

# Estimation of Physical Parameters Using a New Discrete-time Derivative Algorithm

Davide Tebaldi \*, Riccardo Morselli \*\*, Roberto Zanasi \*\*\*

\* *DIEF, Department of Engineering “Enzo Ferrari”  
University of Modena and Reggio Emilia, Modena, Italy  
(e-mail: davide.tebaldi@unimore.it).*

\*\* *DANA Off-Highway Engineering, Reggio Emilia, Italy  
(e-mail: riccardo.morselli@dana.com)*

\*\*\* *DIEF, Department of Engineering “Enzo Ferrari”  
University of Modena and Reggio Emilia, Modena, Italy  
(e-mail: roberto.zanasi@unimore.it).*

---

**Abstract:** The paper presents a parameters estimation procedure for physical systems modeled using the POG (Power-Oriented Graphs) technique. The coefficients defining the constitutive relation for both static and dynamic physical elements within the system can be estimated, as well as the coefficients describing energy conversions taking place either within the same energetic domain or between two different energetic domains. The evolution of the state vector over time is supposed to be known, whereas its first derivative is supposed to be unknown and is obtained by using a new algorithm for computing the discrete-time derivative of a sampled signal, which is effective even in presence of disturbances affecting the signal samples. As long as the unknown parameters appear linearly within the system differential equations, the system is allowed to exhibit any nonlinear function of the state vector and its first derivative. The procedure is finally applied to two different case studies: a linear one and a nonlinear one.

*Keywords:* Dynamic modelling, Dynamic systems, Signal processing, Parameter estimation, Linear systems, Nonlinear systems.

---

## 1. INTRODUCTION

The adoption of model-based approaches is increasingly common when dealing with mechatronic physical systems, as it enables a deep knowledge and permits a more effective control of the system under consideration. Immediately after the system modeling, the estimation of the unknown parameters describing the system behavior needs to be undertaken, in order to make the model suitable for performing reliable simulations of the system behavior. For this purpose, several approaches are presented in the literature; e.g. in Dongyu et al. (2005), where the authors study the application of a genetic algorithm to power systems, and in Aso et al. (2002), where the maximum likelihood method is applied to sector cell systems. Different versions and applications of the least squares algorithm are shown in Chen et al. (2011), Huibo and Jiangbo (2017), Li et al. (2018) and Tan et al. (2019). In this paper, a procedure for performing physical parameters estimation, exploiting the POG technique as a tool for modeling physical systems, see Zanasi (2010), is presented. The following hypotheses are made:

- The system differential equations are known;
- The system differential equations are linear with respect to the unknown parameters;
- The evolution of the state vector over time is known;

Note that the first hypothesis is always verified, as the modeling of the system is easily and effectively carried out

using the POG technique as a first step, see Zanasi and Grossi (2009a), Fei et al. (2011), Zanasi and Tebaldi (2019a) and Tebaldi and Zanasi (2019) for examples of applications of this technique to the modeling of complex physical systems in the automotive and agricultural fields. The second hypothesis poses a constraint on the appearance of the unknown parameters in the system, but does not set any constraint regarding the state vector and its first derivative, whose components can also appear nonlinearly within the system differential equations. The third hypothesis requires the knowledge of the state vector time behavior, whereas the knowledge of the first derivative of the state vector components is not required, as a very effective computation of the latter can be performed thanks to the proposed algorithm, see Sec. 2 and App. A. Other solutions are present in the literature concerning the computation of the discrete-time derivative of a sampled signal, see Ridha et al. (1997) and Hermanowicz (2001). The algorithm proposed in this paper iteratively computes the filtered  $i$ -th sample of the considered signal and its first derivative on the basis of the previous and next samples, namely  $(i - 1)$ -th and  $(i + 1)$ -th respectively.

This paper is structured as follows: Sec. 2 provides a theoretical explanation of the proposed algorithm for computing the discrete-time derivative of a sampled signal, as well as the comparison of the results it provides with those given by the Simulink discrete derivative block. The estimation procedure is then applied to two different case

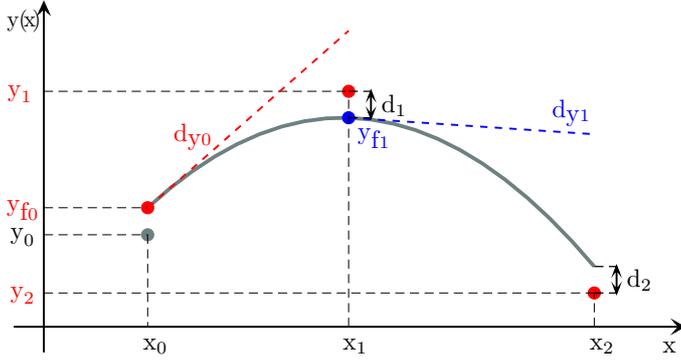


Fig. 1. Graphical interpolation of points  $(x_i, y_i)$ .

studies: a linear system composed of a DC electric motor driving an hydraulic pump, see Sec. 3, and a nonlinear system composed of a Permanent Magnet Synchronous Motor driven by a control and subject to an external load torque, see Sec. 4. Finally, the conclusions of this work are given in Sec. 5.

## 2. DISCRETE-TIME DERIVATIVE ALGORITHM

Let us consider a discrete-time signal  $(\mathbf{x}_k, \mathbf{y}_k)$  and let  $(x_i, y_i)$  denote the samples of such signal, where  $x_i$  is the  $i$ -th time sample,  $y_i$  is the  $i$ -th signal sample and  $i = \{0, 1, \dots, N\}$ , being  $N$  the number of samples. The first three input points are graphically represented in Fig. 1 for the sake of clearness.

The proposed algorithm is based on the idea of iteratively solving the following problem: given  $y_{f0}$ ,  $dy_0$ ,  $(x_1, y_1)$  and  $(x_2, y_2)$ , and using a parabolic interpolation, compute  $y_{f1}$  and  $dy_1$  in order to minimize  $d_1^2 + d_2^2$ . Let us consider the following parabola:

$$y(x) = ax^2 + bx + c \quad (1)$$

and compute the parameters  $a$ ,  $b$  and  $c$  such that:

$$y(x_0) = y_{f0}, \quad \left. \frac{dy(x)}{dx} \right|_{x=x_0} = dy_0, \quad \min_a (d_1^2 + d_2^2) \quad (2)$$

The first two constraints provide the following relations:

$$\begin{cases} ax_0^2 + bx_0 + c = y_{f0} \\ 2ax_0 + b = dy_0 \end{cases} \Rightarrow \begin{cases} b = dy_0 - 2ax_0 \\ c = y_{f0} - ax_0^2 - bx_0 \end{cases} \quad (3)$$

Distance  $d_1$  in (2) can be expressed as follows:

$$d_1 = |ax_1^2 + bx_1 + c - y_1| \quad (4)$$

Substituting (3) in (4), one obtains:

$$d_1(a) = |a(x_1 - x_0)^2 + y_{f0} + dy_0(x_1 - x_0) - y_1|$$

Similarly, from (3) and (4), one obtains:

$$d_2(a) = |a(x_2 - x_0)^2 + y_{f0} + dy_0(x_2 - x_0) - y_2|$$

The minimum of  $f(a) = d_1^2(a) + d_2^2(a)$  occurs when  $\frac{df(a)}{da} = 0$ :

$$\begin{aligned} \frac{df(a)}{da} = & 2(a(x_1 - x_0)^2 + y_{f0} + dy_0(x_1 - x_0) - y_1)(x_1 - x_0)^2 + \\ & + 2(a(x_2 - x_0)^2 + y_{f0} + dy_0(x_2 - x_0) - y_2)(x_2 - x_0)^2 = 0 \end{aligned}$$

By solving with respect to  $a$ , it results:

$$a = -\frac{a' + a''}{(x_1 - x_0)^4 + (x_2 - x_0)^4} \quad (5)$$

where

$$\begin{cases} a' = (y_{f0} + dy_0(x_1 - x_0) - y_1)(x_1 - x_0)^2 \\ a'' = (y_{f0} + dy_0(x_2 - x_0) - y_2)(x_2 - x_0)^2 \end{cases} \quad (6)$$

By substituting (5) and (6) in (3), the values of parameters  $b$  and  $c$  can then be derived.

By replacing parameters  $b$  and  $c$  in (1), one derives a parabola that can be used to determine the filtered version  $y_{f1}$  of the original data sample at  $x = x_1$ :

$$\begin{aligned} y_{f1} &= ax_1^2 + bx_1 + c \\ &= ax_1^2 + bx_1 + y_{f0} - ax_0^2 - bx_0 \\ &= a(x_1^2 - x_0^2) + b(x_1 - x_0) + y_{f0} \\ &= a(x_1^2 - x_0^2) + (dy_0 - 2ax_0)(x_1 - x_0) + y_{f0} \\ &= y_{f0} + dy_0(x_1 - x_0) + a(x_1 - x_0)^2 \end{aligned} \quad (7)$$

Similarly, the derivative  $dy_1$  at  $x = x_1$  can be determined:

$$dy_1 = 2ax_1 + b = 2ax_1 + dy_0 - 2ax_0 = dy_0 + 2a(x_1 - x_0) \quad (8)$$

If the samples are equally spaced, namely satisfying:

$$T = (x_1 - x_0) = (x_2 - x_1) = (x_3 - x_2) = \dots$$

the solution in (5) and (6) simplifies as follows:

$$a = -\frac{(y_{f0} + dy_0T - y_1) + 4(y_{f0} + 2dy_0T - y_2)}{17T^2}$$

whereas parameters  $dy_1$  and  $y_{f1}$ , defined in (8) and (7) respectively, can be expressed in the following way:

$$dy_1 = dy_0 + 2aT, \quad y_{f1} = y_{f0} + dy_0T + aT^2$$

The algorithm described in this section has been implemented in the Matlab function “[dz, zf]=DX\_DT(t, z)”, whose internal code is provided in App. A. Note that the code reported in the function already accounts for the case of non-equally spaced samples, i.e. the general solution described by (5), (6), (7) and (8) is implemented.

### 2.1 Application of the algorithm

This section shows the application of the algorithm described in Sec. 2 to the computation of the discrete-time derivative of a sampled signal.

Let us consider the following continuous-time signal:

$$x(t) = \sum_{i=1}^5 a_i \sin(2\pi f_i t), \quad \begin{cases} a_i = [50 \ 73 \ 33 \ 12 \ 96] \\ f_i = [22 \ 87 \ 94 \ 61 \ 46] \end{cases} \quad (9)$$

whose exact time derivative is given by:

$$\frac{dx(t)}{dt} = \dot{x}(t) = \sum_{i=1}^5 2\pi a_i f_i \cos(2\pi f_i t) \quad (10)$$

Let us assume that signal  $x(t)$  in (9) is sampled with sampling time  $T$  over a time interval  $t$ , being  $t \in [0, 0.025]$  [s] and  $T = 10^{-4}$  [s]. The resulting sampled signal  $\mathbf{x}(kT) = x(t)|_{t=kT}$  is then derived by using the following tools: the Simulink discrete derivative block and the Matlab function in App. A implementing the proposed algorithm. For the latter,  $y_{f0}$  and  $dy_0$  introduced in Sec. 2 are initialized as shown in App. A, namely to the first sample value and to the difference quotient between the second and first sample values, respectively:

$$y_{f0} = y_0, \quad dy_0 = \frac{y_1 - y_0}{x_1 - x_0}.$$

Note that, since the sampling time  $T$  is constant, the current case is that of equally spaced samples. For this

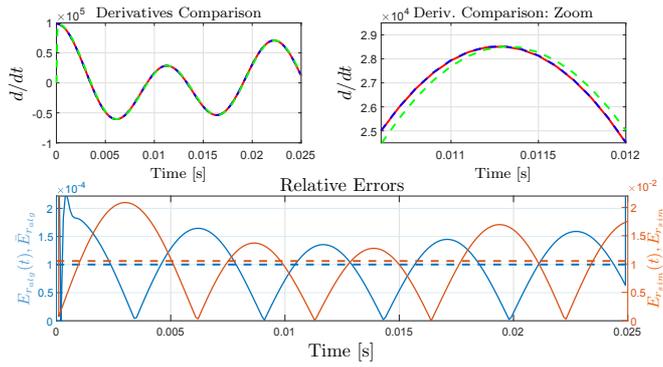


Fig. 2. Derivatives comparison on the original signal.

reason, the simulation for computing the discrete-time derivative using the Simulink discrete derivative block is performed using a fixed-step size solver with  $T = 10^{-4}$  [s].

The results of the computations are shown in Fig. 2, whose legend is depicted in the following. The top-left subplot shows the exact time derivative of the signal reported in (10) (red line), the time derivative of the sampled signal computed by using the Simulink discrete derivative block (green dashed line) and the time derivative of the sampled signal computed by the proposed algorithm (blue dashed line). The top-right subplot shows a zoomed version of the same pieces of information as the top-left subplot using the same color notation, from which the reader can clearly see the exactness of the derivative computation performed by the proposed algorithm as well as the error which is made instead by the Simulink discrete derivative block. The bottom subplot shows the relative error  $E_{r_{sim}}(t)$  between the exact derivative in (10) and the computation by the Simulink discrete derivative block over time normalized with respect to the maximum value of the exact derivative (orange line), and its mean value  $\bar{E}_{r_{sim}}$  (orange dashed line) with reference to the right vertical axis. Additionally, the bottom subplot of Fig. 2 shows the relative error  $E_{r_{alg}}(t)$  between the exact derivative in (10) and the computation by the proposed algorithm over time normalized with respect to the maximum value of the exact derivative (light blue line) and its mean value  $\bar{E}_{r_{alg}}$  (light blue dashed line) with reference to the left vertical axis. The better results given by the proposed algorithm with respect to the Simulink discrete derivative block can be appreciated, since the ratio between the mean relative errors  $\bar{E}_{r_{alg}}/\bar{E}_{r_{sim}}$  is equal to 0.0095.

Let us now assume that the original signal  $x(t)$  in (9) is affected by a random disturbance  $d(t)$  whose maximum amplitude is eight times lower than the maximum amplitude of signal  $x(t)$ . The result is shown in the top-left subplot of Fig. 3, where the blue dashed line is the original signal  $x(t)$ , the green line is the disturbed signal  $x_d(t) = x(t) + d(t)$  and the red line is the filtered signal  $x_f(t)$  given by the proposed algorithm using Eq. (7). The same derivation tools as before are applied to the disturbed signal  $x_d(t)$  and the results are shown in the top-right subplot of Fig. 3 in a zoomed time interval, to better appreciate the results. The blue dashed line is the exact derivative of the original signal in (10), the cyan line is the time derivative of the disturbed sampled signal computed by using the Simulink discrete derivative block and the magenta line is the time

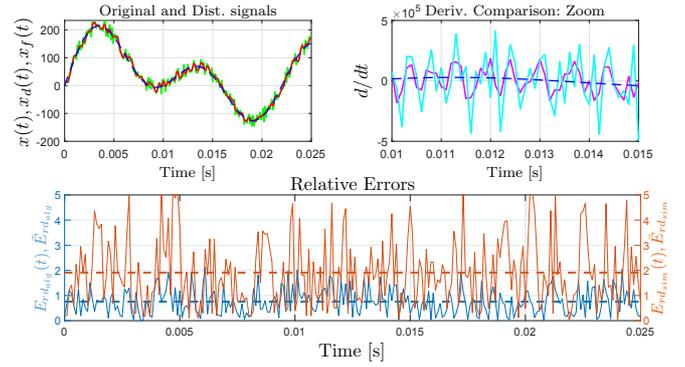


Fig. 3. Derivatives comparison on the disturbed signal.

derivative of the disturbed sampled signal computed by the proposed algorithm using Eq. (8). The bottom subplot shows the relative error  $E_{r_{dsim}}(t)$  between the exact derivative in (10) of the original signal and the derivative of the disturbed sampled signal computed by the Simulink discrete derivative block over time normalized with respect to the maximum value of the exact derivative (orange line) and its mean value  $\bar{E}_{r_{dsim}}$  (orange dashed line) with reference to the right vertical axis. Additionally, the bottom subplot of Fig. 3 shows the relative error  $E_{r_{dalg}}(t)$  between the exact derivative in (10) of the original signal and the derivative of the disturbed sampled signal computed by the proposed algorithm over time normalized with respect to the maximum value of the exact derivative (light blue line) and its mean value  $\bar{E}_{r_{dalg}}$  (light blue dashed line) with reference to the left vertical axis. Note that the ratio between the mean relative errors  $\bar{E}_{r_{dalg}}/\bar{E}_{r_{dsim}}$  is equal to 41.95. From this latter ratio and from Fig. 3, the reader can appreciate how the proposed algorithm works better than the Simulink discrete derivative block even in presence of a random disturbance affecting the signal. This result is achieved thanks to the fact that, at each  $i$ -th time sample  $x_i$ , the algorithm proposed in Sec. 2 computes the derivative  $dy_i$  of the *preliminarily filtered version*  $y_{fi}$  of the original signal sample  $y_i$ , where the preliminary filtering is performed thanks to a parabolic interpolation, see Eq. (7).

### 3. A DC MOTOR DRIVING AN HYDRAULIC PUMP

The first case study consists of a DC electric motor connected to an hydraulic pump, as shown in Fig. 4. This system involves three different energetic domains: electrical, mechanical rotational and hydraulic. The corresponding POG graphical representation is shown in Fig. 5: the power sections present in the POG scheme have a direct correspondence with the real physical sections. Let  $\mathbf{x} = [I_a \ \omega_m \ P_0]^T$  be the state vector of the system, i.e. the output variables of the dynamic elements. The state space dynamic model can be obtained by direct inspection of the POG scheme, see Zanasi (2010):

$$\underbrace{\begin{bmatrix} L_a & 0 & 0 \\ 0 & J_m & 0 \\ 0 & 0 & C_0 \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \dot{I}_a \\ \dot{\omega}_m \\ \dot{P}_0 \end{bmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{bmatrix} -R_a & -K_m & 0 \\ K_m & -b_m & -K_p \\ 0 & K_p & -\alpha_p \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} I_a \\ \omega_m \\ P_0 \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} V_a \\ Q_0 \end{bmatrix}}_{\mathbf{u}} \quad (11)$$

where  $\mathbf{u}$  is the input vector and  $\mathbf{L}$ ,  $\mathbf{A}$  and  $\mathbf{B}$  are the energy, power and input matrices, respectively, see Zanasi (2010). The meaning of the parameters and variables in system (11) is described in the following with reference to Fig. 4 and Fig. 5. Parameters  $L_a$ ,  $J_m$  and  $C_0$  are the

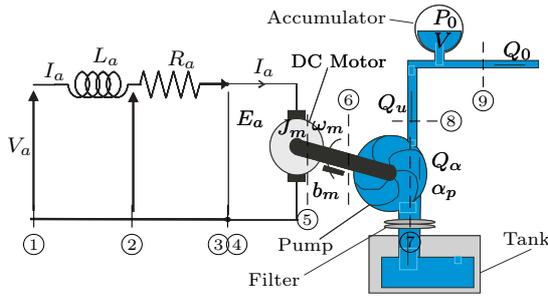


Fig. 4. A DC motor connected to an hydraulic pump.

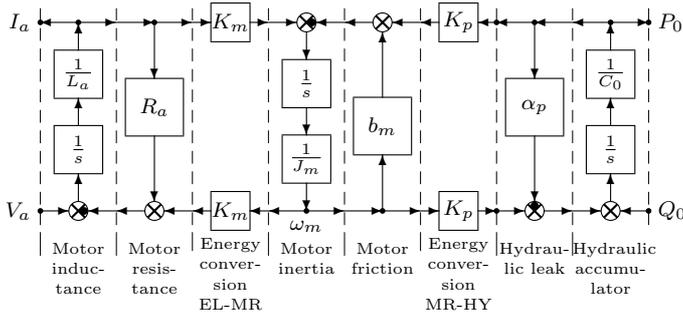


Fig. 5. POG scheme of the DC motor with hydraulic pump.

motor inductance, the rotor inertia and the capacitance of the hydraulic accumulator, respectively. Parameters  $R_a$ ,  $b_m$  and  $\alpha_p$  represent the motor electric resistance, the rotor friction coefficient and the hydraulic leak, respectively. Terms  $K_m$  and  $K_p$  describe the energy conversion taking place between electrical/mechanical rotational and between mechanical rotational/hydraulic energetic domains, respectively. The system state variables are the motor armature current  $I_a$ , the rotor angular speed  $\omega_m$  and the pressure  $P_0$  within the hydraulic accumulator; the system inputs are the motor voltage  $V_a$  and volume flow rate  $Q_0$ .

System (11) can be rewritten as follows:

$$\underbrace{\begin{bmatrix} \dot{I}_a & 0 & 0 & I_a & \omega_m & 0 & 0 & 0 \\ 0 & \dot{\omega}_m & 0 & 0 & -I_a & \omega_m & P_0 & 0 \\ 0 & 0 & \dot{P}_0 & 0 & 0 & 0 & -\omega_m & P_0 \end{bmatrix}}_{\Phi(t)} \underbrace{\begin{bmatrix} L_a \\ J_m \\ C_0 \\ R_a \\ K_m \\ b_m \\ K_p \\ \alpha_p \end{bmatrix}}_{\mathbf{p}} = \underbrace{\begin{bmatrix} V_a \\ 0 \\ Q_0 \end{bmatrix}}_{\mathbf{q}(t)} \quad (12)$$

where  $\mathbf{p}$  is the vector of the unknown parameters. System (12) satisfies the first two hypotheses introduced in Sec. 1, as the system equations are known and linear with respect to the components of vector  $\mathbf{p}$ . Additionally, the availability of experimental observations of the evolution of the state vector  $\mathbf{x}$  over time, in response to the input vector  $\mathbf{u}$ , makes the third hypothesis satisfied as well.

Matrix  $\Phi(t) = \Phi(\mathbf{x}(t), \dot{\mathbf{x}}(t))$  in system (12) is a function of vectors  $\mathbf{x}(t)$  and  $\dot{\mathbf{x}}(t)$ , whereas vector  $\mathbf{q}(t) = \mathbf{q}(\mathbf{u}(t))$  is function of the input vector  $\mathbf{u}(t)$ . Matrix  $\Phi(t)$  and vector  $\mathbf{q}(t)$  are known if vector  $\mathbf{x}(t)$ ,  $\dot{\mathbf{x}}(t)$  and  $\mathbf{u}(t)$  are known.

Let us suppose that vectors  $\mathbf{x}(t)$  and  $\mathbf{u}(t)$  are known for  $t = kT$ , where  $T$  is the sampling period and  $k \in [0, 1, 2, \dots, N]$ :  $\mathbf{x}(t) \rightarrow \mathbf{x}(kT) = \mathbf{x}_k$ ,  $\mathbf{u}(t) \rightarrow \mathbf{u}(kT) = \mathbf{u}_k$ .

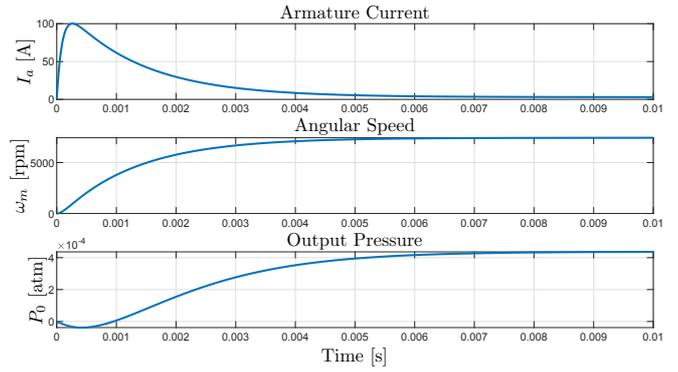


Fig. 6. Measured forced evolution of  $\mathbf{x}_k$  for system in Fig. 5.

Vector  $\dot{\mathbf{x}}_k$  can be numerically obtained from vector  $\mathbf{x}_k$  by applying the discrete-time derivative algorithm in Sec. 2:

$$\dot{\mathbf{x}}(t) \rightarrow \dot{\mathbf{x}}(t)|_{t=kT} \simeq \frac{\partial \mathbf{x}_k}{\partial(kT)} = \dot{\mathbf{x}}_k.$$

From  $\mathbf{x}_k$ ,  $\dot{\mathbf{x}}_k$  and  $\mathbf{u}_k$ ,  $\Phi(kT)$  and  $\mathbf{q}(kT)$  are known:

$$\Phi(kT) = \Phi(\mathbf{x}(t), \dot{\mathbf{x}}(t))|_{t=kT} \simeq \Phi(\mathbf{x}_k, \dot{\mathbf{x}}_k)$$

$$\mathbf{q}(kT) = \mathbf{q}(\mathbf{u}(t))|_{t=kT} = \mathbf{q}(\mathbf{u}(kT)) = \mathbf{q}(\mathbf{u}_k)$$

Considering the sampling instants  $t = kT$ , for  $k \in [0, 1, 2, \dots, N]$ , system (12) can be rewritten as follows:

$$\underbrace{\begin{bmatrix} \Phi(0) \\ \Phi(T) \\ \Phi(2T) \\ \vdots \\ \Phi(NT) \end{bmatrix}}_{\Phi} \mathbf{p} = \underbrace{\begin{bmatrix} \mathbf{q}(0) \\ \mathbf{q}(T) \\ \mathbf{q}(2T) \\ \vdots \\ \mathbf{q}(NT) \end{bmatrix}}_{\mathbf{q}} \quad \Leftrightarrow \quad \Phi \mathbf{p} = \mathbf{q}.$$

and solved by finding the vector  $\mathbf{p}$  minimizing the Euclidean norm  $\|\mathbf{e}\|_2$  of the error vector  $\mathbf{e} = \Phi \mathbf{p} - \mathbf{q}$ , namely:

$$\mathbf{p} = \underbrace{(\Phi^T \Phi)^{-1} \Phi^T}_{\Phi^{-\dagger}} \mathbf{q}. \quad \Leftrightarrow \quad \mathbf{p} = \Phi^{-\dagger} \mathbf{q} \quad (13)$$

where  $\Phi^{-\dagger}$  is the pseudo-inverse of matrix  $\Phi$ .

### 3.1 DC motor with hydraulic pump: Parameters Estimation

Let us suppose that the forced evolution of the components of the state vector  $\mathbf{x} = [I_a \ \omega_m \ P_0]^T$  acquired from experimental measurements are those shown in Fig. 6. The latter figure is referred to as “forced evolution” because the system initial conditions are set to zero:  $I_{a0} = 0$  [A],  $\omega_{m0} = 0$  [rpm] and  $P_{00} = 0$  [atm], whereas the system inputs are constant and equal to  $V_a = 400$  [V] and  $Q_0 = 0.18$  [dm<sup>2</sup>/s].

By applying the least square algorithm resolutive formula (13), and exploiting the algorithm presented in Sec. 2 for computing the discrete-time derivative of the forced evolution, the components of vector  $\mathbf{p}$  in (12) containing the system unknown parameters are estimated.

A comparison between the estimated and actual system parameters values is reported in Table 1, from which it is possible to appreciate the very good matching resulting from the estimation, since the error between the actual and estimated parameters is very low, as reported in the last column of Table 1.

Table 1. Estimation results for system in Fig. 5.

	Estimated Values	Actual Values	Error
$L_a$	0.00030056	0.0003	$5.5823 \cdot 10^{-7}$
$J_m$	$9.9999 \cdot 10^{-5}$	0.0001	$8.1943 \cdot 10^{-10}$
$C_0$	$0.9505 \cdot 10^{-7}$	$1 \cdot 10^{-7}$	$4.9478 \cdot 10^{-9}$
$R_a$	3.5	3.5	$1.0699 \cdot 10^{-5}$
$K_m$	0.50001	0.5	$5.2025 \cdot 10^{-6}$
$b_m$	0.00191	0.0019099	$1.4551 \cdot 10^{-7}$
$K_p$	$7.7704 \cdot 10^{-6}$	$8 \cdot 10^{-6}$	$2.2959 \cdot 10^{-7}$
$a_p$	$9.5987 \cdot 10^{-5}$	0.0001	$4.0129 \cdot 10^{-6}$

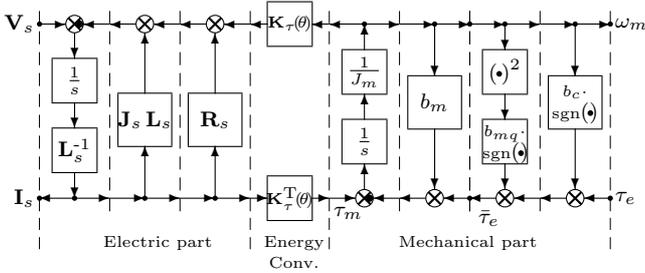


Fig. 7. POG scheme of the PMSM.

#### 4. PERMANENT MAGNET SYNCHRONOUS MOTOR

The second case study consists of a Permanent Magnet Synchronous Motor (PMSM). The dynamics of this system is described by the POG block scheme in Fig. 7 in a transformed rotating d-q frame  $\Sigma_\omega$ , that is a reference frame which rotates together with the rotation of the rotor, see Fei et al. (2011). Under the hypothesis of sinusoidal rotor flux and star-connected stator phases, the state space model of the system is the following one:

$$\begin{bmatrix} pL_{se} & 0 & 0 \\ 0 & pL_{se} & 0 \\ 0 & 0 & J_m \end{bmatrix} \begin{bmatrix} \dot{I}_d \\ \dot{I}_q \\ \dot{\omega}_m \end{bmatrix} = - \begin{bmatrix} pR_s & -p^2\omega_m L_{se} & 0 \\ p^2\omega_m L_{se} & pR_s & K_q \\ 0 & -K_q & b_m \end{bmatrix} \begin{bmatrix} I_d \\ I_q \\ \omega_m \end{bmatrix} + \begin{bmatrix} V_d \\ V_q \\ -\tau_e \end{bmatrix} \quad (14)$$

where

$$\tau_e = \tau_e + b_c \operatorname{sgn}(\omega_m) + b_{mq} \operatorname{sgn}(\omega_m) \omega_m^2. \quad (15)$$

The meaning of the system parameters and variables in (14) and (15) can be found in Fei et al. (2011), Zanasi and Tebaldi (2019b) and Zanasi and Tebaldi (2019c).

System (14) can be rewritten as follows:

$$\underbrace{\begin{bmatrix} pI_d - p^2\omega_m I_q & pI_d & 0 & 0 & 0 & 0 & 0 \\ pI_q + p^2\omega_m I_d & pI_q & \omega_m & 0 & 0 & 0 & 0 \\ 0 & 0 & -I_q & \dot{\omega}_m & \omega_m & \operatorname{sgn}(\omega_m) & \operatorname{sgn}(\omega_m)\omega_m^2 \end{bmatrix}}_{\Phi(t)} \underbrace{\begin{bmatrix} L_{se} \\ R_s \\ K_q \\ J_m \\ b_m \\ b_c \\ b_{mq} \end{bmatrix}}_{\mathbf{p}} = \underbrace{\begin{bmatrix} V_d \\ V_q \\ -\tau_e \end{bmatrix}}_{\mathbf{q}(t)}$$

This system satisfies the first two hypotheses introduced in Sec. 1: the system equations are known and are linear with respect to the components of vector  $\mathbf{p}$ . Furthermore, thanks to the experimental observations of the evolution of the state vector  $\mathbf{x}$ , the third hypothesis is satisfied too.

##### 4.1 PMSM: Parameters Estimation

Let us assume that a speed control is applied to the considered PMSM, making the motor speed  $\omega_m$  follow a step speed profile  $\omega_{m_{des}}$ , see the blue dashed line in the top subplot of Fig. 9. Let us also assume that the operating

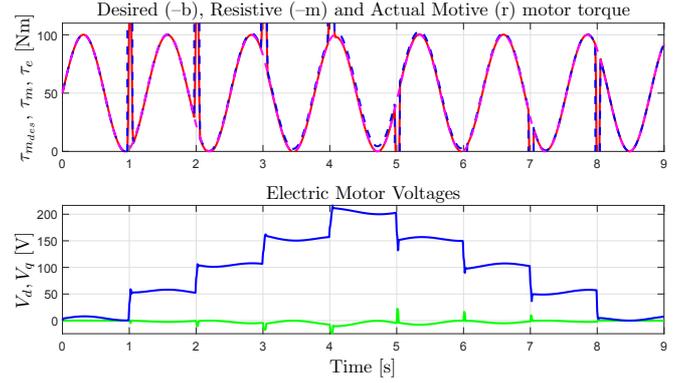


Fig. 8. Motor torques and voltages.

conditions are such as to make the motor subject to a sinusoidal load torque  $\tau_e$ . The desired motive torque  $\tau_{m_{des}}$  minimizing the speed error  $\Delta\omega_m = \omega_{m_{des}} - \omega_m$  is computed using the following control law in the Laplace domain:

$$\tau_{m_{des}} = \left( \frac{K_P + K_I s}{K_I s} \right) \Delta\omega_m \quad (16)$$

being  $K_P$  and  $K_I$  two design parameters properly chosen and  $s$  the complex variable. The desired torque  $\tau_{m_{des}}$  is turned into an input voltage vector  $\mathbf{V}_s$  driving the motor through a vectorial control, see Zanasi and Grossi (2009b).

Let us suppose that the experimental observations are those shown in Fig. 8 and Fig. 9. The upper subplot of Fig. 8 shows the sinusoidal load torque  $\tau_e$  applied to the motor (magenta dashed line), the desired motive torque  $\tau_{m_{des}}$  demanded by (16) (blue dashed line) and the generated motive torque  $\tau_m$  (red line), see Fig. 7. The components  $V_d$  and  $V_q$  of voltage vector  $\mathbf{V}_s$  imposed by the vectorial control as inputs for the system are shown in the lower subplot of Fig. 8 in green and blue line, respectively. The forced evolution for the components of the state vector  $\mathbf{x} = [I_d \ I_q \ \omega_m]^T$ , see (14), acquired from experimental measurements is shown in Fig. 9. The upper subplot shows the desired step speed profile  $\omega_{m_{des}}$  (blue dashed line) and the actual motor speed  $\omega_m$  (red line). The lower subplot of Fig. 9 shows the components  $I_d$  and  $I_q$  of current vector  $\mathbf{I}_s$  in green and blue line, respectively. The latter figure is referred to as ‘‘forced evolution’’ because the system initial conditions are set to zero:  $\mathbf{I}_{s_0} = [0 \ 0]^T$  [A] and  $\omega_{m_0} = 0$  [rpm]. The very good matching resulting from the estimation can be appreciated in Table 2, showing the very low error between the actual and estimated system parameters.

Table 2. Estimation results for system in Fig. 7.

	Estimated Values	Actual Values	Error
$L_{se}$	$2.8426 \cdot 10^{-5}$	$2.84 \cdot 10^{-5}$	$2.5606 \cdot 10^{-8}$
$R_s$	0.02	0.02	$3.4427 \cdot 10^{-7}$
$K_q$	0.95	0.95	$3.6194 \cdot 10^{-6}$
$J_m$	0.1184	0.1184	$3.2935 \cdot 10^{-6}$
$b_m$	0.00099531	0.001	$4.6904 \cdot 10^{-6}$
$b_c$	$6.4539 \cdot 10^{-5}$	0	$6.4539 \cdot 10^{-5}$
$b_{mq}$	0.00010002	0.0001	$2.0756 \cdot 10^{-8}$

## 5. CONCLUSIONS

In this paper, a parameters estimation procedure suitable for physical systems modeled using the POG technique has

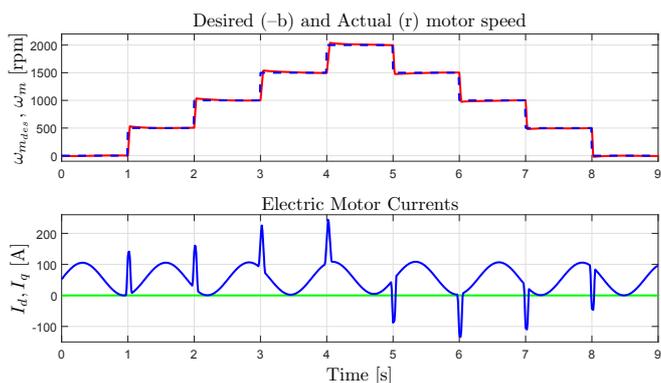


Fig. 9. Motor speeds and currents.

been presented. The procedure is based on the exploitation of a new algorithm for computing the discrete-time derivative of a sampled signal, allowing to compute the discrete-time derivative very effectively even in presence of disturbances affecting the signal samples, thanks to the preliminary filtering action it performs. The joint use of the least-squares method and the proposed algorithm for computing the first derivative of the state vector evolution over time has been applied to two different case studies, a linear one and a nonlinear one, showing the effectiveness of the presented approach.

#### REFERENCES

M. ASO, T. SAIKAWA, T. HATTORI. Mobile Station Location Estimation Using The Maximum Likelihood Method in Sector Cell Systems. Proceedings of the IEEE 56th Vehicular Technology Conference, 24-28 Sept. 2002.

J. Chen, Y. Zhang, F. Ding. Least Squares Based Iterative Parameter Estimation for Output Nonlinear Systems with Piece-Wise Nonlinearities. Proceedings of the 30th Chinese Control Conference, 22-24 July 2011.

S. Dongyu, X. Xiaorong, T. Luyuan. Studies on Parameter Identification Using Genetic Algorithm in Power System. Proceedings of the IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific, 18-18 Aug. 2005.

M. Fei, R. Zanasi, F. Grossi. Modeling of Multi-phase Permanent Magnet Synchronous Motors under Open-phase Fault Condition. Proceedings of the IEEE International Conference on Control and Automation (ICCA), 19-21 Dec. 2011.

E. Hermanowicz. ESTIMATION OF SAMPLES OF THE DERIVATIVE OF A SIGNAL BASED ON THE SIGNAL DIFFERENCE DERIVATIVE. Proceedings of the 8th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2-5 Sept. 2001.

C. Huibo, F. Jiangbo. Decomposition based recursive least squares parameter estimation for input nonlinear equation-error systems. Proceedings of the 36th Chinese Control Conference, 26-28 July 2017.

R. Li, Z. Wang, J. Yu, Y. Lei, Y. Zhang, J. Je. Dynamic Parameter Identification of Mathematical Model of Lithium-Ion Battery Based on Least Square Method. Proceedings of the IEEE International Power Electronics and Application Conference and Exposition (PEAC), 4-7 Nov. 2018.

H. Ridha, J. Vesma, T. Saramäki, M. Renfors. DERIVATIVE APPROXIMATIONS FOR SAMPLED SIGNALS BASED ON POLYNOMIAL INTERPOLATION. Proceedings of the 13th International Conference on Digital Signal Processing, 2-4 July 1997.

Z. Tan, H. Zhang, J. Sun, K. Du. Research on Identification Process of Nonlinear System Based on An Improved Recursive Least Squares Algorithm. Proceedings of the Chinese Control And Decision Conference (CCDC), 3-5 June 2019.

D. Tebaldi, R. Zanasi. Modeling and Control of a Power-Split Hybrid Propulsion System. Proceedings of the IEEE 45th Annual Conference of the Industrial Electronics Society (IECON), 14-17 Oct. 2019.

R. Zanasi. The Power-Oriented Graphs technique: System modeling and basic properties. Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC), 1-3 Sept. 2010.

R. Zanasi, F. Grossi (a). The POG Technique for Modeling Planetary Gears and Hybrid Automotive Systems. Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC), 7-10 Sept. 2009.

R. Zanasi, F. Grossi (b). Vectorial Control of Multi-phase Synchronous Motors using POG Approach. Proceedings of the IEEE 35th Annual Conference of the Industrial Electronics Society (IECON), 3-5 Nov. 2009.

R. Zanasi, D. Tebaldi (a). Planetary Gear Modeling Using the Power-Oriented Graphs Technique. Proceedings of the European Control Conference (ECC), 25-28 June 2019.

R. Zanasi, D. Tebaldi (b). Power-Flow Efficiency of Linear and Nonlinear Physical Systems. Proceedings of the European Control Conference (ECC), 25-28 June 2019.

R. Zanasi, D. Tebaldi (c). Study of the Bidirectional Efficiency of Linear and Nonlinear Physical Systems. Proceedings of the IEEE 45th Annual Conference of the Industrial Electronics Society (IECON), 14-17 Oct. 2019.

#### Appendix A. FUNCTION DX\_DT(T,Z)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [dz,zf]=DX_DT(t,z)
% t = time vector
% z = input vector
% dz = derivative of z with respect to t
% zf = filtered input vector
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initial value:
dz(1)=(z(2)-z(1))/(t(2)-t(1));   zf(1)=z(1);
% Intermediate values:
for i=[2:length(t)-1]
    x0=t(i-1);   x1=t(i);   x2=t(i+1);   s0=zf(i-1);
    ds0=dz(i-1);   y1=z(i);   y2=z(i+1);
    a=-((s0-ds0*x0+ds0*x1-y1)*(x0-x1)^2+...
        (s0-ds0*x0+ds0*x2-y2)*(x0-x2)^2)...
        /((x0-x1)^4+(x0-x2)^4);
    b=ds0-2*a*x0;   c=s0-a*x0^2-b*x0;
    zf(i)=a*x1^2+b*x1+c;   dz(i)=2*a*x1+b;
end
% Last value:
i=length(t);
zf(i)=z(i);   dz(i)=(z(i)-zf(i-1))/(t(i)-t(i-1));
return
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    
```