



Estimation of stepping motor current from long distances through cable-length-adaptive piecewise affine virtual sensor

Alberto Oliveri¹, Mark Butcher², Alessandro Masi², Marco Storage¹

¹ Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture (DITEN), University of Genoa, Via Opera Pia 11a, 16145, Genova, Italy

² Department of Engineering, CERN, 1211 Geneva, Switzerland

ABSTRACT

In this paper a piecewise affine virtual sensor is used for the estimation of the motor-side current of hybrid stepper motors, which actuate the LHC (Large Hadron Collider) collimators at CERN. The estimation is performed starting from measurements of the current in the driver, which is connected to the motor by a long cable (up to 720 m). The measured current is therefore affected by noise and ringing phenomena. The proposed method does not require a model of the cable, since it is only based on measured data and can be used with cables of different length. A circuit architecture suitable for FPGA implementation has been designed and the effects of fixed point representation of data are analyzed.

Section: RESEARCH PAPER

Keywords: virtual sensor; estimation; FPGA; piecewise-affine functions

Citation: Alberto Oliveri, Mark Butcher, Alessandro Masi, Marco Storage, Estimation of stepping motor current from long distances through cable-length-adaptive piecewise affine virtual sensor, Acta IMEKO, vol. 4, no. 3, article 9, September 2015, identifier: IMEKO-ACTA-04 (2015)-03-09

Editor: Paolo Carbone, University of Perugia, Italy

Received February 13, 2015; In final form May 29, 2015; Published September 2015

Copyright: © 2015 IMEKO. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Funding: Work supported by the University of Genoa.

Corresponding author: Marco Storage, e-mail: marco.storage@unige.it

1. INTRODUCTION

The LHC (Large Hadron Collider) at CERN is a circular particle accelerator which, as all machines of this kind, needs a collimation system to block highly energetic particles flying off their trajectories. If these particles are not properly collected, they can seriously damage the accelerator. The module that collects these potentially harmful particles is called collimation system and comprises some moving parts that need to be actuated with high precision inside the tunnel. Hybrid stepper motors are often used as the actuators in the collimation system because of their high positioning repeatability and open loop control [1]. These motors, and their electronic drives, are subject to a number of requirements that are relatively unique to accelerators. The environment surrounding the motors is highly radioactive and driver electronics are damaged by this radioactivity, then the drivers are placed in radiation-safe zones at a distance of up to 1 km from the motors. They must therefore be connected to the hybrid stepper motors via long cables, as shown in Figure 1.

Pulse Width Modulated (PWM) control voltages are largely used to increase power efficiency; however, they generate significant electromagnetic interference (EMI) emissions, which can affect neighbouring electronics. High-frequency PWM chopping frequencies must therefore be used to shift the emissions to higher frequencies. These high-frequency voltage signals, nonetheless, cause the long cables to act as transmission lines, and produce a *ringing phenomenon* in the currents on the driver-side of the cable, the only side where measurements are possible. Figure 2 shows a comparison of the current in the driver-cable-motor circuit for a single motor phase.

Good motor positioning repeatability is of great importance and requires to have real-time knowledge of the motor's position, in order that compensatory action can be taken to correct any misalignments (i.e., steps losses). Radiation-hard resolvers are used to measure the motor position and to detect lost steps. It is, nonetheless, desirable to have sensor redundancy. Additionally, even if the stepping motors work at nominal torque, chosen by design to be at least twice the nominal load torque, having an estimate of the real load torque



Figure 1. Illustration of the connection of a 2-phase hybrid stepper motor to its driver via a long cable, with the corresponding phase currents.

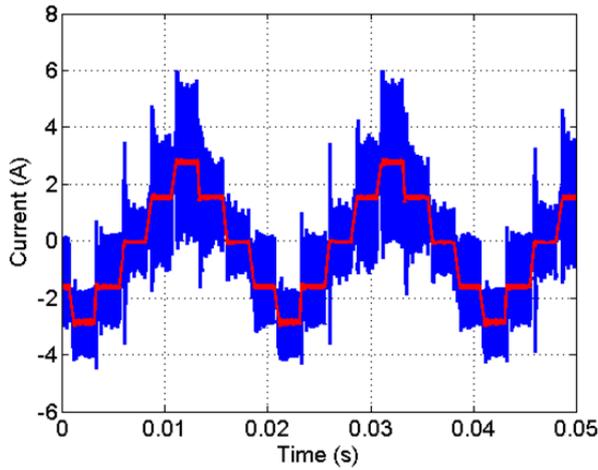


Figure 2. Comparison of the current in the driver-cable-motor circuit for a single motor phase: Driver-side (blue) and Motor-side (red).

can be useful to warn about degradation of the mechanical parts of the collimation system.

A sensorless driver, based on an extended Kalman Filter (EKF), which can work with arbitrarily long power cables between the driver and motor has been developed in [2].

In order to use exclusively the motor model in the EKF, the algorithm's inputs and measurements should be the motor-side voltages and currents, respectively. However, as previously stated, these signals are not directly available in ordinary operation since measurements can only be made on the drive side of the cable. In [2] a cable model is used to obtain a transfer function relating the motor-side current to the driver-side current. The current is therefore estimated by means of a digital filter whose coefficients can be easily adapted to the used cable.

A *physical sensor* is a hardware device that converts physical phenomena into electric signals. A *virtual sensor*, instead, uses mathematical models (usually obtained through a black box-approach) and the reading of other physical sensors to indirectly estimate a given physical variable. Virtual sensors are basically nonlinear functions of past inputs and measured outputs of the considered system. They can be useful when some variables are not easily observable or measurable for various reasons, such as availability, cost or maintenance of a physical sensor. In summary, a virtual sensor can be used to estimate an unmeasurable system variable, without needing the knowledge of a mathematical model of the system.

A procedure for the design of a virtual sensor is described in [3], which relies on choosing a suitable set of basis functions, so that the resulting virtual sensor satisfies the assumptions required to apply the theoretical results in [4]. In 0, [6] and [7], PWAS functions (i.e. PieceWise Affine functions defined over a Simplicial partition) have been proposed as basis functions for the design of virtual sensors. The main advantage of using PWAS functions is that they can be implemented very efficiently in digital programmable circuits such as FPGAs,

providing low power consumption, fast response times, and, at least for high-volume applications, low cost. Moreover, convergence and optimality properties of the general approach [3] are maintained.

In 0 a single PWAS function is used to obtain the estimate (*standard virtual sensor*), but if a relatively large number of inputs or measurable outputs is available, or if a large number of past data are used, the exponential increase of the complexity (curse of dimensionality) makes the approach impractical. An alternative solution (*reduced-complexity virtual sensor*) leading to a complexity reduction has been proposed in [6] and [7]. The method used for estimation of the motor-side currents in [2] is based on an approximation of a first principles model of the dynamical system relating the drive-side current to the motor-side current. It therefore requires multiple, time intensive steps of modelling, parameter estimation and approximation to produce a real-time implementable, reduced-order estimator.

In [8] the reduced-complexity PWAS virtual sensor has been used for the estimation of the motor-side current with a fixed cable length of 720 m. Here we generalize the sensor in order to make it cable-length-adaptive. The sensor is indeed designed (trained) starting from measurements obtained with cable lengths of 180 m, 360 m and 540 m while the validation is performed on measurements related to a 720 m long cable.

A circuit architecture suitable for FPGA implementation has been also designed and simulation results considering fixed-point data representation effects are shown.

2. PWAS VIRTUAL SENSOR

In this section the PWAS virtual sensor approach is briefly summarized.

Consider the following nonlinear discrete-time dynamical model:

$$\begin{aligned} x_{k+1} &= g(x_k, u_k) \\ y_k &= h_y(x_k) \\ z_k &= h_z(x_k) \end{aligned}$$

where $x \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the exogenous input vector of manipulated variables, $y \in \mathbb{R}^{n_y}$ is the vector of measurable outputs, and k denotes the discrete-time instant. Vector $z \in \mathbb{R}^{n_z}$ collects a set of variables to be estimated. We assume that the vector z_k can be measured by a *real sensor* for $k = 0, \dots, K-1$. A portion of these measurements, from $k = 0$ to $k = K_t - 1$, $K_t < K$, is considered as a *training set* and is used to design the virtual sensor, which will operate without measuring z_k ; the remaining data, from $k = K_t$ to $k = K-1$ is used as a *validation set* to verify the estimation capabilities of the virtual sensor. We aim to construct a virtual sensor that estimates z_k for $k \geq K_t$, when the real sensor is no more available. The functions $g(\cdot, \cdot): \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$, $h_y(\cdot): \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ and $h_z(\cdot): \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_z}$ are assumed to be unknown. For the sake of simplicity, we assume $n_z = 1$; non-scalar z can be easily estimated component-wise.

Since the actual variables y , u and z at past time instants are not available, the noisy measurements of them are assumed to be:

$$\begin{aligned} \tilde{u}_k &= u_k + \eta_{u,k} & k \geq 0 \\ \tilde{y}_k &= y_k + \eta_{y,k} & k \geq 0 \\ \tilde{z}_k &= z_k + \eta_{z,k} & 0 \leq k < K \end{aligned}$$

where η_u , η_y and η_z are unmeasured stochastic variables. For given values of M_u , M_y and M_z , the inputs of the virtual sensor will be noisy sequences of measurements of y and u , and a vector of past values of \hat{z} , namely

$$\begin{aligned}\tilde{U}_k &\triangleq [\tilde{u}_{k-M_u+1}^T \quad \tilde{u}_{k-M_u+2}^T \quad \cdots \quad \tilde{u}_k^T]^T \\ \tilde{Y}_k &\triangleq [\tilde{y}_{k-M_y+1}^T \quad \tilde{y}_{k-M_y+2}^T \quad \cdots \quad \tilde{y}_k^T]^T \\ \hat{Z}_k &\triangleq [\hat{z}_{k-M_z}^T \quad \hat{z}_{k-M_z+1}^T \quad \cdots \quad \hat{z}_{k-1}^T]^T\end{aligned}$$

The values of M_u , M_y , M_z are generally considered as tuning parameters.

2.1. Standard virtual sensor

The standard virtual sensor is obtained by estimating z_k in the following way:

$$\hat{z}_k = f_{PWAS}(\tilde{U}_k, \tilde{Y}_k, \hat{Z}_k) \quad (1)$$

f_{PWAS} being a PWA function defined over a simplicial partition made up of N_s vertices and represented through the α -basis, i.e.,

$$f_{PWAS}(x) = \sum_{i=1}^{N_s} w_i \alpha_i(x). \quad (2)$$

By denoting as v_j , $j = 1, \dots, N_s$ the vertices of the simplicial partition, functions $\alpha_i(x)$ are PWAS functions such that

$$\begin{aligned}\alpha_i(v_j) &= 1, \text{ if } i = j \\ \alpha_i(v_j) &= 0, \text{ if } i \neq j\end{aligned}$$

Once the basis functions and the simplicial partition are selected, weights w_i uniquely define f_{PWAS} and, in case the α -basis is used, they correspond to the value of the function in the partition vertices, i.e., $w_i = f_{PWAS}(v_i)$. A detailed discussion about PWAS functions can be found in [9].

Weights vector $w = [w_1 \dots w_{N_s}]^T$ is obtained by solving the following optimization problem:

$$\min_w \left\{ \sum_{k=M}^{K_t-1} [\hat{z}_k - f_{PWAS}(\tilde{U}_k, \tilde{Y}_k, \hat{Z}_k)]^2 + \sigma w^T \Gamma w \right\} \quad (3)$$

K_t being the number of elements in the training set, σ the Tikhonov regularization parameter and $M = \max(M_u, M_y, M_z)$.

The first term can be reformulated as a quadratic function of w and takes into account the square error between the value of the PWAS function (i.e., the estimated data) and the actual data \hat{z}_k . The second term performs a Tikhonov regularization that depends on the structure of Γ . In the simplest case, $\Gamma = I$ provides the zero-order Tikhonov regularization. First-order (or higher order) Tikhonov regularizations can be obtained alternatively by considering the gradient of the PWAS function f_{PWAS} (or higher order derivatives) in constructing Γ . The choice of the regularization parameter σ is quite critical since it can highly influence the performances of the virtual sensor. A low value may lead to an ill-conditioned problem, i.e. the solution is sensitive to small changes in data. A high value may lead to an inaccurate estimate of the unmeasurable output.

The regularized least squares problem (3) can be recast as an unconstrained quadratic programming (QP) problem in the form:

$$\min_w w^T H w + \frac{1}{2} f^T w \quad (4)$$

A rigorous convergence analysis of the standard virtual sensor is reported in 0.

2.2. Reduced-complexity virtual sensor

The reduced-complexity virtual sensor approach expresses the estimate \hat{z} as a sum of lower-dimensional PWAS functions, in order to mitigate the effects of the curse of dimensionality, which prevents the applicability of the standard virtual sensor in many applications. For the sake of compactness, henceforth the input of the virtual sensor is referred to as

$$\Xi_k \triangleq [\tilde{U}_k^T \quad \tilde{Y}_k^T \quad \hat{Z}_k^T]^T \quad (5)$$

where $n_\xi \triangleq M_u n_u + M_y n_y + M_z$. Assume to split vector Ξ into $\nu \in \mathbb{N}$ subsets $\Xi_1, \Xi_2, \dots, \Xi_\nu$, such that all elements of Ξ are included in one and only one of these subsets. Each subset Ξ_j ($j = 1, \dots, \nu$) has dimension equal to n_j , such that $1 \leq n_j \leq n_\xi$ and $n_1 + n_2 + \dots + n_\nu = n_\xi$. The n_j elements of each Ξ_j are denoted as $\xi_{j,1}, \xi_{j,2}, \dots, \xi_{j,n_j}$. The proposed reduced-complexity virtual sensor is defined through a sum of PWAS functions f_{PWAS}^j , $j = 1, \dots, \nu$, each being the weighted sum of N_j PWAS basis functions:

$$\hat{z}_k = f_{PWAS}(\Xi_k) = \sum_{j=1}^{\nu} f_{PWAS}^j(\Xi_j) = \sum_{j=1}^{\nu} \sum_{i=1}^{N_j} w_{j,i} \alpha_{j,i}(\Xi_k^j) \quad (6)$$

where $f_{PWAS}: \mathbb{R}^{n_\xi} \rightarrow \mathbb{R}$ (for fixed w), $\alpha_{j,i}$ denotes the i -th α -basis of the j -th PWAS function and N_j is the number of basis functions in each domain Ξ^j . Also,

$$w \triangleq [w_{1,1} \dots w_{1,N_1} \quad w_{2,1} \dots w_{2,N_2} \quad \cdots \quad w_{\nu,1} \dots w_{\nu,N_\nu}]^T \quad (7)$$

The vector of parameters w (which determines the shape of f_{PWAS}) is obtained, as for the standard virtual sensor, by solving the least-squares problem

$$\min_w \left\{ \sum_{k=M}^{K_t-1} [\hat{z}_k - f_{PWAS}(\Xi_k)]^2 + \sigma w^T \Gamma w \right\} \quad (8)$$

The regularized least squares problem (8) can be recast again as an unconstrained QP problem in the form (4). A rigorous convergence analysis of the reduced-complexity virtual sensor is reported in [7].

3. DESIGN OF THE SPECIFIC VIRTUAL SENSOR

In this section, the specific virtual sensor design is described.

3.1. Virtual sensor architecture

We exploited the capabilities of MOBY-DIC Toolbox for MATLAB [10] to train a reduced-complexity PWAS virtual sensor for the cable-length adaptive estimation of the motor-side current, starting from measurements of the drive-side current. The same tool also allowed the automatic generation of VHDL files describing a circuit architecture for the FPGA implementation of the sensor. The block scheme of the architecture, described in detail in [7], is shown in Figure 3. At each sampling time, the values of the system inputs \tilde{u}_k and measurable outputs \tilde{y}_k are loaded into First-In-First-Out (FIFO) blocks which behave as buffers, storing the data at current and past time instants (i.e. \tilde{U}_k^T and \tilde{Y}_k^T). Once the FIFO

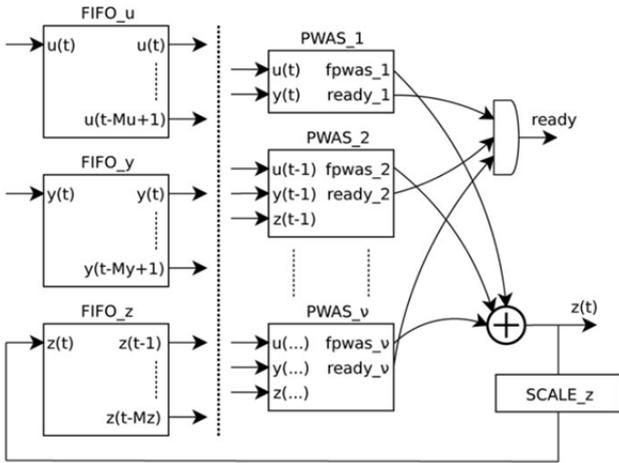


Figure 3. Block scheme of the circuit implementing the reduced-complexity virtual sensor.

blocks are full, the computation of the estimated output is performed by the $PWAS_i$ ($i = 1, \dots, v$) blocks, which are responsible for the evaluation of the v PWAS functions. A detailed description of these blocks is available in [11]. Each $PWAS_i$ block communicates the end of its computation with a $ready_i$ signal; as soon as all $ready_i$ signals are active, the results of each function evaluation ($fpwas_i$) are added up by an adder block, which provides the estimation of the unmeasurable output z_k , and a global ready signal is set to logic value '1' indicating that the result is available. If $M_z > 0$, the estimation is brought back to the $FIFO_z$ block and is used as input for next estimations. An affine scaling of \hat{z}_k (performed by block $SCALE_z$) is necessary to bring the value to the correct range needed by circuit inputs. The architecture, has a latency of 9 clock cycles (i.e., 180 ns at 50 MHz).

In our case we have $n_u = 0$, $n_y = 2$ and $n_z = 1$, in particular y contains the drive-side current and the cable length and z is the motor-side current. We designed the sensor by heuristically setting $M_y = 1$ and $M_z = 2$. The domain of the PWAS functions constituting the virtual sensor has been partitioned by using two subdivisions along the axis representing the drive-side current and only one subdivision along the axis representing the cable length.

3.2. Experimental dataset

A dataset made up of 2,000,000 samples of the drive-side current and the corresponding motor-side current, sampled at 500 kS/s was employed. The samples correspond to 4 different cable lengths: 180 m, 360 m, 540 m and 720 m (500,000 samples for each length). Figure 4 shows the drive-side currents for the different cable lengths.

In order to acquire simultaneously both the drive-side and motor-side currents, two channels of an 8-bit oscilloscope were used to oversample the signals at 10 MS/s to avoid aliasing. Current probes based on combined Hall Effect and Current Transformer technologies were used as the sensors. The scope has an effective resolution of 64 mA, which corresponds to a quantization noise variance of $0.064^2/12 = 0.000341$ A². The probes have a 3σ accuracy of 0.25 A. Considering the noise sources to be independent, the combined variance can be found by summing the individual variances in quadrature. This sum leads to an effective standard deviation of 85.4 mA, which can

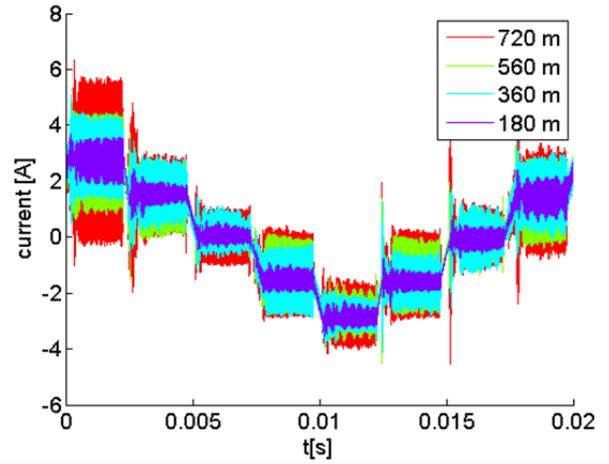


Figure 4. Drive-side currents for different cable lengths.

be considered acceptable compared to the $2\sqrt{2}$ A amplitude of current waveform.

The 10 MS/s sampled data is then low pass filtered to attenuate all frequency components above 250 kHz to below the quantization level before downsampling the signals to 500 kS/s. This lower rate is high enough to capture the significant PWM-excited harmonics, but low enough to be sampled by standard drive ADCs, using analog anti-aliasing, thereby allowing real-time estimator implementation.

The samples corresponding to cable lengths of 180 m, 360 m and 540 m have been used as a training set, to derive the weights w . The remaining 500,000 samples (corresponding to cable length 720 m) have been used to validate the virtual sensor.

The weights w defining the PWAS virtual sensor, see (6), are shown in Table 1.

4. RESULTS

Figures 5 and 6 show the measured motor-side current of the validation dataset (blue) and the estimated current obtained by the FPGA in fixed-point precision (red). The root mean squared (RMS) estimation error obtained is 0.0858 A in fixed point precision and 0.0794 A in MATLAB double precision, whilst the overall current waveform has an RMS value of 2 A. In [8] a lower RMSE value was obtained by employing the PWAS virtual sensor, nevertheless that sensor was trained with measurements taken with a 720 m long cable, while here the training is performed with data referred to different cable lengths, and the cable length is a sensor input. This solution is therefore more general, since it is cable-length-adaptive.

The motor-side estimated current obtained with the

Table 1: Weights of the three PWAS functions constituting the reduced-complexity virtual sensor.

$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$w_{1,4}$	$w_{1,5}$	$w_{1,6}$
-0.3474	-0.1560	0.6341	0.2473	-0.1559	0.0123

$w_{1,1}$	$w_{1,2}$	$w_{1,3}$
-9.9865	0.1665	10.0544

$w_{1,1}$	$w_{1,2}$	$w_{1,3}$
3.9596	-0.0135	-3.7117

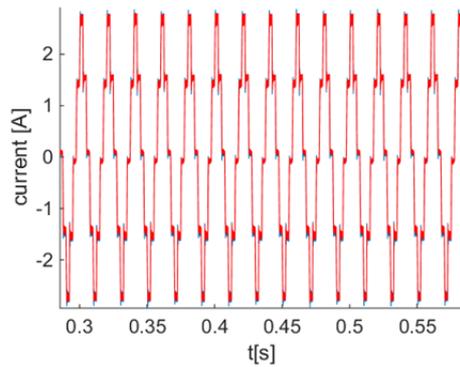


Figure 5. Motor-side current: measured (blue), estimated in fixed-point precision (red).

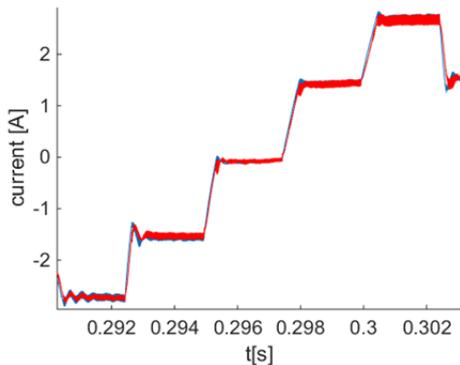


Figure 6. Zoom of Figure 5.

estimator in [2] in MATLAB double precision is shown in Figure 7 for comparison. The RMS estimation error is 0.0454 A. Despite being lower than that obtained with the proposed method, it requires significantly more effort to obtain the estimator. Depending on the application, this extra effort may not be worth the small increase in estimation precision.

Figure 8 shows the estimation errors for both estimators, PWAS virtual sensor (blue curve) and estimator in [2] (red curve). Mean value and variance are -0.017 and 0.007, respectively, for the PWAS case and $2.5 \cdot 10^{-3}$ and $2.1 \cdot 10^{-3}$, respectively, for the case in [2].

5. CONCLUSIONS

We designed a piecewise-affine virtual sensor for the estimation of motor-side current of hybrid stepper motors,

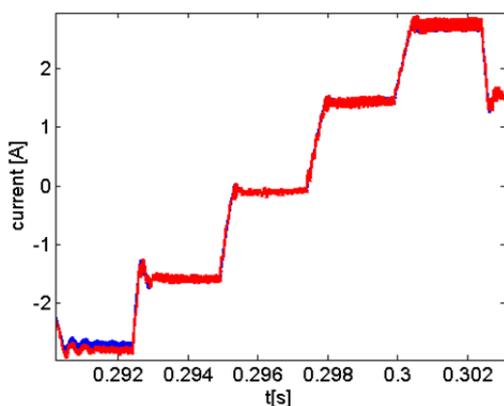


Figure 7. Motor-side measured current (blue) and current estimated with the estimator in [2] (red).

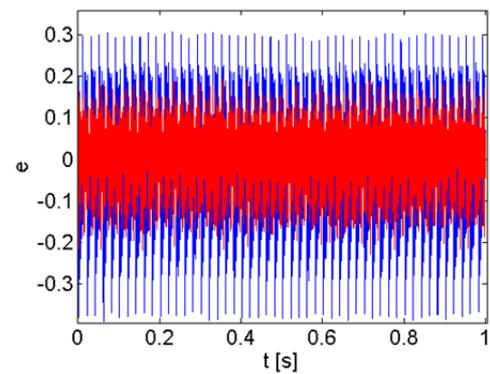


Figure 8. Estimation error for the estimator in [2] (red) and for the PWAS virtual sensor (blue).

starting from measurements of drive-side currents through cables of different length. This sensor is therefore cable-length-adaptive. A circuit architecture has also been designed for the implementation of the virtual sensor. The proposed solution exhibits slightly poorer performances compared with the ones obtained by standard model-based estimators, but results in a much faster and simpler design, since the identification of a model is not required. Furthermore, even this slightly higher estimation error is negligible compared to the waveform amplitude and so allows the desired current levels to be tracked such that a negligible effect occurs on the motor position and torque. This latter is usually chosen to include a significant safety margin anyway.

REFERENCES

- [1] P. Acamley, "Stepping motors: a guide to theory and practice. Control Eng", Series 63, 4-th edition, The Institution of Electrical Engineers, London, UK, 2002.
- [2] A. Masi, M. Butcher, M. Martino, and R. Picatoste, "An application of the extended Kalman filter for a sensorless stepper motor drive working with long cables", *IEEE Transactions on Industrial Electronics*, 2012, 59(11), pp. 4217–4225.
- [3] M. Milanese, C. Novara, K. Hsu, and K. Poolla, "Filter design from data: direct vs. two-step approaches", *Proceedings of the American Control Conference (ACC)*, 14-16 June 2006, Minneapolis, MN, USA, pp. 4466–4470.
- [4] L. Ljung, "Convergence analysis of parametric identification methods", *IEEE Transactions on Automatic Control*, 1978, 23(5), pp. 770–783.
- [5] T. Poggi, M. Rubagotti, A. Bemporad, and M. Storaice, "High-speed piecewise affine virtual sensors", *IEEE Transactions on Industrial Electronics*, 2012, 59(2), pp. 1228–1237.
- [6] M. Rubagotti, T. Poggi, A. Bemporad, and M. Storaice, "Piecewise affine direct virtual sensors with reduced complexity", *Proceedings of the 2012 IEEE Conference on Decision and Control (CDC)*, Maui, Hawaii (USA), 10-13 Dec. 2012, pp. 656–661.
- [7] M. Rubagotti, T. Poggi, A. Oliveri, C. A. Pascucci, A. Bemporad, and M. Storaice, "Low-complexity piecewise-affine virtual sensors: theory and design", *International Journal of Control*, 2014, 87(3), pp. 622–632.
- [8] A. Oliveri, M. Butcher, A. Masi, and M. Storaice, "Piecewise affine virtual sensor: a case study-estimation of stepping motor current from long distances", *Proceedings of the 20th IMEKO TC4 International Symposium and 18th International Workshop on ADC Modelling and Testing Research on Electric and Electronic Measurement for the Economic Upturn*. Benevento, Italy, 15-17 Sep. 2014.
- [9] M. Parodi, M. Storaice, and P. Julián, "Synthesis of multiport resistors with piecewise-linear characteristics: a mixed-signal

- architecture”, *International Journal of Circuit Theory and Applications*, 2005, 33(4), pp. 307–319.
- [10] A. Oliveri, D. Barcelli, A. Bemporad, B. A. G. Genuit, W. P. M. H. Heemels, T. Poggi, M. Rubagotti, and M. Storace, “MOBY-DIC: A MATLAB toolbox for circuit-oriented design of explicit MPC”, *Proceedings of the 4th IFAC Conference on Nonlinear Model Predictive Control (NMPC)*, Noordwijkerhout (The Netherlands), 23-27 Aug. 2012, pp 218–225.
- [11] M. Storace, T. Poggi, “Digital architectures realizing piecewise-linear multivariate functions: Two FPGA implementations”, *International Journal of Circuit Theory and Applications*, 2011, 39, pp. 1–15.