On the compensation of incremental encoder imperfections for motion control: a DC motor case-study

Missie Aguado-Rojas, William Pasillas-Lépine, Antonio Loria, Alexandre de Bernardinis

On the compensation of incremental encoder imperfections for motion control: a DC motor case-study ⋆

Missie Aguado-Rojas * **William Pasillas-Lépine** **
Antonio Loría ** **Alexandre De Bernardinis** ***

* *Université Paris-Sud, Université Paris-Saclay, L2S-CentraleSupélec, 91192 Gif-sur-Yvette, France*
** *CNRS, L2S-CentraleSupélec, 91192 Gif-sur-Yvette, France*
*** *SATIE, IFSTTAR-LTN, 78000 Versailles, France*
(e-mails: {missie.aguado}{pasillas}{loria}@l2s.centralesupelec.fr, alexandre.de-bernardinis@ifsttar.fr)

Abstract: We address the problem of rotational velocity and acceleration estimation from incremental encoders in the presence of sensor imperfections. In the area of motion control, measurements of shaft velocity and acceleration are often affected by large disturbances with a period of one revolution that arise from sensor imperfections and degrade the performance of most closed-loop control algorithms. We present an algorithm to identify and remove such periodic perturbations online, without the need of error compensation look-up tables, and without assuming constant velocity. The proposed algorithm is evaluated via open-loop experimental tests, and its implementation in closed-loop control applications is investigated via numerical simulations through a DC motor case-study.

Keywords: Velocity and acceleration estimation; incremental encoders; periodic measurement noise; parameter estimation; DC motor control;

1. INTRODUCTION

Real-time measurement of angular position, velocity and acceleration plays a crucial role in numerous motion control applications [Ohnishi et al., 1996; Iwasaki et al., 2012; Huang and Chou, 2015]. The most commonly used technology to measure such signals is based on incremental shaft encoders. They consist, mainly, in a toothed wheel (or a slotted disc) attached to the rotating shaft and a fixed pick-off sensor that detects the passing of the teeth and outputs a square wave signal, in which each edge corresponds to the edge of one tooth [De Silva, 2015, § 6.7]. Thus, the position, velocity, and acceleration are not directly measured, but they have to be reconstructed from the encoder pulses.

Several algorithms have been proposed to that end, hence the problem of rotational velocity and acceleration estimation from (ideal) incremental encoder measurements is well-addressed (see Bascetta et al. [2009] and Ronsse et al. [2013] for comprehensive reviews on the state-of-the-art). However, since these algorithms rely on the output of a real sensor subject to manufacturing and assembly tolerances, the resulting estimations are inevitably affected by the encoder imperfections. In Aguado-Rojas et al. [2017], we introduced a velocity estimation algorithm that reduces the effects of encoder imperfections, without requiring the availability of error look-up tables. The advantage of this approach, based on a time-stamping algorithm [Merry et al., 2010], is that it has a good performance in terms of reducing the effect of encoder imperfections without introducing a long delay, which is a typical drawback of approaches based on notch filters, like those proposed in Panzani et al. [2012] and Hoàng et al. [2012].

The aim of this paper is to extend the approach proposed in Aguado-Rojas et al. [2017] to provide, in addition to the velocity, an estimation of the acceleration. The key point of our approach is to take into account the effects of the delay introduced by the time-stamping algorithm, by considering a model for the acceleration perturbation that differs from the derivative of the velocity perturbation. Our model also takes into account the link between the number of encoder events considered by the time-stamping algorithm and the spectral width of the perturbation on the measured velocity. Using this model and the least-square method (or, equivalently, the gradient algorithm) one can identify in real-time the characteristics of the encoder imperfections, and compensate them in order to deduce the actual velocity and acceleration of the system. As an example, we illustrate the potential usefulness of this new estimation algorithm in closed-loop control applications through a DC motor case-study.

The rest of the paper is organized as follows. In Section 2 we present a brief review on the most common encoder imperfections, and the effects that they have on the position, velocity and acceleration measurements. In Section 3 we introduce three models that capture the effects of sensor imperfections on the measured signals. Based on these models, in Section 4 we propose a method to identify and remove the periodical disturbances introduced

⋆ The work of the first author is supported by CONACYT and SEP, Mexico.

by sensor imperfections, and validate it via open-loop experimental tests. The implementation of the proposed method in closed-loop control applications is investigated in Section 5. Concluding remarks are given in Section 6.

2. PRELIMINARIES: ON ENCODER IMPERFECTIONS AND THEIR EFFECTS

An ideal incremental encoder is characterised by identical and equidistant teeth (or slits) distributed over the encoder's code-wheel, and quadrature output channels with 50% duty cycles. However, real devices are subject to well-known manufacturing and assembly imperfections [Kavanagh, 2000; 2001; 2002]. Generated by these imperfections, the most significant error sources in the encoder output signals are due to [Merry et al., 2013]:

- Cycle error: stochastic variations in the location of the rising or falling edges of the encoder pulses from their nominal values, due to unequal positional spacings of the slits, as well as limitations and irregularities of the encoder's signal generation and sensing hardware.
- Pulse-width error: the deviation of the pulse width from its ideal value of 180°.
- Phase error: the deviation of the phase between the two output channels from its ideal value of 90°.
- Eccentricity or tilt of the encoder's code-wheel, due to concentricity and assembly tolerances.

These error sources introduce a time shift to all edges of the output pulses, hence resulting in inexact readings of displacement that affect the quality of the position, velocity and acceleration measurements.

Cycle, pulse-width, and phase errors appear as high-frequency variations when viewed over the circumference of the encoder's code-wheel, while eccentricity or tilt describe a systematic low-frequency variation with a period of one revolution. Furthermore, because the measurement error caused by sensor imperfections is periodic over one mechanical revolution, the measurements of velocity and acceleration contain large periodic perturbations whose frequency is locked with the rotational frequency of the shaft. In the following section we introduce three models that capture the effects of sensor imperfections on the measured position, velocity, and acceleration signals, based on which our estimation algorithm is designed.

3. MEASUREMENT MODELS

The arguments given in the previous section and the experimental data reported in Aguado-Rojas et al. [2017], lead us to model the effects of sensor imperfections using a 2π-periodic function, that we denote f_r, added to the real position θ_r. Hence the measured position θ_m corresponds to

$$\theta_m = \theta_r + f_r(\theta_r).$$

If the function f_r is approximated using a finite number M_r of Fourier coefficients, one obtains

$$\theta_m = \theta_r + \sum_{k=1}^{M_r} \left[a_k^r \sin(k\theta_r) + b_k^r \cos(k\theta_r)\right].$$

In what follows, we assume that both $a_k^r \ll 1$ and $b_k^r \ll 1$. This condition is always satisfied if the encoder is of reasonable quality and if the sensor has been correctly mounted on its shaft.

The fact that θ_r is unknown limits the applicability of the previous model. Nevertheless, when the perturbation f_r is sufficiently small, the function

$$\varphi_r(\theta_r) = \theta_r + f_r(\theta_r)$$

is invertible so, denoting this inverse by φ_m, we have

$$\theta_m = \theta_r + f_r \circ \varphi_m(\theta_m)$$

and, defining $f_m = f_r \circ \varphi_m$, we obtain

$$\theta_m = \theta_r + f_m(\theta_m).$$

The function f_m is also a 2π-periodic. Note, however, that the Fourier coefficients of a finite approximation of f_m, given by

$$\theta_m = \theta_r + \sum_{k=1}^{M} \left[a_k \sin(k\theta_m) + b_k \cos(k\theta_m)\right], \qquad (1)$$

are in general different from those of f_r, even when $M = M_r$. Moreover, the fact that the Fourier approximation of f_r of order M_r is exact does not necessarily imply that the same is true for f_m. In fact, it might be necessary to take $M > M_r$ to obtain a good approximation.

The new model (1) is the starting point for our estimation algorithm. The differentiation with respect to time of the expression above leads to the velocity and acceleration measurement models

$$\omega_m = \omega_r + \omega_m \sum_{k=1}^{M} \left[k a_k' \cos(k\theta_m) - k b_k' \sin(k\theta_m)\right] \qquad (2)$$

and

$$\alpha_m = \alpha_r + \alpha_m \sum_{k=1}^{M} \left[k a_k'' \cos(k\theta_m) - k b_k'' \sin(k\theta_m)\right] - \omega_m^2 \sum_{k=1}^{M} \left[k^2 a_k'' \sin(k\theta_m) + k^2 b_k'' \cos(k\theta_m)\right], \qquad (3)$$

in which the Fourier coefficients a_k', b_k', a_k'' and b_k'' are unknown. Therefore, our goal is to estimate the Fourier coefficients in order to recover reliable estimates of ω_r and α_r.

Remark 1. The notation $(\cdot)'$ and $(\cdot)''$ is used here in order to emphasize that, depending on the algorithm employed to reconstruct the position, velocity and acceleration signals from the encoder pulses, the Fourier coefficients in (2) and (3) may not necessarily be equal to those in (1).

In our work, in order to reconstruct angular velocity and acceleration from the output pulses of an incremental encoder, we employ a method known as the *time-stamping algorithm* [Merry et al., 2013]. It consists in capturing, via a high-resolution clock, the time instants in which the rising or falling edges of the output pulses are detected, and using the information of the last n events to approximate the evolution of the angular position with a polynomial of order m. Estimates of the angular velocity ω and acceleration α are then obtained via analytic differentiation of the fitted polynomial with respect to time.

An evaluation of the quality of these models as a function of the number of harmonics M seems to indicate that the fit between data and the model improves until $M = \lceil N/n \rceil + 1$, where N is the number of pulses per revolution of the encoder, n is the number of events used in the fit, and $\lceil \cdot \rceil$ is the smallest integer greater than or equal to the argument. In other words, taking less events reduces

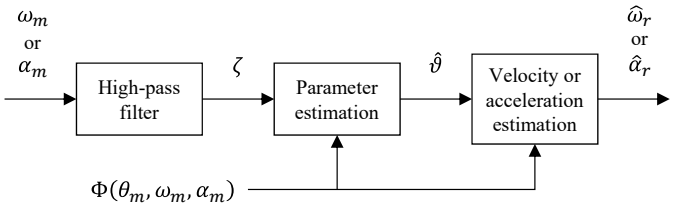

Fig. 1. Three-stage velocity and acceleration estimation algorithm.

the measurement delay but increases the complexity of the model. Based on (2) and (3), an algorithm to reduce the periodic noise present in the measured velocity and acceleration can be designed as discussed in the following section.

4. VELOCITY AND ACCELERATION ESTIMATION

Let us rewrite (2) and (3) as

$$\omega_m = \omega_r + \omega_m \phi(\theta_m)^\top D \vartheta' \tag{4}$$

and

$$\alpha_m = \alpha_r + \left[\alpha_m \phi(\theta_m)^\top D - \omega_m^2 \psi(\theta_m)^\top D^2\right] \vartheta'', \tag{5}$$

where $D \in \mathbb{R}^{2M \times 2M}$, ϕ and $\psi \in \mathbb{R}^{2M}$ are defined as

$$D = \mathrm{diag}(1, 1, 2, 2, \ldots),$$

$$\phi(\theta_m) = [\cos(\theta_m) \ -\sin(\theta_m) \ \cos(2\theta_m) \ -\sin(2\theta_m) \ \cdots]^\top,$$

$$\psi(\theta_m) = [\sin(\theta_m) \ \cos(\theta_m) \ \sin(2\theta_m) \ \cos(2\theta_m) \ \cdots]^\top,$$

and $\vartheta', \vartheta'' \in \mathbb{R}^{2M}$ contain the corresponding coefficients a'_k, b'_k, a''_k and b''_k.

From these models, the measured velocity ω_m and the measured acceleration α_m can be seen as the sum of a low-frequency term —ω_r in (4) and α_r in (5)— and a high-frequency (with respect to the first one) term, of the form

$$\bar{\zeta} = \Phi(\theta_m, \omega_m, \alpha_m)^\top \vartheta, \tag{6}$$

that depends on the known signals θ_m, ω_m, and α_m, and is linear in the unknown parameters ϑ. For simplicity of notation, in (6) and in the rest of the paper, we use Φ and ϑ to denote

$$\Phi = \begin{cases} D\phi(\theta_m)\omega_m & \text{for the model (4)} \\ D\phi(\theta_m)\alpha_m - D^2\psi(\theta_m)\omega_m^2 & \text{for the model (5),} \end{cases}$$

and

$$\vartheta = \begin{cases} \vartheta' & \text{for the model (4)} \\ \vartheta'' & \text{for the model (5).} \end{cases}$$

In order to estimate the real velocity (resp. acceleration), we perform the three-stage algorithm illustrated in Fig. 1, which is based on the work presented in Aguado-Rojas et al. [2017]. In the first stage, in order to separate $\bar{\zeta}$ from the other terms in (4) (resp. (5)), the measured signal is filtered using a first-order high-pass filter with a cutoff frequency that is considerably below that of the wheel revolution (for example 1 Hz).

In the second stage, assuming that the filters are ideal and they completely eliminate the low-frequency term of the measured signals, i.e. $\zeta \approx \bar{\zeta}$, the Fourier coefficients of the periodic perturbation in (4) (resp. (5)) are estimated using the parametric model

$$\zeta = \Phi(\theta_m, \omega_m, \alpha_m)^\top \vartheta \tag{7}$$

via the normalized recursive least-squares (RLS) algorithm with forgetting factor [Ioannou and Sun, 2012, Ch. 4]:

$$\dot{\widehat{\vartheta}} = -\Gamma \Phi \varepsilon, \qquad \widehat{\vartheta}(0) = \widehat{\vartheta}_0 \tag{8}$$

$$\dot{\Gamma} = \beta\Gamma - \frac{\Gamma\Phi^\top\Phi\Gamma}{1 + \kappa\Phi^\top\Phi}, \qquad \Gamma(0) = \Gamma_0 \tag{9}$$

$$\varepsilon = \frac{\Phi^\top\widehat{\vartheta} - \zeta}{1 + \kappa\Phi^\top\Phi} \tag{10}$$

where $\kappa > 0$, $\beta > 0$ and $\Gamma_0 = \Gamma_0^\top > 0$.

Note that due to the form of the parametric model (7) any other standard algorithm may be used. Note also that, due to the form of $\phi(\theta_m)$, $\psi(\theta_m)$ and D, the real-time implementation of the RLS algorithm can be greatly simplified, especially for large values of M. That is, the term $\Phi^\top\Phi$ in (9) and (10) simplifies to

$$\Phi^\top\Phi = \begin{cases} \varsigma_2\omega_m^2 & \text{for the model (4)} \\ \varsigma_2\alpha_m^2 + \varsigma_4\omega_m^4 & \text{for the model (5)} \end{cases}$$

where $\varsigma_2 = \sum_{k=1}^{M} k^2$ and $\varsigma_4 = \sum_{k=1}^{M} k^4$.

Finally, using the estimated parameters $\widehat{\vartheta}$, we construct our velocity estimate as

$$\widehat{\omega}_r = \omega_m - \omega_m \phi(\theta_m)^\top D \widehat{\vartheta}' \tag{11}$$

and our acceleration estimate as

$$\widehat{\alpha}_r = \alpha_m - \left[\alpha_m \phi(\theta_m)^\top D - \omega_m^2 \psi(\theta_m)^\top D^2\right] \widehat{\vartheta''}. \tag{12}$$

Remark 2. As a direct consequence of Remark 1, in general $\alpha_m \neq \dot{\omega}_m$. Therefore, the estimated parameters $\widehat{\vartheta}'$ of the velocity measurement model (4) should not be used for acceleration estimation using (12), and the estimated parameters $\widehat{\vartheta''}$ of the acceleration measurement model (5) should not be used for velocity estimation using (11).

The (open-loop) experimental validation of the estimation algorithm is illustrated in Figures 2 and 3, using a piecewise-constant reference. The initial condition for the estimated parameters was arbitrarily set to zero, and Γ_0 was chosen as a matrix of the form

$$\Gamma_0 = \mathrm{diag}(\gamma_1, \gamma_1, \gamma_2, \gamma_2, \ldots)$$

with $\gamma_1 > \gamma_2 > \ldots > \gamma_M > 0$. A comparison with respect to the results obtained using a notch filter to remove the periodic perturbations introduced by sensor imperfections is shown as well.

The depicted results show that, once the estimated parameters have converged to appropriate values, the estimated signals contain significantly smaller oscillations than the measured signals. This can be more easily noticed during the intervals in which the velocity or the acceleration are constant. During this intervals, the level of attenuation of the periodic perturbations of the estimation algorithm is similar to the one of the notch filter. The usefulness of the estimation algorithm over a notch filter is, however, more evident during the intervals in which the velocity or the acceleration are changing. Even though both the notch filter and the estimation algorithm show a good performance in terms of reducing the amplitude of the oscillations, the filter clearly introduces a significant delay, whereas the estimation scheme follows the reference with no noticeable delay. The use of the estimation algorithm in a closed-loop control scenario is investigated in the following section.

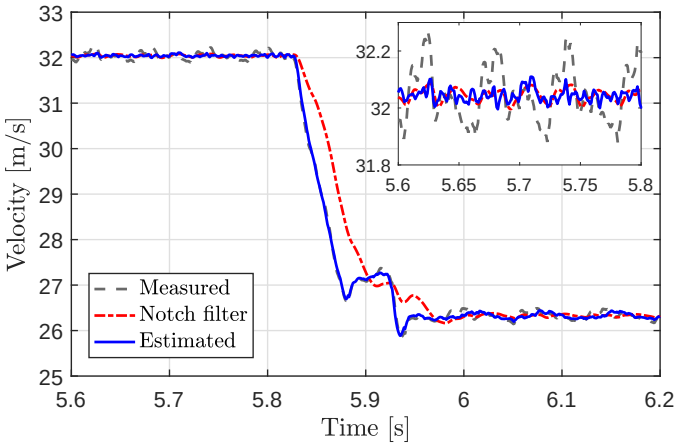

Fig. 2. Measured vs. filtered and estimated velocity.

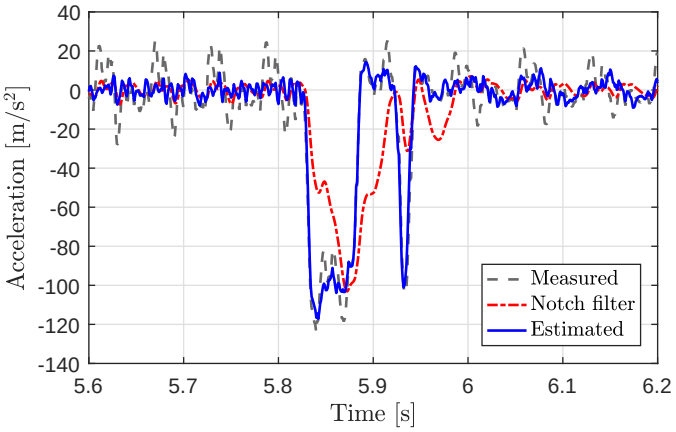

Fig. 3. Measured vs. filtered and estimated acceleration.

5. CASE-STUDY: DC MOTOR CONTROL

In what follows, we present a control algorithm for the velocity-tracking problem of the DC motor. It is important to stress at this point, that our aim is not to develop a new control strategy that would supersede the ones that already exist. Instead, our aim is to investigate, through a simple academic case-study, the advantages and drawbacks of our estimation algorithm in control applications, relative to other estimation methods, such as approximate differentiation.

5.1 System modelling

Let us consider the mechanical dynamics of a DC motor, that is,

$$J\frac{d\omega}{dt} + B\omega = T_M - T_L \tag{13}$$

$$\frac{d\theta}{dt} = \omega, \tag{14}$$

where θ and ω are the angular position and velocity of the shaft, respectively, J is the rotor inertia, B is the viscous friction coefficient, T_M is the torque exerted by the motor, and T_L is the load torque. The electrical dynamics is described by

$$L_a\frac{di_a}{dt} + R_a i_a = V_a - E, \tag{15}$$

where i_a is the armature current, V_a is the armature voltage, which is used as control input, L_a is the armature inductance, R_a is the armature resistance, and E is the back-emf. Assuming that the magnetic flux Φ is constant, the dynamics (13) and (15) are related by the expressions

$$T_M = K_t i_a \tag{16}$$

$$E = K_e\omega, \tag{17}$$

where $K_t = K_t'\Phi$ and $K_e = K_e'\Phi$ are constants. In SI units, the motor torque and back-emf proportionality constants are equal. Hence, in what follows, $K := K_t = K_e$ is used. In addition, it is assumed that the parameters J, B, K, R_a, and L_a are known while the constant perturbation T_L is unknown. Furthermore it is also assumed that only the armature current i_a and the angular position θ are measured. The latter, however, is obtained from an imperfect incremental encoder.

Then, the control objective is formulated as follows. Given a bounded and twice differentiable velocity reference $\omega^* : \mathbb{R}_{\geq 0} \to \mathbb{R}$, design V_a such that

$$\omega - \omega^* \to 0 \quad \text{as} \quad t \to \infty.$$

5.2 Benchmark controller

If the velocity and acceleration of the motor were measured, a simple pole-placement controller might be designed as follows. First, we design a virtual control law T_M^* for the mechanical dynamics (13)–(14), taking T_M as a virtual control input. More precisely, we design T_M^* to steer the tracking error $e_1 := \omega - \omega^*$ to zero, for any given velocity reference ω^*. Then, defining $i_a^* := T_M^*/K$, we design a control law V_a to stabilize (15) at the reference i_a^*.

There is a variety of ways to stabilize the system (13), (14) while rejecting the constant perturbation T_L. For the purpose of stressing the utility of our velocity-and-acceleration-estimation method, we employ the commonly-used Proportional Integral controller, to which we add a feedforward term to ensure tracking control of non-constant references ω^*. That is, let

$$T_M^* = -k_p' e_1 - k_i' \nu + B\,\omega^* + J\,\dot{\omega}^* \tag{18}$$

$$\dot{\nu} = e_1, \tag{19}$$

where k_p' and k_i' are positive gains. Then, let us define $e_2 := \nu + \frac{T_L}{k_i'}$ and $\tilde{T}_M := T_M - T_M^*$ so that, using $\nu = e_2 - \frac{T_L}{k_i'}$ in (19) and $T_M = \tilde{T}_M + T_M^*$ in (13), the closed-loop dynamics become

$$\dot{e}_1 = -\frac{B + k_p'}{J} e_1 - \frac{k_i'}{J} e_2 + \frac{\tilde{T}_M}{J} \tag{20}$$

$$\dot{e}_2 = e_1. \tag{21}$$

Clearly, provided that $\tilde{T}_M \equiv 0$, $\{(e_1, e_2) = (0,0)\}$ is exponentially stable for any positive values of k_p' and k_i'. Furthermore, the system (20), (21) is input-to-state stable with respect to $\tilde{T}_M$, hence $(e_1, e_2) \to (0,0)$ if $\tilde{T}_M \to 0$.

In order to steer $\tilde{T}_M$ to zero, it is left to design a tracking controller to stabilize the system (15) on the reference current i_a^*, which is defined using (18). That is,

$$i_a^* = \frac{T_M^*}{K} = \frac{J}{K}\left(-k_p\, e_1 - k_i\, e_2 + \frac{B}{J}\,\omega + \dot{\omega}^*\right) \tag{22}$$

$$\dot{e}_2 = e_1, \tag{23}$$

where

$$k_p := \frac{B + k_p'}{J}, \quad k_i := \frac{k_i'}{J}.$$

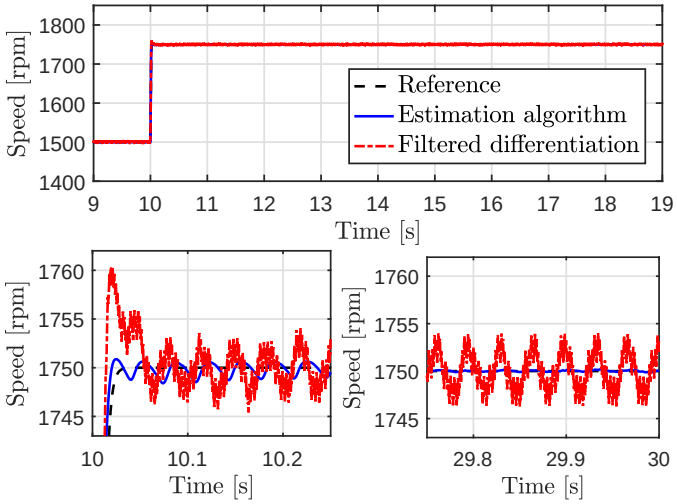

Fig. 4. Speed tracking for a step-like reference.

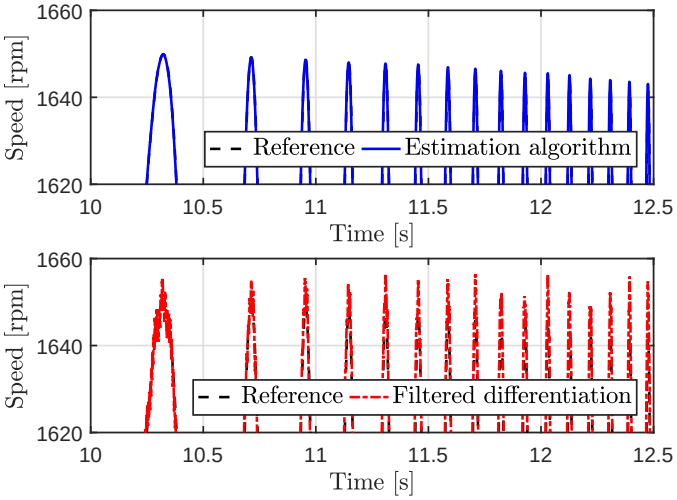

Fig. 5. Speed tracking for a chirp-like reference.

Steering the current tracking error $e_3 := i_a - i_a^*$ to zero may be achieved using a simple proportional-plus-feedforward controller. That is,

$$V_a = -L_a k_e\, e_3 + R_a\, i_a + K\omega + L_a \frac{di_a^*}{dt}, \tag{24}$$

where $k_e > 0$, and

$$\frac{di_a^*}{dt} = \frac{J}{K}\left(-k_p\,(\dot\omega - \dot\omega^*) - k_i\, e_1 + \frac{B}{J}\,\dot\omega + \ddot\omega^*\right). \tag{25}$$

Such a choice, leads to the error dynamics

$$\dot e_1 = -k_p\, e_1 - k_i\, e_2 + \sigma\, e_3 \tag{26}$$
$$\dot e_2 = e_1 \tag{27}$$
$$\dot e_3 = -k_e\, e_3, \tag{28}$$

where $\sigma = K/J$. Exponential stability of the origin is easily checked by computing the characteristic polynomial

$$p(\lambda) = (\lambda + k_e)\left(\lambda^2 + k_p\lambda + k_i\right),$$

that has roots with strictly negative real part, provided that k_p, k_i, and k_e are strictly positive. Moreover, these parameters may be easily tuned according to specific performance requirements.

Although effective, the PI-based controller defined above has the drawback that the actual control input (24) requires measurement of both velocities ω and accelerations $\dot\omega$. Not only these are assumed to be unavailable from measurement but only an inaccurate position measurement θ_m is available. In the next section we discuss and compare two possible implementations of the controller (22)–(24) using, on one hand, the estimates $\hat\omega_r$ and $\hat\alpha_r$ obtained via the algorithm presented in Section 4 and, on the other, the *ad-hoc* technique of dirty derivatives.

5.3 Simulation results

In this section we illustrate the performance of the controller described above via numerical simulations. It is considered that the signals used for the computation of the control input are the output pulses of an imperfect incremental encoder.

For the purpose of comparison, two implementation scenarios are considered. The first one employs the estimates $\hat\omega_r$ and $\hat\alpha_r$ from the estimation algorithm. In the second scenario, the estimates of angular velocity and acceleration are computed using filtered differentiation of the shaft angular position. A second-order filter with natural frequency ω_n and damping ratio ξ was implemented, using the following realization:

$$\dot x_1 = x_2$$
$$\dot x_2 = -\omega_n^2 x_1 - 2\xi\omega_n x_2 + \omega_n^2 u$$

where the input u is the quantized angular position (decoded from the encoder pulses), and the outputs are the filtered position, velocity and acceleration, $\hat\theta_f$, $\hat\omega_f$, and $\hat\alpha_f$, given by

$$y = \begin{bmatrix}\hat\theta_f\\ \hat\omega_f\\ \hat\alpha_f\end{bmatrix} = \begin{bmatrix} x_1\\ x_2\\ -\omega_n^2 x_1 - 2\xi\omega_n x_2 + \omega_n^2 u\end{bmatrix}. \tag{29}$$

For the simulations included in this paper, the values $\omega_n = 60$ Hz, and $\xi = 1$ were chosen, in order to have a good compromise between the response-time of the filter and the level of amplification of noise (which is due to the digital nature of the encoder).

Figures 4 and 5 illustrate the response of the controller to a smoothed step and a chirp-like reference, under the two implementation scenarios described above. Note that, due to the periodic nature of encoder imperfections, the period of the oscillations present in the velocity signals corresponds to one shaft revolution. A better performance is observed when the controller is implemented using the algorithm for estimation and compensation of encoder imperfections, than with filtered differentiation. The tracking error is smaller both during the transient and steady-state intervals. This is consistent with the information depicted in Figure 6, which shows the frequency response of the controller. In both cases, a peak is observed at 25 Hz, that is, the frequency of rotation of the shaft. The amplitude of this frequency component, however, is much smaller for the case of the implementation using the estimation algorithm, than for the case using filtered differentiation.

6. CONCLUSION

We have extended the velocity estimation algorithm introduced in Aguado-Rojas et al. [2017] to the case of acceleration estimation. The algorithm aims at reducing the effects of the most common encoder imperfections, namely the presence of large periodic perturbations whose frequency is locked with the rotational frequency of the

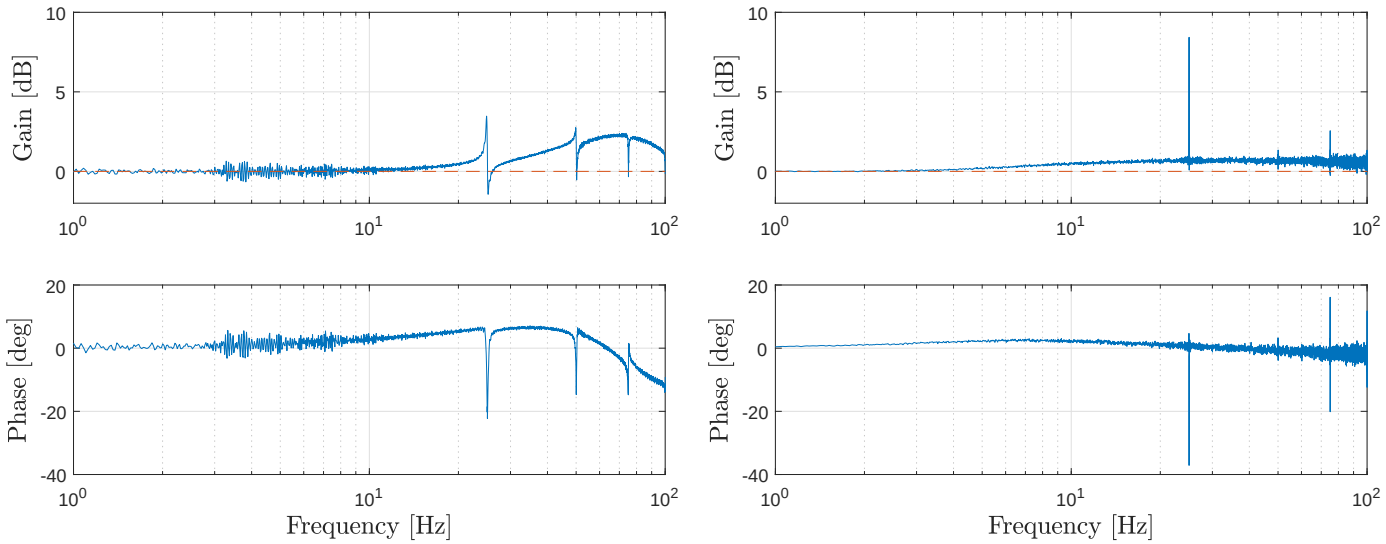

Fig. 6. RIGHT: Frequency response of the controller using estimation algorithm. LEFT: Frequency response of the controller using filtered differentiation.

shaft. The effectiveness of the proposed algorithm has been evaluated via open-loop experimental tests with satisfactory results. Moreover, as a first-step towards the implementation of the estimation algorithm in closed-loop control applications, the performance of a controller with velocity and acceleration feedback has been investigated via numerical simulations through a DC motor case-study.

Future work will focus on the practical implementation of the proposed estimation method. The testbench, in which the closed-loop tests are to be performed consists of two identical three-phase synchronous motors coupled via a torque transducer and driven via an IGBT-based converter and a dSPACE board. The implementation of the time-stamping algorithm could be done with a dSPACE dedicated incremental encoder interface card, or with general-purpose data-acquisition devices, such as a TUeDACs Microgiant as in Merry et al. [2010; 2013].

REFERENCES

Aguado-Rojas, M., Pasillas-Lépine, W., Loría, A., and De Bernardinis, A. (2017). Angular velocity estimation from incremental encoder measurements in the presence of sensor imperfections. In *Proceedings of the 20th IFAC World Congress*, 6153–6158.

Bascetta, L., Magnani, G., and Rocco, P. (2009). Velocity estimation: Assessing the performance of non-model-based techniques. *IEEE Transactions on Control Systems Technology*, 17(2), 424–433.

De Silva, C.W. (2015). *Sensors and actuators: Engineering system instrumentation*. CRC Press, second edition.

Hoàng, T.B., Pasillas-Lépine, W., and De Bernardinis, A. (2012). Reducing the impact of wheel-frequency oscillations in continuous and hybrid abs strategies. In *Proceedings of the 11th International Symposium on Advanced Vehicle Control (AVEC'12)*.

Huang, H. and Chou, W. (2015). Hysteresis switch adaptive velocity evaluation and high-resolution position subdivision detection based on FPGA. *IEEE Transactions on Instrumentation and Measurement*, 64(12), 3387–3395.

Ioannou, P. and Sun, J. (2012). *Robust Adaptive Control*. Dover Books on Electrical Engineering. Dover Publications.

Iwasaki, M., Seki, K., and Maeda, Y. (2012). High-precision motion control techniques: A promising approach to improving motion performance. *IEEE Industrial Electronics Magazine*, 6(1), 32–40.

Kavanagh, R.C. (2000). Shaft encoder characterization via theoretical model of differentiator with both differential and integral nonlinearities. *IEEE Transactions on Instrumentation and Measurement*, 49(4), 795–801.

Kavanagh, R.C. (2001). Performance analysis and compensation of M/T-type digital tachometers. *IEEE Transactions on Instrumentation and Measurement*, 50(4), 965–970.

Kavanagh, R.C. (2002). Shaft encoder characterisation through analysis of the mean-squared errors in nonideal quantised systems. *IEE Proceedings - Science, Measurement and Technology*, 149(2), 99–104.

Merry, R., van de Molengraft, M., and Steinbuch, M. (2010). Velocity and acceleration estimation for optical incremental encoders. *Mechatronics*, 20(1), 20–26. Special Issue on Servo Control for Data Storage and Precision Systems, from 17th IFAC World Congress 2008.

Merry, R., van de Molengraft, M., and Steinbuch, M. (2013). Optimal higher-order encoder time-stamping. *Mechatronics*, 23(5), 481–490.

Ohnishi, K., Shibata, M., and Murakami, T. (1996). Motion control for advanced mechatronics. *IEEE/ASME Transactions on Mechatronics*, 1(1), 56–67.

Panzani, G., Corno, M., and Savaresi, S.M. (2012). On the periodic noise affecting wheel speed measurement. In *Proceedings of the 16th IFAC Symposium on System Identification*, volume 45, 1695–1700.

Ronsse, R., De Rossi, S.M.M., Vitiello, N., Lenzi, T., Carrozza, M.C., and Ijspeert, A.J. (2013). Real-time estimate of velocity and acceleration of quasi-periodic signals using adaptive oscillators. *IEEE Transactions on Robotics*, 29(3), 783–791.