A Robust Exact Differentiator Toolbox for Matlab®/Simulink®

M. Reichhartinger [*] **S.K. Spurgeon** [**] **M. Forstinger** [*,***]
M. Wipfler [*,***]

[*] *Institute of Automation and Control, Graz University of Technology, Graz, Austria, (email: markus.reichhartinger@tugraz.at)*
[**] *Department of Electronic & Electrical Engineering, University College London, London WC1E 7JE, UK (e-mail: s.spurgeon@ucl.ac.uk)*
[***] *Kristl, Seibt & Co GmbH, Baiernstrasse 122a, 8052 Graz, Austria*

Abstract: This paper demonstrates the functionality and ease of use of a recently implemented robust exact differentiator block for numerical simulations performed within the Matlab/Simulink software environment. It is demonstrated that the differentiator block may be used for various applications and may be easily integrated within existing Simulink models. The underpinning discrete time differentiation algorithm is briefly outlined and its parameters up to differentiator order 10 are presented. An extended version of the toolbox supports the so-called automatic code generation feature of Matlab/Simulink. This functionality allows compilable code to be produced for many available hardware platforms. Three applications are presented in the paper, where two require the production of executable code. The simulation based application presents a differentiator based edge detection algorithm for image processing purposes which utilises the simulink block directly.

Keywords: differentiators, robust estimation, software tools, nonlinear control systems, control applications

1. INTRODUCTION

From a theoretical viewpoint, differentiators of arbitrary order based on higher order sliding mode techniques are well studied. This is documented by a large number of publications and their associated citations, see e.g. Levant (2003, 1998). Attractive characteristics such as finite time estimation of the time-derivatives up to an arbitrary order n of a noise-free signal $f(t)$ are proven and discussed in detail. Exploiting these differentiators for state observation of dynamical systems, the separation principle consequently is fulfilled, see Levant (2003). In the case of a noisy signal $f(t)$, these differentiators are shown to provide accurate estimates and even in a discrete-time environment, the behaviour of the differentiators is well documented, see Livne and Levant (2014). However, real world applications or even simulation studies demonstrating the performance of sliding mode based differentiators of order greater than one are sparse. Applications of first order differentiators are more often reported, see e.g. Imine et al. (2015, 2011). The objective of the work summarized in this article is to describe a differentiator toolbox which can be used for both numerical simulation studies and real world experiments. The toolbox is freely available for the Matlab/Simulink environment [1]. It provides a single block which is straightforward to integrate in existing Simulink models. Figure 1 shows a screen shot of the block which has a single input port and two output ports. The signal f to be differentiated has to be connected at the input port of the differentiator block (labeled with `f` in Fig. 1). The first output port, which is labeled `c`, represents the convergence signal. It indicates whether the differentiator provides reliable estimates of the derivatives of the input signal which are included in the second output port labeled with `x`. The output `x` is a vector including the first n time derivatives of f. A detailed discussion including the usage, the tuning and some examples are given in Reichhartinger and Spurgeon (2016). The differentiator algorithm implemented in the toolbox is the discrete time differentiator proposed in Livne and Levant (2014) and outlined in Section 2. Three applications demonstrating the application of the differentiator block are given in Section 3. Section 4 concludes the paper.

[1] The toolbox can be downloaded at www.reichhartinger.at. A version which also supports automatic code generation can be requested from the authors.

2. REVIEW OF THE IMPLEMENTED ALGORITHM

Sliding mode based differentiators are often represented in the so-called recursive structure. This may be motivated by established tuning procedures, typically the $n+1$ parameters of the differentiator are adjusted using a step-by-step algorithm. At each step, the order of the differ-

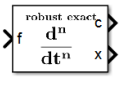


Fig. 1. Simulink block of the implemented differentiator.

entiator is increased requiring one additional parameter to be selected. After appropriately selecting this parameter, the order of the differentiator is further increased by 1 until the desired order is reached. However, in this toolbox the differentiator is implemented based on a non-recursive structure. From the authors' point of view, an implementation of the non-recursive structure is intuitive and straightforward and, additionally, a discrete-time realization may easily be obtained. The differentiator in non-recursive form is given by

$$\frac{dx_0}{dt} = x_1 + \varphi_0, \tag{1a}$$

$$\frac{dx_1}{dt} = x_2 + \varphi_1, \tag{1b}$$

$$\vdots$$

$$\frac{dx_n}{dt} = \varphi_n, \tag{1c}$$

where

$$\varphi_i(e) = \kappa_i \lfloor e \rceil^{\frac{n-i}{n+1}} \tag{2}$$

and $e = f - x_0$. The positive constants κ_i with $i = 0, 1, \ldots, n$ denote the parameters of the differentiator and the operator $\lfloor . \rceil^{\cdot}$ is given by

$$\lfloor a \rceil^b = |a|^b \operatorname{sign}(a). \tag{3}$$

Given an appropriate set of parameters κ_i, it is well known that in the case of a noise-free signal f, the estimation error e and its first n time derivatives converge to zero within finite time. Consequently, the variable x_i of the differentiator (1) is a finite time estimate of $f^{(i)}(t)$, i.e. the i^{th} time derivative of $f(t)$. In the case of a signal f corrupted by noise of maximum amplitude ε, the accuracy of the differentiator (1) is given by

$$\left| x_i - f^{(i)} \right| = \mathcal{O}\left(\varepsilon^{\frac{n+1-i}{n+1}} \right). \tag{4}$$

This relationship motivates selecting the order n of the differentiator higher than required. Consider for example an application requiring an estimate of the velocity of a mass based on a measurement of position. A differentiator of first order provides the accuracy of the first time derivative, i.e. the velocity, by $\mathcal{O}(\varepsilon^{\frac{1}{2}})$ whereas a differentiator of second-order provides an estimate of the first time-derivative with accuracy $\mathcal{O}(\varepsilon^{\frac{2}{3}})$. In order to accommodate a discrete time realization of the differentiator, in this paper the homogeneous discrete-time differentiator as proposed in Livne and Levant (2014) is constructed by the composition of a discrete-time version of a chain of integrators combined with the non-linear terms as given in Eq. (2), i.e.

$$\begin{bmatrix} x_{0,k+1} \\ x_{1,k+1} \\ \vdots \\ x_{n,k+1} \end{bmatrix} = \mathbf{\Phi}(T) \begin{bmatrix} x_{0,k} \\ x_{1,k} \\ \vdots \\ x_{n,k} \end{bmatrix} + T \begin{bmatrix} \varphi_0(f_k - x_{0,k}) \\ \varphi_1(f_k - x_{0,k}) \\ \vdots \\ \varphi_n(f_k - x_{0,k}) \end{bmatrix}, \tag{5}$$

where the matrix $\mathbf{\Phi}(T)$ is computed using

$$\mathbf{\Phi}(T) = \sum_{\nu=0}^{\infty} \frac{(\mathbf{A}T)^{\nu}}{\nu!} \tag{6}$$

and the $(n+1)\times(n+1)$ matrix $\mathbf{A}$ is given in the companion form by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ & & & & \ddots & \\ 0 & 0 & 0 & 0 & \ddots & 1 \\ 0 & 0 & 0 & 0 & \ldots & 0 \end{bmatrix}. \tag{7}$$

The variables $x_{i,k}$ denote the discrete time estimates of the i^{th}-time derivative of f at the time instant $t = kT$ with $k = 0, 1, 2, \ldots$ and the constant discretization time T. Equation (5) was implemented in a so-called Simulink C s-function which eventually represents the robust exact differentiator block, see Reichhartinger and Spurgeon (2016). In addition to the estimates of the derivatives of the input signal f, the block provides a convergence output signal given by $f_k - x_{0,k}$. This signal supports the tuning of the differentiator as explained in detail in Reichhartinger and Spurgeon (2016). The parameters κ_i of the differentiator implemented in the s-function are listed in Table 1. Note that these parameters are listed assuming that the convergence rate / robustness factor of the differentiator is selected as 1. This factor, which represents the only tuning parameter of the differentiator, scales the parameters κ_i such that uncertainty represented by $f^{(n+1)}$ is eventually dominated by $\kappa_n c^{n+1}$. Due to space restrictions, the parameter scaling factor is not included in Table 1.

Table 1. Parameters implemented in the differentiator block.

n	κ_0	κ_1	κ_2	κ_3	κ_4	κ_5	κ_6	κ_7	κ_8	κ_9	κ_{10}
1	2.1	1.1									
2	3.1	3.2	1.1								
3	4.1	6.3	4.3	1.1							
4	5.1	10.4	10.6	5.4	1.1						
5	6.1	15.5	21.0	16.0	6.5	1.1					
6	7.1	21.6	36.5	37.0	22.5	7.6	1.1				
7	8.1	28.7	58.1	73.5	59.5	30.1	8.7	1.1			
8	9.1	36.8	86.8	131.6	133.0	89.6	38.8	9.8	1.1		
9	10.1	45.9	123.6	218.4	264.6	222.6	128.4	48.6	10.9	1.1	
10	11.1	56.0	169.5	342.0	483.0	487.2	351.0	177.0	59.5	12.0	1.1

It should be noted that the settings in Table 1 describe a completely new set of parameters and do not rely on the sets typically used, see e.g. Shtessel et al. (2014). Additionally, no set corresponding to a differentiator order greater than 6 has been published to date.

3. APPLICATIONS

Three applications are considered.

3.1 Suppression of drive-train oscillations

An automotive test bed is first analysed. The main components of the test bed are a combustion engine and electric motor which are coupled via a shaft, see Fig. 2. The electric motor is used to emulate the load characteristics of the combustion engine. Compared to a conventional drive train as typically attached to a combustion engine, the stiffness of the shaft is high. This increased stiffness excites undesirable torsional oscillations in the shaft which occur only on the test bed and are not present in the real environment of the combustion engine. Hence, unrealistic load behaviour is generated and, even worse, components of the test bed may be destroyed. Any hardware solution

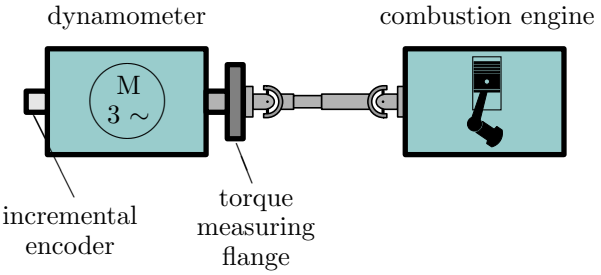


Fig. 2. Engine test bench system.

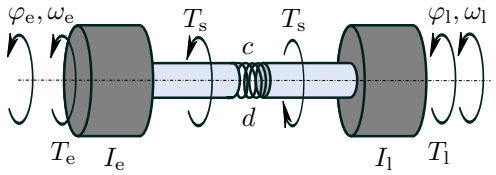


Fig. 3. Drive train, see Wipfler et al. (2016).

(e.g. attaching a shaft with lower stiffness and improved damping characteristics) to solve this problem is typically highly undesirable. Such an approach would have to be adapted dependent on the combustion engine, i.e. it would cause additional cost, and furthermore would not allow high transient torques to be transmitted from the electric drive to the combustion engine, which is a requirement for dynamical test scenarios. So-called active damping solutions tackle this problem by applying an additional torque generated by the electric drive so that undesired oscillations are adequately damped.

The torsional dynamics of the test bed are typically described by a two-mass oscillator (see Fig. 3) with linear shaft characteristics, see Kokal et al. (2013). The corresponding mathematical model describing the system dynamics is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\varphi\\ \omega\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{c}{I} & -\frac{d}{I}\end{bmatrix}\begin{bmatrix}\varphi\\ \omega\end{bmatrix} + \begin{bmatrix}0 & 0\\ \frac{1}{I_e} & -\frac{1}{I_l}\end{bmatrix}\begin{bmatrix}T_e\\ T_l\end{bmatrix}, \tag{8}$$

where the state variables

$$\varphi := \varphi_e - \varphi_l \tag{9}$$
$$\omega := \omega_e - \omega_l \tag{10}$$

denote the torsion angle of the shaft and its time derivative, i.e. the angular speed, respectively. The torsional stiffness c and the damping coefficient d are constant shaft parameters, the moment of inertia of the electric drive is given by I_e. The positive constant I_l is the load moment of inertia w.r.t. the electric drive, which represents mainly the moment of inertia of the shaft and of the combustion engine. The overall moment of inertia is given by

$$I = \frac{I_e I_l}{I_e + I_l}. \tag{11}$$

System (8) may be excited by the torque of the electric drive, i.e. T_e and the load torque T_l generated by the combustion engine. The shaft torque

$$T_s = -c\varphi - d\omega \tag{12}$$

is typically measured in such test beds and, therefore, is directly available as an input for control tasks. Some characteristic parameters of the test bed used in this application are listed in Table 2.

The active damping strategy considered in this paper, see Wipfler et al. (2016), also acts as a control signal T_c which

Table 2. Characteristic parameters of the test bed.

parameter	value	unit
c	30	kNm/rad
d	0.01	Nms/rad
I_e	6.31	kgm^2
I_l	2.66	kgm^2

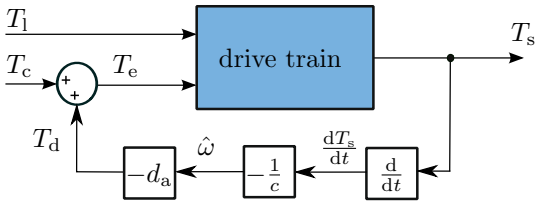


Fig. 4. Closed loop system, see Wipfler et al. (2016).

is designed to fulfil the overall control task of the test bed (driving the combustion engine at a desired speed profile), see Fig. 4. The torque T_e yields

$$T_e = T_c + T_d, \tag{13}$$

where T_d represents the corresponding torque designed to suppress the undesired torsional oscillations. If the torque T_d is chosen as

$$T_d = -d_a\omega, \tag{14}$$

where $d_a > 0$ is the active damping constant, the damping properties of system (8) can be improved as the overall damping coefficient w.r.t. to the electric drive is now $d + d_a$. According to (10), the calculation of ω is based on the rotational speed of the combustion engine; thus, an accurate measurement of ω_l is desired. As this is not the case with a typical engine test bed, an alternative based on the measured shaft torque is used. Assuming the damping coefficient d of the drive train is negligible, the measured shaft torque may be approximated by

$$T_s = -c\varphi - d\omega \approx -c\varphi. \tag{15}$$

Differentiating Eq. (15) w.r.t. time t yields

$$\frac{\mathrm{d}T_s}{\mathrm{d}t} \approx -c\omega \tag{16}$$

and an estimate of ω is established by

$$\hat{\omega} = -\frac{1}{c}\frac{\mathrm{d}T_s}{\mathrm{d}t} \tag{17}$$

which is used in the active damping procedure as

$$T_d = -d_a\hat{\omega}. \tag{18}$$

The active damping constant d_a may be chosen so that the characteristic polynomial of the dynamic matrix of the closed loop system consisting of system (8) and control law (13) with T_d as selected in Eq. (18) has real roots.

This damping strategy was implemented in a simulation model of a test bed using the robust exact differentiator toolbox as well as a linear differentiator, as shown in Wipfler et al. (2016) to generate the estimate (17). The test bed considered in this work is used for end-of-line tests in which the engine is dragged from 100 to 500 rpm without injection. During the procedure, characteristic values of the engine are measured and potential failures can be detected before the engine is taken into regular operation.

The simulation model considers a more complex mechanical system than that presented in Fig. 3. Dead time, first order dynamics for the torque measurement and the torque

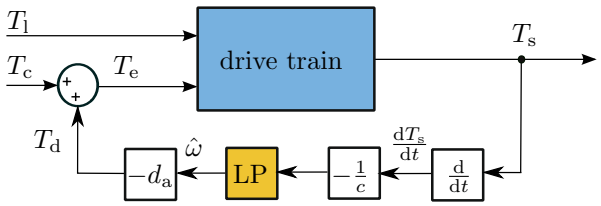


Fig. 5. Closed loop system with linear differentiation.

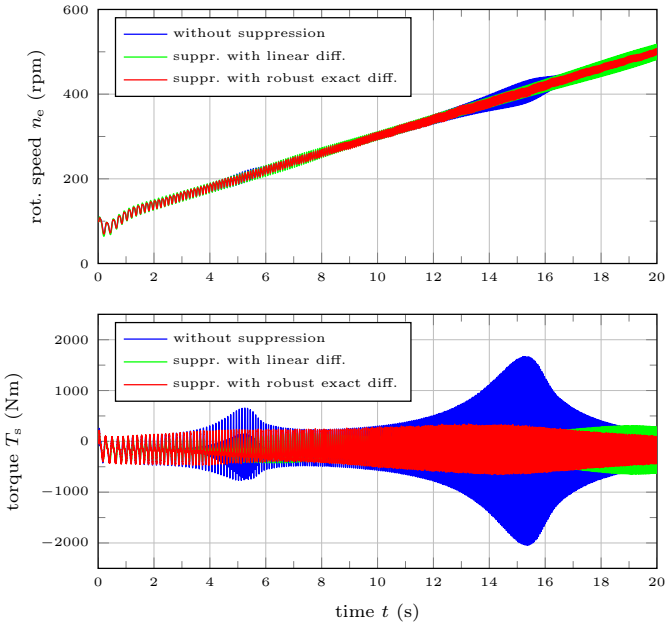


Fig. 6. Simulated rotational speed n_e and shaft torque T_s.

build-up in the electrical drive are all considered. The first resonance frequency is located at 20 Hz and is excited with the 3rd and 6th order of the engine speed. The order of the excitation depends upon the number of engine cylinders. While suppression with the linear differentiator requires a low-pass filter after the ideal differentiation to suppress noise (see Fig. 5), there is no filter necessary for the robust exact differentiator. Therefore, the phase shift due to the filtering properties of the differentiator is smaller using the robust exact differentiator.

The simulation results depicted in Fig. 6 compare the closed loop system behaviour with the robust exact differentiator and with the linear differentiator, respectively. For comparison, both variants are tuned for maximum damping. The discrete time first order transfer function

$$F_{LP}(z) = \frac{(1-p_z)\,z}{z-p_z} \tag{19}$$

with the filter pole p_z is used for low-pass filtering in the linear differentiator. It can be seen that both variants suppress the first resonance frequency. The frequency spectrum of the shaft torque T_s in Fig. 7 reveals that when using the linear differentiator, the resulting resonance frequency moves from 20 Hz to 24 Hz. The resonant frequency peak moves dependent on d_a, i.e. also the possible damping, due to the occurring phase shift of the additional filter. In contrast, the shift of the resonance frequency peak is smaller using the robust exact differentiator.

To further assess these results, the damping torque T_d is calculated from real measurement data of the shaft

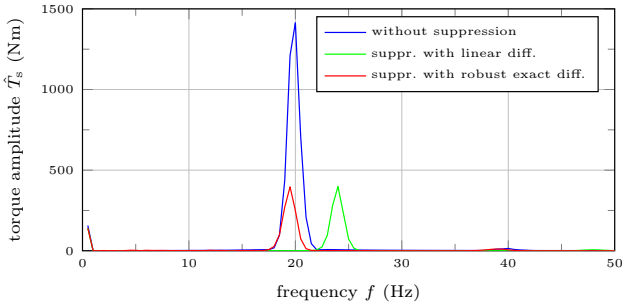


Fig. 7. FFT analysis of the shaft torque T_s.

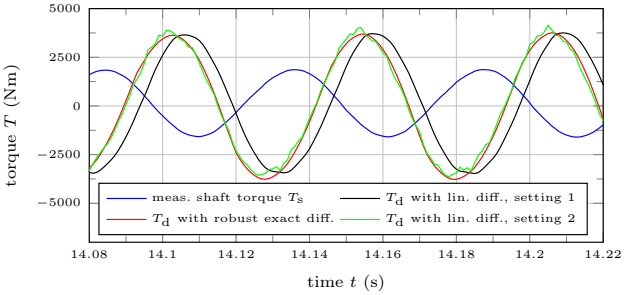


Fig. 8. Comparison of damping torque T_d calculated based on shaft torque measurement with linear differentiation and robust exact differentiator.

torque using both differentiation methods with the same parameters as in the closed-loop simulation (setting 1). The results in Fig. 8 show that the damping torque T_d calculated using the robust exact differentiator has less phase shift in comparison to T_d gained from the linear differentiator with the additional filter. Due to the filter, the phase shift increases. Additionally, the filter within the linear differentiator is tuned in such a way that both signals have the same phase (setting 2). Now it can be seen that the signal gained with setting 2 of the linear differentiator has a lower signal quality with more remaining noise and therefore cannot be used for damping. In particular, the phase shift and the noise reduce the possible damping in this application. A bigger phase shift could lead to instability of the closed loop system in the presence of dead time and increased time constants for the torque build-up. Additionally, the remaining noise following differentiation can lead to the excitation of the second resonance frequency, which also results in an unstable system. The settings used in the simulation to perform active damping are

- differentiator order: 4,
- convergence rate / robustness factor: 350,
- step size: 1 ms,
- robust exact differentiator: d_a: 0.007,
- lin. differentiator setting 1: d_a: 0.02,
- lin. differentiator setting 1: p_z: 0.8,
- lin. differentiator setting 2: d_a: 0.018,
- lin. differentiator setting 2: p_z: 0.5.

3.2 Reconstruction of measured acceleration

In this application, the acceleration of a moving body is estimated by the differentiation of the measured position with respect to time. As the laboratory model that is used

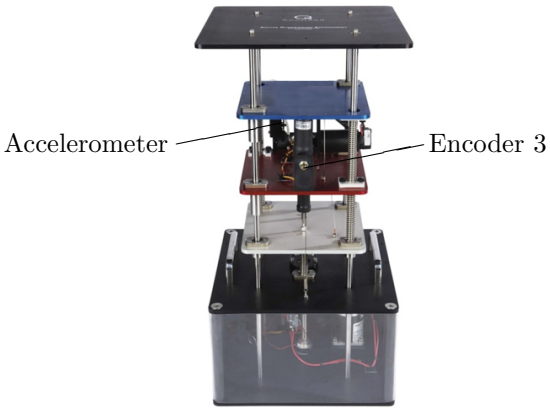


Fig. 9. Active Suspension System (Quanser (2013)).

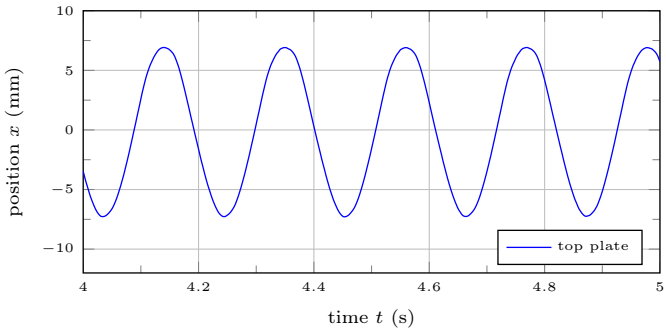


Fig. 10. Measured position signal.

includes an accelerometer, a comparison of estimated and measured acceleration is possible.

The active suspension laboratory system is depicted in Fig. 9. It consists of three masses or plates on top of each other. Each mass slides along steel shafts and is supported by a set of springs. The lower plate (silver) is driven by a DC motor via a lead screw and a gearing mechanism. This motor can be used to emulate different road profiles. The middle plate (red) is in contact with the lower plate through springs and constitutes the tyre in the quarter-car model. The top plate (blue) represents the vehicle body supported by a set of springs working as the quarter-car's suspension. Attached to the top plate is an additional DC motor that can emulate an active suspension via a capstan drive. Both motors are equipped with encoders (Enc1, Enc2), which can be used to measure angular positions. Additionally, there is a third encoder (Enc3) attached to the top plate to track the absolute position of the vehicle body. The top plate is also instrumented with an accelerometer to measure the acceleration of the vehicle body. For additional information see Quanser (2013). The accelerometer signal could be used for control; however, in this work it is utilized as a reference to assess the performance of the differentiator toolbox.

For the experiment presented in this work, the three plates were mechanically connected and the DC motor in the base of the model was used to move the plates up and down. Therefore, the motor was operated in position control mode to follow the given sinusoidal reference. The encoder, Enc3, was used to measure the position x of the top plate, this signal being presented in Fig. 10.

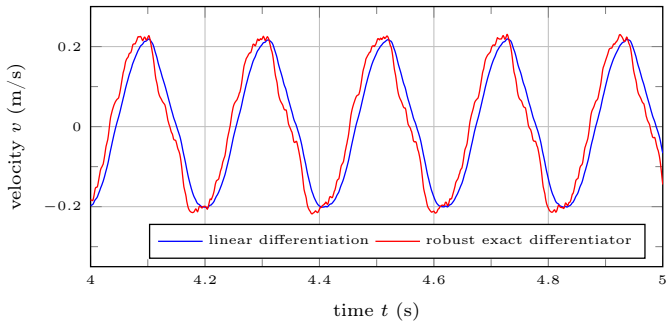


Fig. 11. Estimated velocity signals.

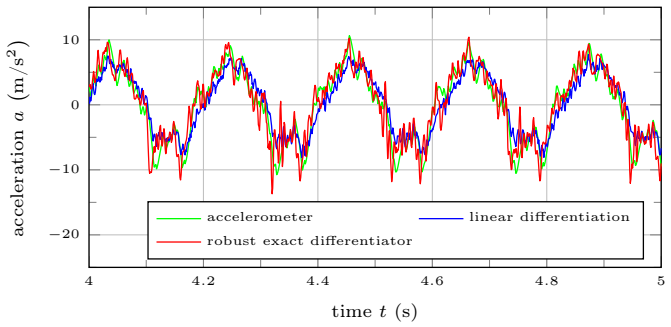


Fig. 12. Measured and estimated acceleration signals.

To estimate the plate's acceleration, the position signal x can be differentiated twice. The estimated and measured acceleration can then be compared. To better assess the results obtained by using the differentiator from the robust exact differentiator toolbox, the acceleration will also be estimated using the classical linear approach of numerical differentiation and low-pass filtering. The corresponding discrete time transfer function is given by

$$\frac{\hat{a}(z)}{x(z)} = \left(\frac{1-z^{-1}}{T_d}\right)^2 \frac{(1-p_z)\,z}{z-p_z} \tag{20}$$

where $\hat{a}$ is the estimated acceleration, T_d is the sampling interval and p_z is the pole of the discrete time low-pass filter. To estimate the plate's velocity $\hat{v}$

$$\frac{\hat{v}(z)}{x(z)} = \frac{1-z^{-1}}{T_d} \frac{(1-p_z)\,z}{z-p_z} \tag{21}$$

can be used. The results presented in the following were obtained by using a filter pole of $p_z = 0.9$ with a sampling interval of $T_d = 1\,\text{ms}$. The parameter setting for the robust exact differentiator is given by:

- differentiator order: 3,
- convergence rate / robustness factor: 13,
- step size: 1 ms.

The estimated velocities are presented in Fig. 11, while a comparison of the estimated and the measured acceleration signals is shown in Fig. 12. The settings for the robust exact differentiator (differentiator order, convergence rate) and the linear differentiator (filter pole) were tuned, so that the high frequency signal components have similar amplitudes. The estimated velocities in Fig. 11 show that with the robust exact differentiator the time lag introduced by the filtering properties of the differentiator is significantly smaller.

Table 3. Comparison of estimation errors for the signal section shown in Figs. 10–12.

type	estimation error	unit
robust exact differentiator	1.66	m/s^2
linear differentiator	1.95	m/s^2

This can also be seen in Fig. 12 where a comparison of measured and estimated acceleration clearly shows that with the robust exact differentiator the time lag between measurement and estimation can be reduced. Additionally, high order dynamic components in the acceleration signal can be reconstructed more effectively. These benefits can also be seen in the average estimation error

$$e = \frac{1}{N+1} \sum_{k=0}^{N} |a(k) - \hat{a}(k)| \tag{22}$$

where N is the number of samples considered and the results are presented in Table 3 for both differentiators.

3.3 Edge detection for image processing

In this section the differentiator toolbox is used in the context of an image processing application. The black and white logo from Graz University of Technology shown in Fig. 13a serves as the basis image for an edge detection algorithm. Among other methods the detection of the edges can be based on computation of the Laplacian,

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}, \tag{23}$$

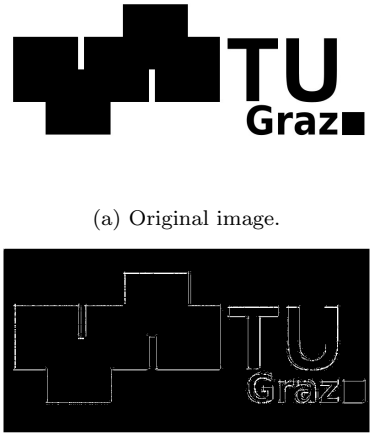
see Marr and Hildreth (1980). In equation (23), the variables x and y denote the direction of the image w.r.t the length and height respectively. The function $f(x, y)$ maps into the interval $[0, 255]$ where 0 and 255 indicate a white and a black pixel respectively.

In order to compute the required second derivatives, as a first step the rows of the image are combined into a single vector which is differentiated using the Simulink block. Then differentiation is performed with a vector generated column wise, i.e. differentiation into the y-direction of the image. According to Eq. (23) the results are added and depicted as a black and white image, see Fig. 13b. The settings of the differentiator block used are

- differentiator order: 2,
- convergence rate / robustness factor: 1.6,
- step size: 1.

4. CONCLUSION

A recently developed toolbox implementing a robust exact differentiator has been introduced. A range of problems involving both simulation studies and experimental trials have been used to illustrate the practical application of the toolbox. Whether implemented as a simulink block in simulation studies or implemented via autocode generation, the software environment has been seen to provide a very effective tool for signal differentiation across a variety of applications.



(a) Original image.

(b) Result of the edge detection.

Fig. 13. Result obtained by the differentiator based edge detection algorithm.

REFERENCES

Imine, H., Fridman, L., and Madani, T. (2015). Identification of vehicle parameters and estimation of vertical forces. *International Journal of Systems Science*, 46(16), 2996–3009. doi:10.1080/00207721.2014.886741. URL `http://dx.doi.org/10.1080/00207721.2014.886741`.

Imine, I., Fridman, L., Shraim, H., and Djemai, M. (2011). *Sliding Mode Based Analysis and Identification of Vehicle Dynamics*, volume 414 of *Lecture Notes in Control and Information Sciences*. Springer Berlin Heidelberg.

Kokal, H., Colaneri, P., del Re, L., Schmidt, M., and Paulweber, M. (2013). Feed forward disturbance rejection by a multiple FIFO approach for transient operation of an engine test bench. In *IEEE International Conference on Control Applications*, 65–70. Hyderabad, India.

Levant, A. (1998). Robust exact differentiation via sliding mode technique. *Automatica*, 34(3), 379 – 384.

Levant, A. (2003). Higher-order sliding modes, differentiation and output-feedback control. *International Journal of Control*, 76(9-10), 924–941.

Livne, M. and Levant, A. (2014). Proper discretization of homogeneous differentiators. *Automatica*, 50(8), 2007 – 2014.

Marr, D. and Hildreth, E. (1980). Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207, 187–217.

Quanser (2013). *Active Suspension System Specifications*. URL `http://www.quanser.com/Products/Docs/1844/Active_Suspension__System_Specifications.pdf`.

Reichhartinger, M. and Spurgeon, S.K. (2016). A robust exact differentiator block for matlab/simulink. *Graz University of Technology*. doi:10.13140/RG.2.1.3243.4803. URL `www.reichhartinger.at`.

Shtessel, Y., Edwards, C., Fridman, L., and Levant, A. (2014). *Sliding Mode Control and Observation*. Birkhäuser, Springer Science+Business Media, New York.

Wipfler, M., Bauer, R., Dourdoumas, N., and Rossegger, W. (2016). Regelungstechnische Methoden zur Drehschwingungsdämpfung eines Wellenstrangs am Beispiel eines Motorprüfstands. *e & i Elektrotechnik und Informationstechnik*, 133(2), 142–152.