2020 17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)
Mexico City, Mexico. November 11-13, 2020

Discretization of the Robust Exact Filtering Differentiator Based on the Matching Approach

J. E. Carvajal-Rubio*, J. D. Sánchez-Torres†, M. Defoort‡, A. G. Loukianov§ and M. Djemai¶
*§Department of Electrical Engineering, CINVESTAV, Guadalajara, México
Email: Jose.Carvajal@cinvestav.mx*, louk@gdl.cinvestav.mx§
† Department of Mathematics and Physics, ITESO, Tlaquepaque, México
Email: dsanchez@iteso.mx
‡¶ LAMIH, CNRS UMR 8201, UPHF, Valenciennes, France.
Email: michael.defoort@uphf.fr‡, Mohamed.Djemai@uphf.fr¶

***Abstract*—This paper presents a time discretization of the robust exact filtering differentiator, a sliding mode differentiator coupled to filter, which provides a suitable approximation of the derivatives for some noisy signals. This realization rely on the stabilization of a pseudo linear discrete-time system, it is attained through the matching approach. As in the original case, the convergence of the robust exact filtering differentiator depends on the bound of a higher-order derivative. Nevertheless, this new realization can be implemented with or without the knowledge of such constant. It is demonstrated that the system trajectories converge to a neighborhood of the origin for a free-noise input. Finally, comparisons between the behavior of the differentiator with different design parameters are presented.**

***Index Terms*—Discrete-time systems, Online differentiation, Sliding mode differentiator, Homogeneous systems.**

I. Introduction

Usually, a control law or an observer is designed in continuous-time, but it is implemented in a digital system. They are implemented under the assumption that the sampling time is small enough to preserve its continuous-time property. However, due to discretization, such properties can be lost or modified. Different methodologies have been proposed to obtain adequate realizations which preserve some properties of the continuous-time systems. Some examples are Euler method, Exact discretization [1] and implicit discretization [2] to name a few.

An online robust differentiator is usefull for many applications, such as control laws based on derivatives of a signal, estimation of unmeasured states and parameters [3]–[5] for instance. In [6], a homogeneous differentiator was proposed. It can estimate the first n derivatives of a signal with a bounded $(n+1)$-th time derivative while presenting robustness properties to delays and bounded noises. Different time discretization methodologies have been used with the objective to preserve its accuracy and robustness properties [7]–[11]. Recently, a robust exact filtering differentiator was presented in [12], which improves some properties of the original differentiator presented in [6].

The authors would like to thank CONACyT for the scholarship given to the student with No. CVU 555845.

The contributions of this paper are to introduce a new discrete-time differentiator and to study its convergence property. Although a previous version of this idea was presented in [13], important details are presented in this work. The proposed discrete-time differentiator is based on the methodology presented in [11] and the robust exact filtering differentiator [12]. This paper is organized as follows. In Section II, some preliminaries on the differentiation problem are presented. In Section III, the standard differentiator and robust exact filtering differentiator [12] are introduced and compared. In Section IV, our discrete-time realization is introduced and analyzed. In Section V, in order to show the performance of the new discrete-time differentiator, two simulations are presented with different parameters and input signal. In Section VI, the main results of the paper are summarized and future work is presented.

II. Problem statement and preliminaries

A. Notations and properties

Let $x \in \mathbb{R}$. The absolute value of x, denoted by $|x|$, is defined as $|x| = x$ if $x \geq 0$ and $|x| = -x$ if $x < 0$. The set-valued function $\operatorname{sign}(x)$ is defined as $\operatorname{sign}(x) = \{1\}$ for $x > 0$, $\operatorname{sign}(x) = \{-1\}$ for $x < 0$, and $\operatorname{sign}(x) = [-1, 1]$ for $x = 0$. For $\gamma \geq 0$, the signed power γ of x is defined as $\lfloor x \rceil^{\gamma} = |x|^{\gamma} \operatorname{sign}(x)$, particularly, $\lfloor x \rceil^{0} = \operatorname{sign}(x)$.

For any matrices $\boldsymbol{C}$, $\boldsymbol{D} \in \mathbb{R}^{n \times m}$ and any positive definite matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ the following inequality holds:

$$\boldsymbol{C}^T\boldsymbol{D} + \boldsymbol{D}^T\boldsymbol{C} \leq \boldsymbol{C}^T\boldsymbol{\Lambda}\boldsymbol{C} + \boldsymbol{D}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{D}, \tag{1}$$

see [14] for details.

B. Problem statement

The objective of a differentiator is to obtain online the first n derivatives of a function even if there is noise in the measurement. $f_0(t)$ represents this function, $f_0 : \mathbb{R} \to \mathbb{R}$. $f_0(t)$ is assumed to be a function at least $(n+1) - th$ differentiable and with a $n+1$ derivative bounded by a known real number $L > 0$, i.e., $|f_0^{(n+1)}(t)| \leq L$. The input of the differentiator is defined as $f(t) = f_0(t) + \Delta(t)$ and $\Delta(t)$ corresponds to noise in the input. Additionally, it is also

978-1-7281-8987-1/20$31.00 ©2020 IEEE

assumed that $\Delta(t)$ is a Lebesgue-measurable bounded noise with $|\Delta(t)| \leq \delta$ for a real number $\delta > 0$, which can be unknown.

To design a differentiator, a space state representation is used. It allows to compute the derivatives $f_0^{(1)}(t)$, $f_0^{(2)}(t)$, $\cdots$, $f_0^{(n)}(t)$. The state variables are defined as $x_i(t) = f_0^{(i)}(t)$ and $\boldsymbol{x}(t) = \begin{bmatrix} x_0(t) & x_1(t) & x_2(t) & \cdots & x_n(t) \end{bmatrix}^T \in \mathbb{R}^{n+1}$. Therefore, one can obtain the following representation for the differentiation problem in the state space:

$$\begin{array}{rcl} \dot{\boldsymbol{x}}(t) & = & \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{e}_{n+1,n} f_0^{(n+1)}(t) \\ f(t) & = & \boldsymbol{e}_{1,n}^T \boldsymbol{x}(t) + \Delta(t) \end{array} \tag{2}$$

where the canonical vectors $\boldsymbol{e}_{i,j}$ are vectors such that $\boldsymbol{e}_{i,j} \in \mathbb{R}^{(j+1)\times 1}$, and are composed of zeros and one element 1 at position i, for instance, $\boldsymbol{e}_{1,n} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \end{bmatrix}^T$. Furthermore, $\boldsymbol{A} = [\boldsymbol{0} \ \boldsymbol{e}_{1,n} \ \boldsymbol{e}_{2,n} \ \cdots \ \boldsymbol{e}_{n,n}]$ is a nilpotent matrix of appropriate dimensions. Notice that the successive time derivatives of $f_0(t)$ can be obtained through the design of a state observer.

III. Differentiation

A. Standard Differentiator

In order to obtain the first n derivatives of $f_0(t)$, a continuous-time differentiator was proposed in [6] as:

$$\dot{\boldsymbol{z}}(t) = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{u}(\sigma_0 - \Delta(t)) \tag{3}$$

where $\boldsymbol{u}(\sigma_0) = [\upsilon_{0,n}(\sigma_0) \ \upsilon_{1,n}(\sigma_0) \ \cdots \ \upsilon_{n,n}(\sigma_0)]^T$, $\upsilon_{j,n}(\cdot) = -\lambda_{n-j} L^{\frac{j+1}{n+1}} \lfloor \cdot \rceil^{\frac{n-j}{n+1}}$, $\sigma_j = z_j - x_j$ and $\boldsymbol{z} = \begin{bmatrix} z_0 & z_1 & z_2 & \ldots & z_n \end{bmatrix}^T$ is the finite-time estimate of the state vector $\boldsymbol{x}$ using adequate $\lambda_j > 0$ (see [12]). Sequences of parameters λ_j are presented in [12] for $n \leq 7$, but they are not unique due to the fact that the sequences can be built for any $\lambda_0 > 1$ [6]. For instance, in [15], λ_j is defined for $1 \leq n \leq 10$. Since function $\lfloor z_0 - f(t) \rceil^0$ is discontinuous at $z_0 = f$, the solutions of system (3) are understood in the Filippov sense [16]. Under the above assumptions, the standard differentiator (3) ensures the following precision

$$\begin{array}{c} |z_j - f_0^{(j)}(t)| \leq \mu_j L^{\frac{j}{n+1}} \delta^{\frac{n+1-j}{n+1}}, \ \mu_j > 0, \\ j = 0, 1, \cdots, n, \end{array} \tag{4}$$

which corresponds to an asymptotically optimal accuracy [17].

B. Robust Exact Filtering Differentiator

Although differentiator (3) offers good performance when there exists a Lebesgue-measurable bounded noise $\Delta(t)$ such that $|\Delta(t)| \leq \delta$ with small average δ, its performance becomes significantly reduced when δ is large. On the other hand, a bounded noise is a signal of filtering order 0 and integral magnitude $\epsilon_0 \geq 0$. Now, it is assumed that $\Delta(t)$ is presented as $\Delta(t) = \Delta_0(t) + \Delta_1(t) + \cdots + \Delta_{n_f}(t)$, where $\Delta_j(t)$ is a signal of the global filtering order j and integral magnitude $\epsilon_j \geq 0$ with $j = 0, 1, \cdots, n_f$. More details can be founded in [12]. Note that a bounded noise signal satisfies the above assumption. In [12], a filtering differentiator has been proposed for such noises, and has the following structure:

$$\begin{aligned} \dot{w}_{j_f} &= -\lambda_{m+1-j_f} L^{\frac{j_f}{m+1}} \lfloor w_1 \rceil^{\frac{m+1-j_f}{m+1}} + w_{j_f+1} \\ \dot{w}_{n_f} &= -\lambda_{n+1} L^{\frac{n_f}{m+1}} \lfloor w_1 \rceil^{\frac{n+1}{m+1}} + z_0 - f(t) \\ \dot{z}_{j_d} &= -\lambda_{n-j_d} L^{\frac{n_f+1+j_d}{m+1}} \lfloor w_1 \rceil^{\frac{n-j_d}{m+1}} + z_{j_d+1} \\ \dot{z}_n &= -\lambda_0 L \lfloor w_1 \rceil^0 \\ j_f &= 1, 2, \cdots, n_f - 1. \quad j_d = 0, 1, 2, \cdots, n-1. \end{aligned} \tag{5}$$

where $m = n + n_f$, $n_f \geq 0$, n_f is the filtering order and the parameters λ_j are selected as in (3). n_f is selected greater than or equal to the highest filtering order of the signals $\Delta_j(t)$. Furthermore, it offers the following accuracy:

$$\begin{array}{l} |z_j - f_0^{(j)}(t)| \leq \mu_j L \rho^{n+1-j}, \ \ \mu_j > 0, \ \ j = 0, 1, 2, \cdots, n, \\ \rho = \max\left[\left(\frac{\epsilon_0}{L}\right)^{\frac{1}{n+1}}, \left(\frac{\epsilon_1}{L}\right)^{\frac{1}{n+2}}, \cdots, \left(\frac{\epsilon_{n_f}}{L}\right)^{\frac{1}{m+1}} \right]. \end{array} \tag{6}$$

For a bounded noise, accuracy (6) has the same form as accuracy (4). The advantage of using the robust exact filtering differentiator (5) instead of the standard one (3), is that (6) is a better accuracy than (3). Moreover, the filtering differentiator (6) rejects unbounded noises with a small local average [12]. As in [11], for a free-noise case ($\Delta(t) = 0$), the error system can be given as a pseudo linear system [18]:

$$\begin{array}{l} \begin{bmatrix} \dot{\boldsymbol{w}} \\ \dot{\boldsymbol{\sigma}} \end{bmatrix} = \boldsymbol{E} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{\sigma} \end{bmatrix} - \boldsymbol{e}_{m+1,m} f_0^{(n+1)}(t), \\ \boldsymbol{E} = \begin{bmatrix} -\lambda_m L^{\frac{1}{m+1}} |w_1|^{\frac{-1}{m+1}} & 1 & 0 & \cdots & 0 \\ -\lambda_{m-1} L^{\frac{2}{m+1}} |w_1|^{\frac{-2}{m+1}} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ -\lambda_1 L^{\frac{m}{m+1}} |w_1|^{\frac{-m}{m+1}} & 0 & 0 & \cdots & 1 \\ -\lambda_0 L |w_1|^{-1} & 0 & 0 & \cdots & 0 \end{bmatrix}, \end{array} \tag{7}$$

where $\boldsymbol{E} \in \mathbb{R}^{(m+1)\times(m+1)}$, $\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_{n_f} \end{bmatrix}^T$ and $\boldsymbol{\sigma} = [\sigma_0 \ \sigma_1 \ \cdots \ \sigma_n]^T$. The characteristic equation of $\boldsymbol{E}$ is $P(s) = s^{m+1} + \lambda_m L^{\frac{1}{m+1}} |w_1|^{\frac{-1}{m+1}} s^m + \lambda_{m-1} L^{\frac{2}{m+1}} |w_1|^{\frac{-2}{m+1}} s^{m-1} + \cdots + \lambda_0 L |w_1|^{-1}$. Its roots can be calculated by using the equation:

$$\left(|w_1|^{\frac{1}{m+1}} s\right)^{m+1} + \lambda_m L^{\frac{1}{m+1}} \left(|w_1|^{\frac{1}{m+1}} s\right)^m + \cdots + \lambda_0 L = 0. \tag{8}$$

Therefore, the $m+1$ roots s_j of (8) can be calculated from the following polynomial:

$$Q(b) = b^{m+1} + \lambda_m L^{\frac{1}{m+1}} b^m + \cdots + \lambda_0 L. \tag{9}$$

Then, s_j is calculated as $s_j = |w_1|^{\frac{-1}{m+1}} b_j$, where b_j corresponds to the roots of polynomial (9). This result will be used in Section IV-A.

IV. Discretization of the continuous-time systems

Let us denote the measurement time as t_k and $x_{j,k} = x_j\left(t_k\right)$, $\boldsymbol{x}_k = \left[x_{0,k}, \ldots, x_{n,k}\right]^T$. Then,

$$x_{j,k+1} = \sum_{l=j}^{n} \frac{\tau^{l-j}}{(l-j)!} x_{l,k} + h_{j,k}(\tau), \quad j = 0,1,2,\cdots,n.$$

is a discrete-time representation of the continuous-time system (2), where the sampling time is defined as $\tau = t_{k+1} - t_k$. It is obtained using Taylor series expansion with Lagrange's remainders [19], [20]. If $f_0^{(n+1)}(t)$ is an absolutely continuous function, $h_{j,k}\left(\tau\right)$ is given as:

$$h_{j,k}(\tau) = \frac{\tau^{n+1-j}}{(n+1-j)} f_0^{(n+1)}(\theta_j), \quad \theta_j \in \left(t_k, t_{k+1}\right), \quad j = 0,1,2,\cdots,n.$$

For a discontinuous function $f_0^{(n+1)}(t)$, $h_{j,k}(\tau)$ is presented as:

$$h_{j,k}(\tau) \in \frac{\tau^{n+1-j}}{(n+1-j)}\left[-1,1\right], \quad j = 0,1,2,\cdots,n.$$

A. Time Discretization of the Robust Exact Filtering Differentiator

For differentiator (7), $z_j(t_{k+1}) = z_{j,k+1}$ is proposed as a copy of $x_{j,k+1}$ with a injection term $\Gamma_{j+n_f+1,k} w_{1,k}$:

$$z_{j,k+1} = \sum_{l=j}^{n} \frac{\tau^{l-j}}{(l-j)!} z_{l,k} + \Gamma_{j+n_f+1,k} w_{1,k}, \quad j = 0,1,2,\cdots,n. \tag{10}$$

Obviously, $h_{j,k}(\tau)$ is omitted because it is not available. Furthermore, τ is considered constant. $\Gamma_{j+n_f+1,k}$ is defined hereafter. Based on Euler discretization, $w_{j,k+1}$ is proposed as:

$$\begin{aligned} w_{j,k+1} &= w_{j,k} + \tau w_{j+1,k} + \Gamma_{j,k} w_{1,k}, \\ w_{n_f,k+1} &= w_{n_f,k} + \tau(z_{0,k} - f(t)) + \Gamma_{n_f,k} w_{1,k}, \\ j &= 1,2,\cdots,n_f-1. \end{aligned} \tag{11}$$

where $\Gamma_{j,k}$ is defined hereafter. Using equations (10)-(11), the discrete-time differentiator is summarized as:

$$\begin{bmatrix} \boldsymbol{w}_{k+1} \\ \boldsymbol{z}_{k+1} \end{bmatrix} = \boldsymbol{\Psi}(\tau) \begin{bmatrix} \boldsymbol{w}_{k} \\ \boldsymbol{z}_{k} \end{bmatrix} - \tau \boldsymbol{e}_{n_f,m} f(t) + \boldsymbol{\Gamma}_k w_{1,k}, \tag{12}$$

where $\boldsymbol{w}_k = \begin{bmatrix} w_{1,k} & w_{2,k} & \cdots & w_{n_f,k} \end{bmatrix}^T$, $\boldsymbol{z}_k = \begin{bmatrix} z_{0,k} & z_{1,k} & \cdots & z_{n,k} \end{bmatrix}^T$, $\boldsymbol{\Gamma}_k = \begin{bmatrix} \Gamma_{1,k} & \Gamma_{1,k} & \cdots & \Gamma_{m+1,k} \end{bmatrix}^T$, $\boldsymbol{\Psi}(\tau)$ is given as:

$$\boldsymbol{\Psi}(\tau) = \begin{bmatrix} 1 & \tau & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \tau & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \tau & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & \tau & \frac{\tau^2}{2!} & \cdots & \frac{\tau^n}{n!} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \tau & \cdots & \frac{\tau^{(n-1)}}{(n-1)!} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix},$$

with $\boldsymbol{\Psi}(\tau) \in \mathbb{R}^{(m+1)\times(m+1)}$. Note that the first n_f rows of $\boldsymbol{\Psi}(\tau)$ only present 1, 0 and τ. Similarly to the continuous-time system error, the discrete-time system error of (12) can be represented as:

$$\begin{bmatrix} \boldsymbol{w}_{k+1} \\ \boldsymbol{\sigma}_{k+1} \end{bmatrix} = \left(\boldsymbol{\Psi}(\tau) + \boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T\right) \begin{bmatrix} \boldsymbol{w}_{k} \\ \boldsymbol{\sigma}_{k} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \boldsymbol{h}_k(\tau) \end{bmatrix} \tag{13}$$

where $\boldsymbol{\sigma}_k = \begin{bmatrix} \sigma_{0,k} & \sigma_{1,k} & \cdots & \sigma_{n,k} \end{bmatrix}^T$, and $\boldsymbol{h}_k(\tau) = \begin{bmatrix} h_{0,k}(\tau) & h_{1,k}(\tau) & \cdots & h_{n,k}(\tau) \end{bmatrix}^T$. Let d_j be the desired eigenvalues of the discrete-time system. Then the desired polynomial is given as $P_d(r) = \prod_{j=1}^{m+1}(r-d_j)$ and for a matrix case $P_d(\boldsymbol{\Psi}(\tau)) = \prod_{j=1}^{m+1}(\boldsymbol{\Psi}(\tau) - d_j \boldsymbol{I})$. The desired polynomial evaluated at $\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T$ is given by $P_d(\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T) = (\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T)^{m+1} + \sum_{j=0}^{m} \alpha_j (\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T)^j$. Here:

$$\begin{aligned} (\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T)^0 &= \boldsymbol{I} \\ (\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T)^1 &= (\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T) \\ (\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T)^2 &= \boldsymbol{\Psi}^2(\tau) + \boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T \boldsymbol{\Psi}(\tau) + \ldots \\ &\quad + \boldsymbol{\Psi}(\tau)\boldsymbol{\Gamma}_k\ \boldsymbol{e}_{1,m}^T + \boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T \boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T \\ &\ \ \vdots \\ (\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T)^{m+1} &= \boldsymbol{\Psi}(\tau)^{m+1} + \boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T \boldsymbol{\Psi}(\tau)^m + \cdots. \end{aligned}$$

Therefore, we obtain the following equation:

$$P_d(\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T) = P_d(\boldsymbol{\Psi}(\tau)) + \begin{bmatrix} * & \cdots & * & \boldsymbol{\Gamma}_k \end{bmatrix} \boldsymbol{S}.$$

Due to the Cayley–Hamilton theorem $P_d(\boldsymbol{\Psi}(\tau)+\boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T) = \mathbf{0}$ and therefore, $\boldsymbol{\Gamma}_k$ can be calculated as:

$$\boldsymbol{\Gamma}_k = -P_d(\boldsymbol{\Psi}(\tau)) \boldsymbol{S}^{-1} \boldsymbol{e}_{m+1,m}, \tag{14}$$

where

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{e}_{1,m}^T \\ \boldsymbol{e}_{1,m}^T \Psi(\tau) \\ \boldsymbol{e}_{1,m}^T \Psi^2(\tau) \\ \vdots \\ \boldsymbol{e}_{1,m}^T \Psi^m(\tau) \end{bmatrix}$$

Now, the objective is to select adequate roots d_j. In order to emulate the behavior of the continuous-time system, a mapping of the continuous-time domain to the discrete-time domain is used. One can use different approaches, Euler with $d_j = 1 + \tau s_j$, matching with $d_j = e^{\tau s_j}$ and bilinear with $d_j = \frac{1+s_j\tau/2}{1-s_j\tau/2}$ ones to name a few [21]. As $s_j = |w_1|^{\frac{-1}{m+1}} b_j$, Euler and Bilinear approaches have a singularity at $w_1 = 0$. Hence, the Matching approach is used:

$$d_j = e^{\tau s_j} = e^{\tau |w_1|^{\frac{-1}{m+1}} b_j}. \tag{15}$$

Theorem 1: Let the discrete-time differentiator (12) with $\boldsymbol{\Gamma}_k$ defined as (14), d_j defined as (15), $|f_0^{n+1}(t)| \leq L$ and $\Delta(t) = 0$. If $\mathcal{RE}(b_j) < 0$, then the trajectories of the error system (13) converge to a neighborhood of the origin and remain in this neighborhood, which is defined as:

$$\left|\left|\begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{\sigma}_k \end{bmatrix}\right|\right|_2 \leq K \left|\left|\boldsymbol{h}_k(\tau)\right|\right|_2.$$

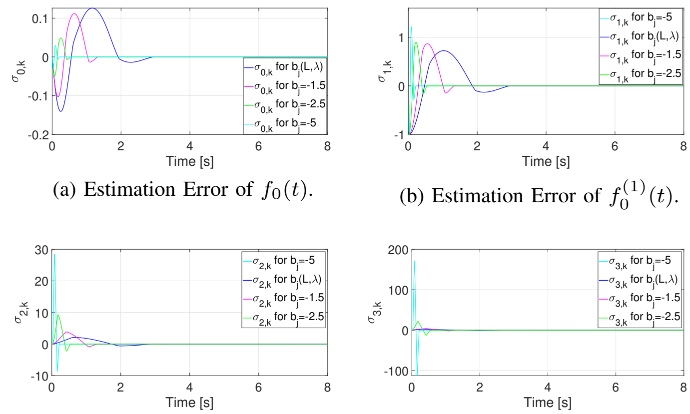
K and the proof are presented in Appendix A. Note that the roots b_j can be selected independently of λ_j and L from Theorem 1. This allows to implement the differentiator even if L is unknown. Furthermore, if b_j are selected as $b_1 = b_2 = b_3 = \cdots = b_{m+1}$, $\boldsymbol{\Gamma}_k$ presents a less complex equation than with $b_j \neq b_{j+1}$.

V. Results

In this Section, two simulations are performed. In the first one, a free-noise case is considered. In the second one, a noisy input is considered. To implement (13), $\boldsymbol{\Gamma}_k$ is calculated offline and expressed as a function of d_j. d_j is updated using Equation (15). Four differentiators are compared, three of them with multiple b_j and one where its roots correspond to the roots of polynomial (9), where λ_j is selected as in [12] and L is as $|f_0(t)| \leq L$. For the last one, b_j will be represented as $b_j(L, \lambda)$.

A. Simulation I

Here, a free-noise case is considered. The parameters of the differentiator are set as $n_f = 2$, $n = 3$, $\tau = 0.01\,\text{sec}$, $\lambda_0 = 1.1$, $\lambda_1 = 6.75$, $\lambda_2 = 20.26$, $\lambda_3 = 32.24$, $\lambda_4 = 23.72$ and $\lambda_5 = 7$. For this simulation, $f_0(t) = t\cos(t/2)$, then $|f^{(4)}(t)| \leq L = 2$ for $t \leq 31.54619\,\text{sec}$, $b_1(L,\lambda) = -2.8072 + 2.7583i$, $b_2(L,\lambda) = -2.8072 - 2.7583i$, $b_3(L,\lambda) = -0.2725 + 0.3729i$, $b_4(L,\lambda) = -0.2725 - 0.3729i$, $b_5(L,\lambda) = -1.0831$ and $b_6(L,\lambda) = -0.6148$. For the differentiator with multiple b_j, the selected roots are $b_j = -1.5$, $b_j = -2.5$, and $b_j = -5$. The estimation errors are presented in Figures 1a-1d. It can be seen that the differentiator gives an adequate estimation of the function and its derivatives. An interesting result is that the trajectories of the differentiators converge to a neighborhood with a different settling-time, 1.64 seconds for $b_j = -1.5$, 0.71 seconds for $b_j = -2.5$, 0.31 seconds for $b_j = -5$, and 2.96 seconds for $b_j(L,\lambda)$. The above fact shows that there is a relation between this settling-time and the roots b_j.



(a) Estimation Error of $f_0(t)$. (b) Estimation Error of $f_0^{(1)}(t)$.

(c) Estimation Error of $f_0^{(2)}(t)$. (d) Estimation Error of $f_0^{(3)}(t)$.

Fig. 1: Estimation Error in Simulation I.

B. Simulation II

The main advantage of differentiator (5) over the standard differentiator [6] is to present a better accuracy than the standard one in the presence of noises. Therefore, in this simulation $\Delta(t) = \cos(10000t) + \eta(t)$, $\eta(t)$ is a normal distribution with mean 0 parameter and standard deviation 1. In contrast to simulation I, $f_0(t) = \sin(t) + \cos(2t) + \sin(3t) + \cos(4t)$, $|f^{(4)}(t)| \leq L = 320$, $b_1(L,\lambda) = -6.5408 + 6.4269i$, $b_2(L,\lambda) = -6.5408 - 6.4269i$, $b_3(L,\lambda) = -0.6348 + 0.8689i$, $b_4(L,\lambda) = -0.6348 - 0.8689i$, $b_5(L,\lambda) = -2.5235$ and $b_6(L,\lambda) = -0.6348$ are considered. The multiple poles are the same as in Simulation I. Furthermore, a lower sampling time is used to obtain adequate estimations, $\tau = 0.0001\,\text{sec}$. The noisy input and $f_0(t)$ are presented in Figure 2 whereas the estimations $z_{j,k}$ of the time derivatives of the noisy signal are shown in Figures 3-6.

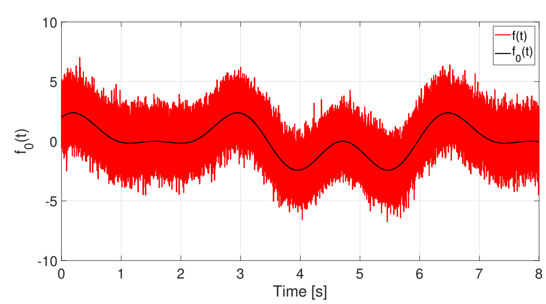


Fig. 2: Estimation of $f_0(t)$.

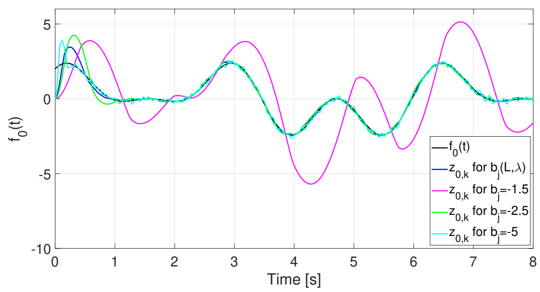


Fig. 3: Estimation of $f_0(t)$.

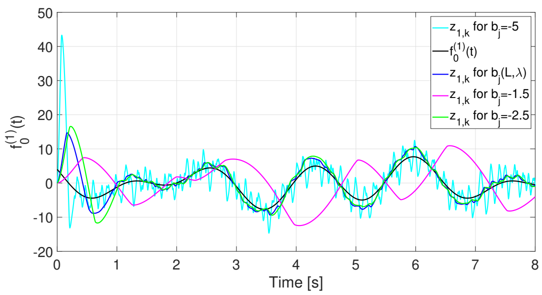


Fig. 4: Estimation of $f_0^{(1)}(t)$.

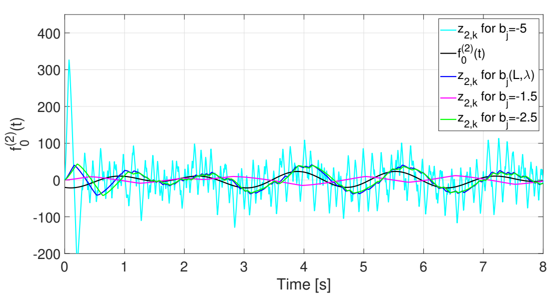


Fig. 5: Estimation of $f_0^{(2)}(t)$.

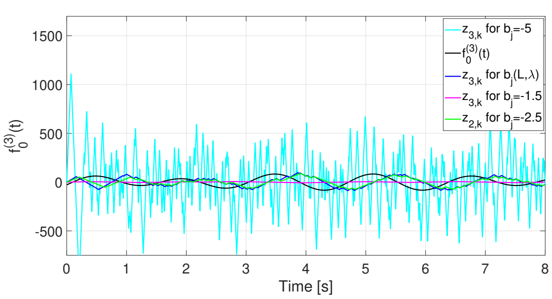


Fig. 6: Estimation of $f_0^{(3)}(t)$.

Note that $\cos(10000t)$ can be represented as a signal of global filtering j for any integer $j \geq 0$. Therefore, accuracy (6) is better the accuracy (4). As it can be seen in Figure 7-10, the best estimations come from the differentiators with $b_j = -2.5$ and $b_j = (L, \lambda)$. Both differentiators present adequate estimations of function $f_0(t)$ and its time derivatives. It is interesting to see that for multiple real b_j, a reduction of b_j increases its sensitivity with respect to noise whereas an increase of b_j reduces its accuracy. Nevertheless, a deeper analysis is required to explain this behavior. As in the previous simulation, the trajectories of the differentiators converge to a neighborhood with different settling-times, 0 seconds for $b_j = -1.5$, 9.643 seconds for $b_j = -2.5$, 2.321 seconds for $b_j = -5$, and 7.718 seconds for $b_j(L, \lambda)$. Note that for $b_j = -1.5$, its initial condition belong to this neighborhood.

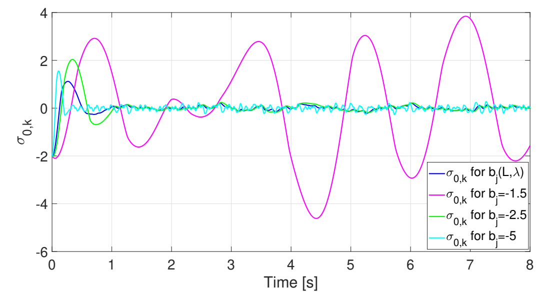


Fig. 7: Estimation Error of $f_0(t)$.

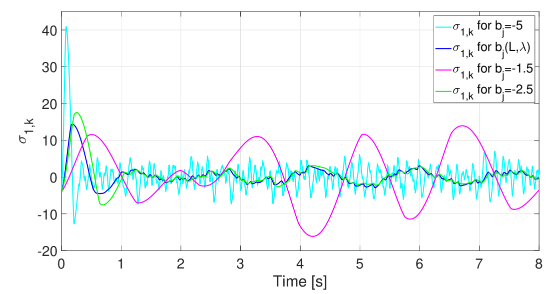


Fig. 8: Estimation Error of $f_0^{(1)}(t)$.

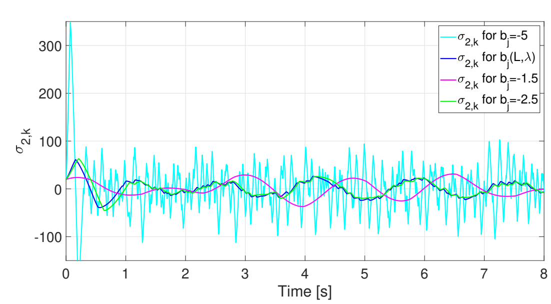


Fig. 9: Estimation Error of $f_0^{(2)}(t)$.

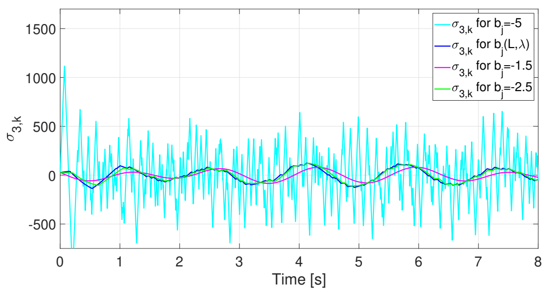


Fig. 10: Estimation Error of $f_0^{(3)}(t)$.

VI. Conclusion

A new time discretization of the robust exact filtering differentiator is presented. It can be implemented with or without the knowledge of L and λ_j. It was demonstrated that, for a free-noise case, and, under some assumptions, the trajectories of the system converge to a neighborhood of the origin. Simulation I suggests a relationship between the settling-time and the roots b_j, whereas Simulation II shows a relation between sensitivity to noise and the values of b_j. Future work will address a theoretical analysis of the convergence of the trajectories of the error system to a neighborhood of the origin in the presence of noisy inputs, an estimation of this neighborhood and the corresponding settling-time function.

Appendix

Let $\boldsymbol{E} = \left(\boldsymbol{\Psi}(\tau) + \boldsymbol{\Gamma}_k \boldsymbol{e}_{1,m}^T\right)$. Consider the candidate Lyapunov function defined as:

$$V_k = \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{\sigma}_k \end{bmatrix}^T \boldsymbol{P} \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{\sigma}_k \end{bmatrix}, \qquad (16)$$

where $\boldsymbol{P}$ is a real positive definite matrix defined such that

$$\boldsymbol{E}^T \boldsymbol{P} \boldsymbol{E} - \boldsymbol{P} = -\boldsymbol{Q},$$

with $\boldsymbol{Q}$ be a real positive definite matrix and $\lambda_{\min}(Q) > 1$. From Equations (13) and (16), one gets

$$\begin{aligned} &V_{k+1} - V_k = \\ &= -\begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{\sigma}_k \end{bmatrix}^T \boldsymbol{Q} \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{\sigma}_k \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{h}_k(\tau) \end{bmatrix}^T \boldsymbol{P} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{h}_k(\tau) \end{bmatrix} \\ &\quad - 2 \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{\sigma}_k \end{bmatrix}^T \boldsymbol{E}^T \boldsymbol{P} \boldsymbol{E} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{h}_k(\tau) \end{bmatrix}. \end{aligned}$$

Using inequality (1), the following inequality is obtained:

$$\begin{aligned} &V_{k+1} - V_k \leq \left(\lambda_{\max}(\boldsymbol{E}) + \lambda_{\max}(\boldsymbol{P})\right) \|\boldsymbol{h}_k(\tau)\|_2^2 - \cdots \\ &- \left(\lambda_{min}(Q) - 1\right) \left\| \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{\sigma}_k \end{bmatrix} \right\|_2^2. \end{aligned}$$

Therefore with the condition

$$\left\| \begin{bmatrix} \boldsymbol{w}_k \\ \boldsymbol{\sigma}_k \end{bmatrix} \right\|_2 > K \left\| \boldsymbol{h}_k(\tau) \right\|_2,$$

$$K = \sqrt{\frac{\lambda_{\max}(\boldsymbol{E}) + \lambda_{\max}(\boldsymbol{P})}{\lambda_{\min}(\boldsymbol{Q}) - 1}},$$

one obtains $V_{k+1} - V_k < 0$. This concludes the proof. □

References

[1] N. Kazantzis and C. Kravaris, "Time-discretization of nonlinear control systems via taylor methods," *Computers & Chemical Engineering*, vol. 23, no. 6, pp. 763 – 784, 1999.

[2] B. Brogliato and A. Polyakov, "Digital implementation of sliding-mode control via the implicit method: A tutorial," *International Journal of Robust and Nonlinear Control*, 2020. [Online]. Available: https://hal.inria.fr/hal-02523011

[3] P. Kaveh and Y. B. Shtessel, "Blood glucose regulation using higher-order sliding mode control," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 4-5, pp. 557–569, 2008.

[4] Y. B. Shtessel, I. A. Shkolnikov, and A. Levant, "Smooth second-order sliding modes: Missile guidance application," *Automatica*, vol. 43, no. 8, pp. 1470–1476, 2007.

[5] M. Iqbal, A. I. Bhatti, S. I. Ayubi, and Q. Khan, "Robust parameter estimation of nonlinear systems using sliding-mode differentiator observer," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 2, pp. 680–689, Feb 2011.

[6] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *International Journal of Control*, vol. 76, no. 9-10, pp. 924–941, 2003.

[7] M. Livne and A. Levant, "Proper discretization of homogeneous differentiators," *Automatica*, vol. 50, no. 8, pp. 2007–2014, 2014.

[8] S. Koch, M. Reichhartinger, M. Horn, and L. Fridman, "Discrete-time implementation of homogeneous differentiators," *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 757–762, Feb 2020.

[9] J.-P. Barbot, A. Levant, M. Livne, and D. Lunz, "Discrete differentiators based on sliding modes," *Automatica*, vol. 112, p. 108633, 2020.

[10] J. E. Carvajal-Rubio, A. G. Loukianov, J. D. Sánchez-Torres, and M. Defoort, "On the discretization of a class of homogeneous differentiators," in *2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, September 2019, pp. 1–6.

[11] S. Koch and M. Reichhartinger, "Discrete-time equivalent homogeneous differentiators," in *2018 15th International Workshop on Variable Structure Systems (VSS)*, July 2018, pp. 354–359.

[12] A. Levant and M. Livne, "Robust exact filtering differentiators," *European Journal of Control*, vol. 55, pp. 33–44, 2020.

[13] J. E. Carvajal-Rubio, J. D. Sánchez-Torres, M. Defoort, A. Loukianov, and M. Djemai, "A discrete-time matching filtering differentiator," *ArXiv*, vol. abs/2008.09863, 2020.

[14] A. S. Poznyak, *Advanced Mathematical Tools for Automatic Control Engineers: Deterministic Techniques*, 1st ed. Elsevier, 2008, vol. 1.

[15] M. Reichhartinger, S. Spurgeon, M. Forstinger, and M. Wipfler, "A robust exact differentiator toolbox for matlab®/simulink®," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1711 – 1716, 2017, 20th IFAC World Congress.

[16] A. F. Filippov, *Differential Equations with Discontinuous Righthand Sides*, 1st ed., ser. Mathematics and its Applications. Springer Netherlands, 1988., vol. 18.

[17] A. Levant, M. Livne, and X. Yu, "Sliding-mode-based differentiation and its application," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1699 – 1704, 2017.

[18] H. Ghane and M. B. Menhaj, "Eigenstructure-based analysis for non-linear autonomous systems," *IMA Journal of Mathematical Control and Information*, vol. 32, no. 1, pp. 21–40, 08 2013. [Online]. Available: https://doi.org/10.1093/imamci/dnt026

[19] W. J. Firey, "Remainder formulae in taylor's theorem," *The American Mathematical Monthly*, vol. 67, no. 9, pp. 903–905, 1960.

[20] T. Apostol, *Calculus: One-Variable Calculus with an Introduction to Linear Algebra*, 2nd ed. John Wiley & Sons, 1967, vol. 1.

[21] G. A. Perdikaris, *Computer Controlled Systems: Theory and Applications*, 1st ed., ser. Intelligent Systems, Control and Automation: Science and Engineering. Springer Netherlands, 1991, vol. 8.