

Stable Approximation of Unstable Transfer Function Models

László Balogh and Rik Pintelon, *Fellow, IEEE*

Abstract—The result of a system identification experiment is usually a parametric continuous-time (s-domain) or discrete-time (z-domain) model. Due to noise on the measurements and/or nonlinear distortions, this model can be unstable. If an additional delay is added to the unstable system, then experience shows that a stable approximation with small approximation error can be obtained. In this paper, a new numerical algorithm is proposed for finding a delay that gives a stable result. Contrary to classical approaches, it needs fewer gradientlike steps during the approximation process.

Index Terms—Delay, differential equations, parameter estimation, stable approximation, system identification.

I. INTRODUCTION

IN A NUMBER of engineering problems, such as modeling, simulation, and filter design, one needs a stable transfer function approximation of a complex system. However, due to disturbing noise and/or nonlinear distortions, the identified transfer function model may be unstable. In addition, in the filter design, the transfer function that best matches some user-defined amplitude and phase specifications is usually unstable.

The transfer function is usually obtained by minimizing some distance (norm) between the target function (frequency response function or user-defined amplitude and phase specifications) and the transfer function model. According to the particular norm used, one gets a minimax (weighted), least squares, or least absolute values approximation.

There are several ways to obtain a stable approximation. One possible solution is to search for the appropriate model in a restricted subset of the whole parameter space. This leads to a suboptimal but stable approximation. The main problem of this kind of method is that the solution found by adapted algorithms is local. An example is a modified gradient algorithm in which the step size is halved every time the model obtained is unstable. Intuitively, if the initial stable value of the gradient method is not close to the global optimum, then it is impossible to reach that with guaranteed stable steps [11]. In many cases, a so-called penalty function is used. This function is close to zero if the point from the model space determines a stable model,

and it is very large if the corresponding model is unstable. The modified cost function is the original function plus the penalty function. The main advantage of this approach is that the same numerical tools can be used without the penalty function. However, it remains a local method.

Another approach consists of stabilizing the unstable poles by reflecting them with respect to (w.r.t.) the stability border and, next, adding an all-pass section. For example, if, in filter design, the resulting infinite impulse response (IIR) filter is unstable, then the unstable poles are stabilized by reflecting into the unit circle giving $H_{\text{stab}}(z)$. This operation does not change the amplitude characteristic but does modify the phase. Therefore, an additional all-pass filter $H_{\text{all}}(z)$ is designed to compensate the phase. The disadvantage of the method is that the resulting filters $H_{\text{stab}}(z)H_{\text{all}}(z)$ are not optimal with regard to their orders [13]. It is shown in [13] that the same accurate approximation can be obtained with lower filter orders by approximating the delayed target in one step. Stability of the final filter is obtained by an appropriate choice of the delay.

In [10], stable approximations are obtained without adding a well-chosen delay to the target function. However, the approximation error in the frequency band of interest converges to zero at the price of an unbounded error outside this band.

This paper handles the method where a delay is added to the target function. Practical examples show that, if enough delay is introduced in the target function, then the poles of the approximator can be stabilized. Unfortunately, this idea implies some problems, which have to be managed in a practical case. The most important question is how the optimal delay value can be determined. In this case, optimal means that the global minimum of the cost function with fixed delay is the smallest and determines a stable model. In this paper, a new method based on ordinary differential equations (ODEs) is introduced for automatically finding a stable (delayed) approximation. The main benefit, in contrast with the full search, is that this algorithm needs fewer gradientlike steps.

II. PRELIMINARIES AND PROBLEM DESCRIPTION

The identification process can be divided into two steps. In the first step, a parametric model is estimated from noisy data. The result is a validated model provided with uncertainty bounds. In the second step, the unstable model is stabilized by adding a well-chosen delay to the target system. The result is a stable model provided with bias bounds. The end result of the two-step procedure is a stable model with uncertainty and bias bounds. This is in contrast with the classical one-step procedures that can provide neither bias bounds nor accurate

Manuscript received July 15, 2006; revised April 27, 2008. First published June 13, 2008; current version published November 12, 2008. An earlier version of this paper was presented at the IEEE Instrumentation and Measurement Technology Conference 2006, Sorrento, Italy, April 24–27, 2006.

L. Balogh is with the Department of Measurement and Instrument Engineering, Budapest University of Technology and Economics, 1521 Budapest, Hungary (e-mail: lbalogh@mit.bme.hu).

R. Pintelon is with the Department of Fundamental Electricity and Instrumentation (ELEC), Vrije Universiteit Brussel, 1050 Brussels, Belgium (e-mail: Rik.Pintelon@vub.ac.be).

Digital Object Identifier 10.1109/TIM.2008.926050

uncertainty bounds [9]. The second step is the topic of this paper. Stabilization means that we want to approximate the model by adding a delay to the transfer function. Without adding a delay, it is very hard to stabilize poles in the frequency band of interest (see, for example, [8] and [10]).

The result of the identification process is a rational transfer function model $T(\Omega)$ given by

$$T(\Omega) = \frac{\delta_0 + \delta_1\Omega + \dots + \delta_{n_\delta}\Omega^{n_\delta}}{\gamma_0 + \gamma_1\Omega + \dots + \gamma_{n_\gamma}\Omega^{n_\gamma}} \quad (1)$$

where Ω denotes the corresponding (angular) frequency variable ($e^{j2\pi f}$ for the z-domain and $j\omega$ for the s-domain); n_δ and n_γ are the orders of the numerator and denominator, respectively; and δ_l and γ_k are the coefficients. If this is stable, then the system identification process ends. However, if T is unstable, then further processing is needed to obtain a stable model. In open-loop applications, adding a delay to target function T is allowed. Practical experience shows that approximating this system in the least squares sense gives stable results. The approximation problem consists of minimizing the following cost function w.r.t. P :

$$C(P, \tau) = \int_I W(\Omega) |T(\Omega)e^{-j\omega\tau} - H(\Omega, P)|^2 d\Omega \quad (2)$$

where H is the approximator, W is a weighting function, P is the parameter vector (the coefficients of the numerator and denominator polynomials of H), I is the frequency interval of the system, τ is the additional delay, and

$$H(\Omega, P) = \frac{\beta_0 + \beta_1\Omega + \dots + \beta_{n_\beta}\Omega^{n_\beta}}{\alpha_0 + \alpha_1\Omega + \dots + \alpha_{n_\alpha}\Omega^{n_\alpha}} \quad (3)$$

$$P = [\beta_0, \dots, \beta_{n_\beta}, \alpha_0, \dots, \alpha_{n_\alpha}] \quad (4)$$

where n_α and n_β are the orders of the denominator and numerator, respectively. The dimension of parameter vector P is $n_\beta + n_\alpha + 1$, because one of the parameters is fixed to obtain a unique solution. The conjecture is that there always exists a delay τ such that the global minimum of the cost function (for a fixed τ) results in a stable model.

In the case of a continuous frequency grid ($I = [0, 2\pi]$ for the z-domain and $I = (-\infty, +\infty)$ for the s-domain), the integral in (2) remains unchanged, but in the case of a finite frequency grid, it is replaced by a sum, i.e.,

$$C(P, \tau) = \sum_{k=1}^F |W(\Omega_k)(T(\Omega_k)e^{-j\omega_k\tau} - H(\Omega_k, P))|^2 \quad (5)$$

where Ω_k and ω_k denote the frequency points, and F denotes the number of elements in the frequency grid. Target function T need not be given as in (1). In (5), only F complex numbers must be given. An advantage of the analytical expression is that frequency point selection from the band of interest can be done with fewer restrictions, which can be important because the delay that determines a minimum of the cost function, which is a stable model, depends on the frequency grid.

III. PROPOSED PRACTICAL ALGORITHM

Equation (5) is continuous w.r.t. τ . The characterization of the extreme values of the cost function can be done by evaluating the following equation:

$$\frac{\partial C(P, \tau)}{\partial P} = 0. \quad (6)$$

This is a necessary but insufficient condition for the global minimum of the cost function. As we can see, this equation defines an implicit function $P(\tau)$, which is continuous w.r.t. the delay, which means that we can write a differential equation for the local minima. The following formula comes from the total differentiation of (6):

$$\frac{\partial^2 C(P, \tau)}{\partial P^2} \frac{dP}{d\tau} + \frac{\partial^2 C(P, \tau)}{\partial P \partial \tau} = 0. \quad (7)$$

Hence

$$\frac{dP}{d\tau} = - \left[\frac{\partial^2 C(P, \tau)}{\partial P^2} \right]^{-1} \frac{\partial^2 C(P, \tau)}{\partial P \partial \tau}. \quad (8)$$

This is a set of ODEs; hence, numerical tools from the numerical differential equations theory can be used [4]. In this paper and in our software, we use, for instance, the following methods [4]:

- the classical Euler method (see Appendix A);
- the Runge–Kutta method (see Appendix B).

The main difference with the classical ODE solvers is that, in the proposed method, a gradient-type algorithm can be rerun at every delay value; hence, the optimal parameter vector can be determined. The trigger and frequency of running the gradient-type method are the control parameters of the algorithm. For example, it is possible to run, in every k th integration step of (8), the gradient-type algorithm, where k is given by the user. Another possibility consists of running the gradient method when the variation of the cost function exceeds the user-defined threshold. The continuity of parameter vector $P(\tau)$ implies that an ODE step can provide a good initial value for the gradient method.

It is important to note that the solution of (6) gives a local extreme value of the cost function for every τ . At least one of these curves is the global optimum P_* at a value of the delay. Since the cost function is a complicated function, $P_*(\tau)$ does not coincide for all real values τ with any solution of (6). Hence, $P_*(\tau)$ is not a continuous function of delay τ . However, $C(P_*(\tau), \tau)$ is continuous w.r.t. delay τ .

The benefits of this algorithm are given here.

- The algorithm is able to find a stable approximation that is the global optimum of the cost function with fixed delay. In the method in which the delay is one of parameters to be optimized, it cannot be guaranteed that the model found is stable. One can obtain a stable approximation, but an appropriate initial value of the delay is necessary. However, finding such an initial delay is not an easy task.
- In Section III-A, it can be seen that the computation loads of a gradient-type step and an ODE step are the same. However, estimating the parameter vector with the ODE

solver needs only one step, whereas solving the minimization problem with a gradient method needs several iteration steps, which means that the required number of steps can be reduced by using the ODE steps.

A. Numerical Efficient Computation Written as an ODE Step

Equation (5) can be written as

$$C(P, \tau) = (y(\tau) - g(P))^T (y(\tau) - g(P)) \quad (9)$$

where $y(\tau) : \mathbb{R} \mapsto \mathbb{R}^{2F}$, and $g(P) : \mathbb{R}^{n_\beta + n_\alpha + 1} \mapsto \mathbb{R}^{2F}$. The definition of $y(\tau)$ is

$$y(\tau) = \begin{bmatrix} \text{Re} \{y_C(\tau)\} \\ \text{Im} \{y_C(\tau)\} \end{bmatrix}$$

where $y_C(\tau) \in \mathbb{C}^F$ denotes the following complex delayed target function:

$$y_C(\tau) = [y_C(\tau, \Omega_1), \dots, y_C(\tau, \Omega_F)]^T$$

where $y_C(\tau, \Omega_k)$, $k = 1, \dots, F$, as given in the following, are the elements of vector $y_C(\tau)$:

$$y_C(\tau, \Omega_k) = W(\Omega_k)T(\Omega_k)e^{-j\omega_k\tau}.$$

The real $g(P)$ is constructed by the usual complex-to-real transformation, i.e.,

$$g(P) = \begin{bmatrix} \text{Re} \{g_C(P)\} \\ \text{Im} \{g_C(P)\} \end{bmatrix}.$$

$g_C(P)$ is defined as

$$g_C(P) = [g_C(P, \Omega_1), \dots, g_C(P, \Omega_F)]^T$$

where

$$g_C(P, \Omega_k) = W(\Omega_k)H(\Omega_k, P).$$

In this expression, the usual map between the complex plane and the 2-D real space is used. Differentiation of (9) w.r.t. P gives

$$\left(\frac{\partial C(P, \tau)}{\partial P}\right)^T = -2 \left(\frac{\partial g(P)}{\partial P}\right)^T (y(\tau) - g(P)) \quad (10)$$

where $(\partial g(P)/\partial P) \in \mathbb{R}^{2F \times (n_\beta + n_\alpha + 1)}$. Then

$$\begin{aligned} \frac{\partial^2 C(P, \tau)}{\partial P^2} = & -2 \sum_{k=1}^{2F} \left(\left(\frac{\partial^2 g_{[k]}(P)}{\partial P^2} \right) (y(\tau) - g(P))_{[k]} \right) \\ & + 2 \left(\frac{\partial g(P)}{\partial P} \right)^T \left(\frac{\partial g(P)}{\partial P} \right) \end{aligned} \quad (11)$$

where subscript $[k]$ denotes the k th element of a vector, and $(\partial^2 g(P)/\partial P^2) \in \mathbb{R}^{(n_\beta + n_\alpha + 1) \times (n_\beta + n_\alpha + 1)}$. If P is close to an extreme value of the cost function, then $y(\tau) \approx g(P)$, and

the sum containing the terms of the second derivative can be neglected, i.e.,

$$\frac{\partial^2 C(P, \tau)}{\partial P^2} \approx 2 \left(\frac{\partial g(P)}{\partial P} \right)^T \left(\frac{\partial g(P)}{\partial P} \right). \quad (12)$$

Since

$$\left(\frac{\partial^2 C(P, \tau)}{\partial P \partial \tau} \right)^T = -2 \left(\frac{\partial g(P)}{\partial P} \right)^T \left(\frac{\partial y(\tau)}{\partial \tau} \right) \quad (13)$$

and using (8) and (12), we obtain

$$J^T J \frac{\partial P}{\partial \tau} = J^T \left(\frac{\partial y(\tau)}{\partial \tau} \right) \quad (14)$$

where J is the Jacobian matrix, i.e., $J = (\partial g(P)/\partial P)$. Explicit expressions are given in Appendices C and D.

The last equation can be solved in a numerical stable way using the singular value decomposition, i.e.,

$$J = USV^T$$

where $U \in \mathbb{R}^{2F \times (n_\alpha + n_\beta + 1)}$ and $V \in \mathbb{R}^{(n_\alpha + n_\beta + 1) \times (n_\alpha + n_\beta + 1)}$ are unitary matrices, and $S \in \mathbb{R}^{(n_\alpha + n_\beta + 1) \times (n_\alpha + n_\beta + 1)}$ is a diagonal matrix that contains the singular values. Hence, the solutions can be calculated as

$$\frac{\partial P}{\partial \tau} = -VS^\dagger UJ^T \left(\frac{\partial y(\tau)}{\partial \tau} \right)$$

where S^\dagger is a matrix in which the nonzero diagonal elements of S are inverted.

B. Proposed Algorithm

The automatic delay selection algorithm can be split into two main parts. The first part is responsible for finding a delay value such that the minimum of the cost function (5) is a stable model. Using this result as a starting point, the second main part of the algorithm improves the parameter vector to find the best model. The two parts can independently be treated.

Automatic delay selection is not an easy task. In this paper, a possible solution is presented. Two kinds of steps are distinguished. A step is equivalent with updating parameter vector P . The first group includes the steps that are computed at a fixed delay value. These steps are called gradient steps, because, in the algorithm, every step at a fixed delay is a minimization of the cost function. The second group includes the steps that also change the delay. These steps are called ODE steps, because the new parameter vector is computed by using the ODE solving algorithms. The overall algorithm is a sequence of different steps.

- 1) Start.
- 2) $\tau = 0$.
- 3) Calculate an initial value of parameter vector $P(\tau)$.
- 4) Decide if it is an ODE step or gradient (for example, every k th step and/or if the variation of the cost function exceeds a threshold).

- 5) If it is an ODE step, then go to 8.
- 6) Gradient steps until the minimum is reached.
- 7) Go to 9.
- 8) Make an ODE estimation.
- 9) Meet the stop condition? If no, then go to 4.
- 10) Stop.

After finding a stable model that minimizes the cost function with fixed delay, the improvement part of the algorithm comes.

- 1) Start.
- 2) Calculate initial estimation of $\Delta\tau$.
- 3) Compute $(\partial C(P, \tau)/\partial \tau)$, which determines the direction of the step.
- 4) If $|\Delta\tau|$ meets the stop condition, then end.
- 5) Calculate the ODE step, i.e., determine P_{new} .
- 6) Is it a stable model? If yes, go to 5.
- 7) Do gradient minimization with the fixed delay value. Update P_{new} .
- 8) Is it a stable model? If no, $\Delta\tau = \Delta\tau/2$, and go to 5.
- 9) $\tau = \tau + \Delta\tau$, $\Delta\tau = 1.4\Delta\tau$, and $P = P_{\text{new}}$.
- 10) Go to 3.

In this part of the algorithm, many of the gradient steps are executed. The reason for this is that there is no guarantee that delay changes do not cause a noncontinuous jump in parameter vector $P_*(\tau)$.

IV. DESIGN OF INTEGRATORS, DIFFERENTIATORS, AND HILBERT TRANSFORMERS

To demonstrate, the algorithm is applied to already published examples. In [6], a design of the digital integrator is shown. The aim is to approximate the characteristic of an ideal integrator in a restricted frequency band. The ideal integrator transfer function is

$$T(j2\pi f) = \frac{1}{j2\pi f}. \quad (15)$$

In [6], it was found that this integrator can be well approximated using a model of $n_\beta = 6/n_\alpha = 6$ order and $\tau = 4.47796$ in the frequency band $[0, 0.25]$. The sampling frequency is equal to 1. The relative complex error of the realized transfer function H is defined as

$$\delta = \left| \frac{H - T}{T} \right| \quad (16)$$

where T is the target function from (15).

To avoid a singularity at $f = 0$, the target T in (15) is replaced by

$$\tilde{T} = \frac{1 - e^{j2\pi f T_s}}{j2\pi f} \quad (17)$$

and an approximator \tilde{H} of order $n_\beta/(n_\alpha - 1)$ is calculated, resulting in exactly the same transfer function order as in [6].

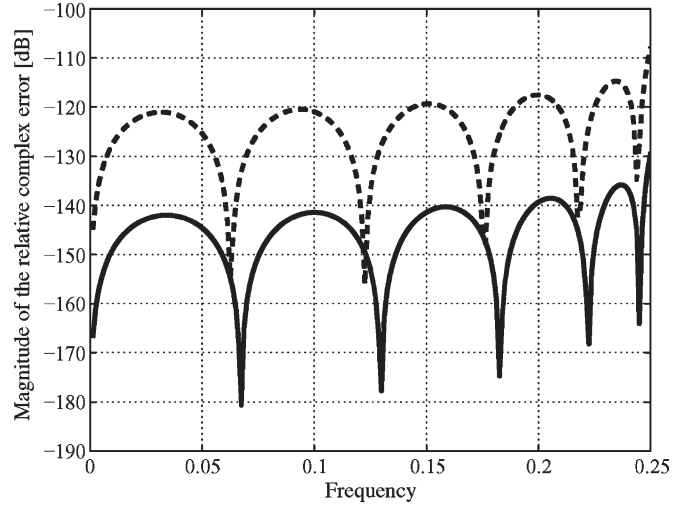


Fig. 1. Magnitude of the complex error of the half-band integrators for (dashed line) $\tau = 4.47796$ and (solid line) $\tau = 2.5853$.

TABLE I
POLES AND ZEROS OF THE FIFTH-ORDER, HALF-BAND INTEGRATOR WITH FRACTIONAL SAMPLE DELAY $\tau = 4.47796$

number	zero	pole
1	$6.7909 \pm j4.1856$	$-0.0505 \pm j0.1149$
2	$-0.8064 \pm j0.8064$	$0.0797 \pm j0.0585$
3	-3.6605	

TABLE II
POLES AND ZEROS OF THE FIFTH-ORDER, HALF-BAND INTEGRATOR WITH FRACTIONAL SAMPLE DELAY $\tau = 2.5853$

number	zero	pole
1	$15.88248 \pm j22.3436$	$0.0158 \pm j0.0294$
2	-14.3794	-0.7819
3	-0.8693	-0.2139
4	-0.2967	

Final approximation H is then equal to

$$\frac{\tilde{H}}{1 - z}. \quad (18)$$

The proposed algorithm finds a stable solution for $\tau = 2.5853$ that has a relative complex approximation error δ that is 20 dB smaller than the solution for $\tau = 4.47796$ given in [6] (see Fig. 1). Tables I and II summarize the poles and zeros of the resulting transfer function.

Applying the proposed method to half-band differentiators and Hilbert transformers gives similar results as in [5] and [6], respectively.

V. EQUALIZATION OF A DATA-ACQUISITION CHANNEL

In this section, the equalization of a data-acquisition channel is presented. The compensation procedure should consist of two steps. First, we have to determine the transfer function of the channel from noisy measurement. This is an identification problem. Second, a digital filter is to be designed for compensation of the identified transfer function. This is a filter design problem, which can be executed by the presented method.

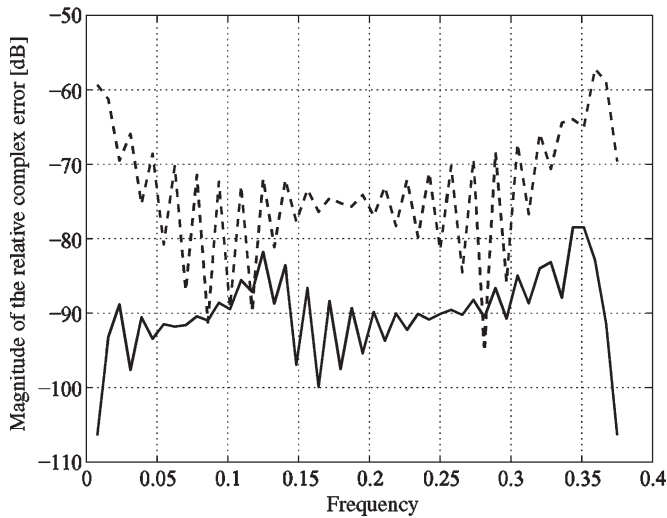


Fig. 2. Magnitude of the relative complex error of the data acquisition channel for (dashed line) $\tau = 35$ and (solid line) $\tau = 31.285868488$.

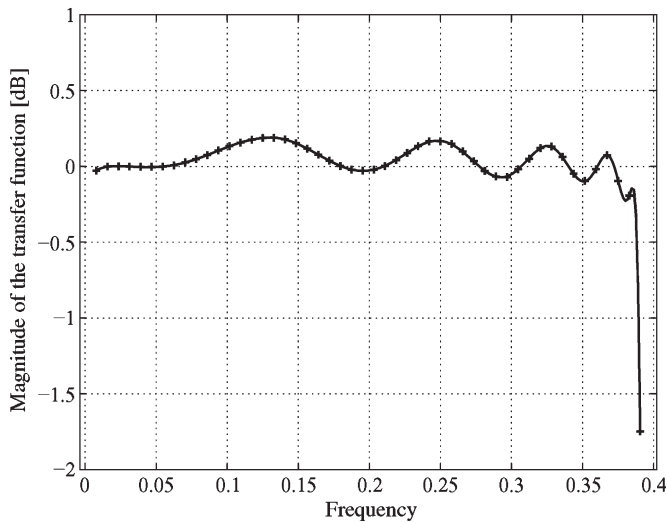


Fig. 3. Magnitude of the (x marks) measured and (solid line) estimated transfer functions.

In [12] and [13], compensation of an audio band Cauer filter with an order of 11 that was produced with laser-trimmed thick-film technology was presented. In [12], an amplitude- and phase-compensating IIR filter was designed. The relative complex error has been reduced to less than -58 dB using a 14/14 IIR filter for the amplitude compensation and a 20/20 IIR filter for the phase equalization. Thus, the resulting compensation filter has an order of 34/34. In [13], the order of the compensation filter was decreased by designing a new IIR filter, which compensated both the amplitude and the phase, with an order lower than 34/34. A 26/26 filter was obtained with practically the same fitting error as the 34/34 filter. The absolute value of the complex approximation error is less than 6 mdB, which means that the relative complex error is approximately -58 dB. The results can be seen in Fig. 2.

In this paper, the design of the compensation filter was done by the presented algorithm. The identified transfer function can be seen in Fig. 3, and the identified model parameters are sum-

TABLE III
POLES AND ZEROS OF THE ESTIMATED TRANSFER FUNCTION

number	zero	pole
1	-18.0234	$-0.7456 \pm j0.6084$
2	-6.3812	$-0.6164 \pm j0.6304$
3	$-1.5102 \pm j0.8343$	$-0.3939 \pm j0.6734$
4	$-0.9106 \pm j0.5618$	0.9463
5	$0.6413 \pm j0.6569$	$0.6413 \pm j0.6570$
6	0.9469	$-0.0377 \pm j0.6572$
7	$-0.5359 \pm j0.4709$	0.4962
8	$-0.0716 \pm j0.3983$	$0.3325 \pm j0.4051$
9	0.3294	

TABLE IV
POLES AND ZEROS OF THE STABLE APPROXIMATOR

number	zero	pole
1	$-0.8472 \pm j0.9889$	$-0.5186 \pm j0.6421$
2	$-0.6436 \pm j1.1940$	$-0.5615 \pm j0.5439$
3	$-0.3916 \pm j1.3260$	-0.6014
4	$-0.1129 \pm j1.3929$	$-0.2691 \pm j0.6962$
5	$-0.5922 \pm j0.6065$	$-0.1453 \pm j0.7015$
6	$0.1758 \pm j1.3957$	$0.0126 \pm j0.6817$
7	$0.4598 \pm j1.3361$	$0.1407 \pm j0.6828$
8	$0.7255 \pm j1.2173$	$0.2685 \pm j0.6489$
9	$0.9609 \pm j1.0452$	$0.3985 \pm j0.5844$
10	$1.1553 \pm j0.8275$	$0.5258 \pm j0.4835$
11	$1.2997 \pm j0.5739$	$0.6172 \pm j0.3588$
12	$1.3875 \pm j0.2946$	0.9421
13	1.4162	$0.6772 \pm j0.2266$
14	0.9414	$0.7138 \pm j0.0809$

marized in Table III. The frequency grid is [400:400:19600] Hz, where the sampling frequency is 51 200 Hz.

The resulting IIR filter is of nonminimum phase; hence, its inverse is unstable. Therefore, the compensation filter is a stable approximation of the inverse IIR filter. A stable approximation of order 26/26 was computed using the proposed algorithm. In the case of integer delay values (dashed line), the approximation error is approximately the same as that in [13]. However, if a fractional delay is allowed, then the algorithm returns a better approximation for which the magnitude of the relative complex error is less than -80 dB. The zeros and poles of the model are shown in Table IV.

VI. CONCLUSION

In this paper, a new method of the stable approximation with additional delay has been given. The theoretical background has been analyzed, and the difficulties of this kind of algorithm have been explained.

The main advantages of the proposed algorithm are given as follows:

- the automatic delay selection;
- the resulting approximation error being at least as small as the existing solutions.

Moreover, it turned out that, for several examples, the approximation error was significantly smaller.

Finally, the algorithm has been illustrated on the design of digital integrators, differentiators, and Hilbert transformers, and the equalization of a data-acquisition channel.

TABLE V
CASH PARAMETERS FOR THE RUNGE–KUTTA METHOD

$\frac{1}{5}$	$\frac{1}{5}$				
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$			
$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$		
1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{27}$	$\frac{35}{27}$	
$\frac{7}{8}$	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$
	$\frac{37}{378}$	0	$\frac{250}{621}$	$\frac{125}{594}$	0
					$\frac{512}{1771}$

APPENDIX A EULER METHOD

Using the Euler method to solve the ODE (8), P_{new} is calculated by applying the following expression:

$$P_{\text{new}} = P + f(P, \tau)\Delta\tau$$

where $f(P, \tau)$ is equal to

$$f(P, \tau) = -\left[\frac{\partial^2 C(P, \tau)}{\partial P^2}\right]^{-1} \frac{\partial^2 C(P, \tau)}{\partial P \partial \tau}.$$

APPENDIX B RUNGE–KUTTA METHOD

All the Runge–Kutta methods are based on

$$P_{\text{new}} = P + \phi(P, \tau)\Delta\tau.$$

To obtain a q -stage Runge–Kutta method (q function evaluations per step) with parameters a , b , and v , we choose

$$\phi(P, \tau) = \sum_{i=1}^q v_i k_i$$

where

$$k_i = f\left(P + \Delta\tau \sum_{j=1}^{i-1} b_{ij} k_j, \tau + a_i \Delta\tau\right) \quad (19)$$

$a_1 = 0$, and f is defined in (8). For convenience, the coefficients a , b , and v of the Runge–Kutta method can be written in the form of a Butcher array, i.e.,

$$\begin{array}{c|c} a & b \\ \hline & v^T \end{array}$$

where $a = [a_1, a_2, \dots, a_q]^T$, $v = [v_1, v_2, \dots, v_q]^T$, and $b = [b_{ij}]$.

In our algorithm, the so-called Cash–Karp parameters are used. The corresponding Butcher array is given in Table V [4].

APPENDIX C EXPLICIT EXPRESSION OF J

Using the fact that the linear projection and the differentiation are commutative operators, we find

$$J = \frac{\partial g(P)}{\partial P} = \begin{bmatrix} \text{Re} \left\{ \frac{\partial g_C(P)}{\partial P} \right\} \\ \text{Im} \left\{ \frac{\partial g_C(P)}{\partial P} \right\} \end{bmatrix}$$

where

$$\frac{\partial g_C(P)}{\partial P} = \left[\frac{\partial g_C(P, \Omega_1)}{\partial P}, \dots, \frac{\partial g_C(P, \Omega_F)}{\partial P} \right]^T \in \mathbb{C}^{F \times (n_\alpha + n_\beta + 1)}.$$

Every row can be formulated as

$$\frac{\partial g_C(P, \Omega_k)}{\partial P} = \begin{bmatrix} \frac{\partial g_C(P, \Omega_k)}{\partial P_{\text{Num}}} & \frac{\partial g_C(P, \Omega_k)}{\partial P_{\text{Den}}} \end{bmatrix}$$

where

$$\frac{\partial g_C(P, \Omega_k)}{\partial P_{\text{Num}}} = \begin{bmatrix} \frac{\partial g_C(P, \Omega_k)}{\partial \beta_0} & \dots & \frac{\partial g_C(P, \Omega_k)}{\partial \beta_{n_\beta}} \end{bmatrix}$$

$$\frac{\partial g_C(P, \Omega_k)}{\partial \beta_l} = W(\Omega_k) \frac{\Omega_k^l}{\sum_{s=0}^{n_\alpha} \alpha_s \Omega_k^s}$$

$$\frac{\partial g_C(P, \Omega_k)}{\partial P_{\text{Den}}} = \begin{bmatrix} \frac{\partial g_C(P, \Omega_k)}{\partial \alpha_0} & \dots & \frac{\partial g_C(P, \Omega_k)}{\partial \alpha_{n_\alpha}} \end{bmatrix}$$

$$\frac{\partial g_C(P, \Omega_k)}{\partial \alpha_l} = -W(\Omega_k) H(\Omega_k, P) \frac{\Omega_k^l}{\sum_{s=0}^{n_\alpha} \alpha_s \Omega_k^s}.$$

APPENDIX D EXPLICIT EXPRESSION OF $(\partial y(\tau)/\partial \tau)$

Reversing the differentiation and the projection operators leads to

$$\frac{\partial y(\tau)}{\partial \tau} = \begin{bmatrix} \text{Re} \left\{ \frac{\partial y_C(\tau)}{\partial \tau} \right\} \\ \text{Im} \left\{ \frac{\partial y_C(\tau)}{\partial \tau} \right\} \end{bmatrix} \in \mathbb{C}^{2F}$$

where

$$\frac{\partial y_C(\tau)}{\partial \tau} = \left[\frac{\partial y_C(\tau, \Omega_1)}{\partial \tau}, \dots, \frac{\partial y_C(\tau, \Omega_F)}{\partial \tau} \right]^T \in \mathbb{C}^F$$

with

$$\frac{\partial y_C(\tau, \Omega_k)}{\partial \tau} = W(\Omega_k) T(\Omega_k) (-j\omega_k) e^{-j\omega_k \tau}.$$

REFERENCES

- [1] L. Balogh and R. Pintelon, "Stable approximation of unstable transfer function models," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, Sorrento, Italy, Apr. 24–27, 2006, pp. 1027–1031.
- [2] R. Vuerinckx, "Design of digital Chebyshev filters in the complex domain," Ph.D. dissertation, Vrije Universiteit Brussel, Brussels, Belgium, 1999.

- [3] W. Rudin, *Real and Complex Analysis*. London, U.K.: McGraw-Hill, 1970.
- [4] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge, U.K.: Cambridge Univ. Press, 1992.
- [5] I. Kollár, R. Pintelon, and J. Schoukens, "Optimal FIR and IIR Hilbert transformer design via LS and minimax fitting," *IEEE Trans. Instrum. Meas.*, vol. 39, no. 6, pp. 847–852, Dec. 1990.
- [6] R. Pintelon and J. Schoukens, "Real-time integration and differentiation of analog signals by means of digital filtering," *IEEE Trans. Instrum. Meas.*, vol. 39, no. 6, pp. 923–927, Dec. 1990.
- [7] R. Vuerinckx, Y. Rolain, J. Schoukens, and R. Pintelon, "Design of stable IIR filters in the complex domain by automatic delay selection," *IEEE Trans. Signal Process.*, vol. 44, no. 9, pp. 2339–2344, Sep. 1996.
- [8] T. D'haene, R. Pintelon, and G. Vandersteen, "An iterative method to stabilise a transfer function in the s- and z- domains," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, Ottawa, ON, Canada, May 17–19, 2005, vol. 1, pp. 666–671.
- [9] T. Van Gestel, J. A. K. Suykens, P. Van Dooren, and B. De Moor, "Identification of stable models in subspace identification by using regularization," *IEEE Trans. Autom. Control*, vol. 46, no. 9, pp. 1416–1420, Sep. 2001.
- [10] J. Baratchart, L. J. Leblond, R. Partington, and N. Torkhani, "Robust identification from band-limited data," *IEEE Trans. Autom. Control*, vol. 42, no. 9, pp. 1318–1325, Sep. 1997.
- [11] S. Ellacott and J. Williams, "Rational Chebyshev approximations in the complex plane," *SIAM J. Numer. Anal.*, vol. 13, no. 3, pp. 310–323, Jun. 1976.
- [12] R. Pintelon, Y. Rolain, M. V. Bossche, and J. Schoukens, "Towards an ideal data acquisition channel," *IEEE Trans. Instrum. Meas.*, vol. 39, no. 1, pp. 116–120, Feb. 1990.
- [13] I. Kollár, R. Pintelon, Y. Rolain, and J. Schoukens, "Another step towards an ideal data acquisition channel," *IEEE Trans. Instrum. Meas.*, vol. 40, no. 3, pp. 659–660, Jun. 1991.



applications.

László Balogh was born in Békéscsaba, Hungary, in 1977. He received the M.S. degree in electrical engineering from Budapest University of Technology and Economics, Budapest, Hungary, in 2000 and the M.S. degree in mathematics from Eötvös Loránd University, Budapest, in 2005.

He is currently a Research Assistant with the Department of Measurement and Instrument Engineering, Budapest University of Technology and Economics. His research interests include system identification, digital signal processing, and related



Rik Pintelon (M'90–SM'96–F'98) was born in Ghent, Belgium, on December 4, 1959. He received the degree of electrical engineer (burgerlijk ingenieur), the degree of doctor in applied sciences, and the qualification to teach at the university level (geaggregeerde voor het hoger onderwijs) from Vrije Universiteit Brussel (VUB), Brussels, Belgium, in 1982, 1988, and 1994, respectively.

From October 1982 to September 2000, he was a Researcher with the Fund for Scientific Research-Flanders, VUB. Since October 2000, he has been a Professor with the Department of Fundamental Electricity and Instrumentation (ELEC), VUB. His research interests are parameter estimation/system identification and signal processing.