. The shape of Lyapunov functions, their local properties and their simplicity matter when they are used to investigate robustness and construct feedbacks and gains.

In the present work, we revisit the problem of constructing strict Lyapunov functions under Matrosov's conditions. Our results have the

Manuscript received September 27, 2007; revised May 05, 2008 and August 07, 2008. Current version published January 14, 2009. This work was supported in part by NSF/DMS Grants 0424011 and 0708084. Recommended by Associate Editor I. Kolmanovsky.

F. Mazenc is with Projet MERE INRIA-INRA, UMR Analyse des Syst̀mes et Biométrie, Montpellier 34060, France (e-mail: mazenc@supagro.inra.fr).

M. Malisoff is with the Department of Mathematics, Louisiana State University, Baton Rouge, LA 70803-4918 USA (e-mail: malisoff@lsu.edu).

O. Bernard is with Projet COMORE INRIA Sophia-Antipolis, Sophia-Antipolis 06902, France (e-mail: olivier.bernard@inria.fr).

Color versions of one or more of the figures in this technical note are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2008.2008353