High-Speed Piecewise Affine Virtual Sensors

Tomaso Poggi, Matteo Rubagotti, *Member, IEEE*, Alberto Bemporad, *Fellow, IEEE*, and Marco Storace, *Member, IEEE*

***Abstract*—This paper proposes piecewise affine (PWA) virtual sensors for the estimation of unmeasured variables of nonlinear systems with unknown dynamics. The estimation functions are designed directly from measured inputs and outputs and have two important features. First, they enjoy convergence and optimality properties, based on classical results on parametric identification. Second, the PWA structure is based on a simplicial partition of the measurement space and allows one to implement very effectively the virtual sensor on a digital circuit. Due to the low cost of the required hardware for the implementation of such a particular structure and to the very high sampling frequencies that can be achieved, the approach is applicable to a wide range of industrial problems.**

***Index Terms*—Digital circuits, nonlinear observers, piecewise affine (PWA) functions, virtual sensors.**

I. Introduction

ESTIMATING the internal states of a dynamical system from measurements is one of the basic problems in control theory. For linear dynamical systems, it is well known that, in case of stochastic noise affecting measurements and dynamics, the Kalman filter provides optimal estimates. On the other hand, for nonlinear systems, the optimal filter is, in general, very difficult (or impossible) to derive, and one must rely on approximate solutions, such as extended Kalman filters [1], unscented Kalman filters [2], ensemble Kalman filters [3], and particle filters [4]. For these solutions, a model of the system must be available; however, in many practical applications, realistic models are not available, and one has to solve a problem of *filter design from data*. The standard procedure to address this problem is to first obtain a model by using system identification and then design an observer based on the resulting model to obtain estimates. We will refer to this strategy as the *two-step procedure*.

When designing a filter from data, the overall performance is usually far from optimal, and alternative strategies were proposed. In particular, we highlight the design method proposed in [5], which is a direct (one-step) procedure for designing an optimal filter, is applicable to nonlinear systems, and is proven to be the minimum variance estimator among the selected class of approximating filters. We refer to the observer obtained using the direct procedure as the *virtual sensor* (cf. [6]). The virtual sensor is a function of past inputs and measured outputs, and no model of the system is required. Only the observability of the variables to be estimated is assumed as a necessary condition. Apart from the contribution [5], where a stochastic framework is considered, virtual sensors for nonlinear systems have been proposed also in [7] using a set membership approach and applied to relevant industrial problems in [8]–[11]. The procedure described in [5] relies on choosing a suitable set of basis functions, which ensures that the resulting virtual sensor satisfies the assumptions required to apply the theoretical results in [12].

In this paper, we propose piecewise affine (PWA) simplicial (PWAS) functions 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 circuits such as field-programmable gate arrays (FPGAs), providing low power consumption, fast response times, and, at least for high-volume applications, low cost [13]. In particular, the use of FPGAs has proven to be very effective in embedded systems in different fields of application [14], [15]. PWA functions have been used extensively in the last few years for control design (see, e.g., [16] and [17]) but surprisingly not much for estimation purposes. The contribution of this paper is to introduce PWAS virtual sensors in the framework of one-step filters, showing that they enjoy all the convergence and optimality properties of the general approach described in [5]. Their main practical advantage is the possibility to implement them on low-cost hardware (in particular, FPGAs), obtaining short execution times on the order of tens of nanoseconds.

The approach of this paper is extremely useful in at least three types of applications:

1) when a real sensor, after being used for experiments, is too expensive or complex to be deployed in a product;
2) in safety-critical applications, where redundancy of measurements is used to detect sensor failures;
3) when a classical observer based on the two-step procedure is too imprecise, slow, expensive, or power demanding to be implemented.

The proposed approach is tested in two case studies. The first one is a discrete-time version of Lorenz's system, whose parameters are set to make the dynamics chaotic, that we chose to compare our results with the ones in [5]. The

Manuscript received December 21, 2010; revised April 12, 2011; accepted June 10, 2011. Date of publication June 30, 2011; date of current version October 18, 2011. This work was supported in part by the European Commission through Project MOBY-DIC "Model-based synthesis of digital electronic circuits for embedded control" under Grant FP7-INFSO-ICT-248858, http://www.mobydic-project.eu.

T. Poggi and M. Storace are with the Biophysical and Electronic Engineering Department, University of Genoa, 16145 Genova, Italy (e-mail: tomaso.poggi@unige.it; marco.storace@unige.it).

M. Rubagotti is with the Department of Mechanical and Structural Engineering, University of Trento, 38100 Trento, Italy (e-mail: rubagotti@ieee.org).

A. Bemporad is with the Institutions, Markets, Technologies (IMT) Institute for Advanced Studies, 55100 Lucca, Italy (e-mail: bemporad@imtlucca.it).

Digital Object Identifier 10.1109/TIE.2011.2161064

0278-0046/$26.00 © 2011 IEEE

second one concerns the control of a laboratory helicopter model and is based on experimental data. In both cases, the resulting FPGA implementation provides good estimation accuracy and good circuit features, in terms of real-time processing times (on the order of tens of nanoseconds) and circuit size.

This paper is organized as follows. Section II introduces the estimation problem. The PWAS virtual sensor is presented in Section III. In Section IV, the convergence properties of the proposed scheme are proven, while the issues related to its practical implementation are discussed in Section V. The two case studies are presented in Section VI, and some conclusions are gathered in Section VII.

II. System Description

Consider the nonlinear discrete-time dynamical model $\mathcal{S}$

$$\mathcal{S}: \begin{cases} x(t+1) = g\left(x(t), u(t)\right) \\ y(t) = h_y\left(x(t)\right) \\ z(t) = h_z\left(x(t)\right) \end{cases} \tag{1}$$

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 t is the discrete-time instant. Vector $z \in \mathbb{R}^{n_z}$ collects a set of variables to be estimated. We assume that the vector $z(t)$ can be measured by a *real sensor* at time instants $t = 0, \ldots, T$. Then, we aim to construct a *virtual sensor* that estimates $z(t)$ for $t > T$. 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 $n_z = 1$; nonscalar z can be estimated componentwise using the techniques presented in this paper.

The available measurements of u, y, and z are assumed to be affected by noise

$$\tilde{u}(t) = u(t) + \eta_u(t) \quad t \geq 0$$

$$\tilde{y}(t) = y(t) + \eta_y(t) \quad t \geq 0$$

$$\tilde{z}(t) = z(t) + \eta_z(t) \quad 0 \leq t \leq T$$

where η_u, η_y, and η_z are unmeasured stochastic variables.

The notion of the observability of nonlinear systems is well developed (see, e.g., [18] and [19]); here, we follow the definitions in [5] that we briefly recall in Section IV. As highlighted in [5], observability implies that the state vector x can be uniquely determined by collecting a finite number M_u ($M_u \geq 0$) of samples of u and M_y ($1 \leq M_y \leq n_x$) samples of y. Therefore, if the system defined by (g, h_y) is *observable*, then there exists a function f_z (which can be obtained from a system model, if known) such that

$$z(t) = f_z\left(U(t), Y(t)\right)$$

with

$$U(t) \triangleq [u(t-M_u+1)' \quad u(t-M_u+2)' \quad \ldots \quad u(t)']'$$

$$Y(t) \triangleq [y(t-M_y+1)' \quad y(t-M_y+2)' \quad \ldots \quad y(t)']'$$

where $'$ is the transposition operator. The reader is referred to [7] for the detailed proof of this result. In case of data perturbed by noise, we define[1]

$$\tilde{U}(t) \triangleq [\tilde{u}(t-M_u+1)' \quad \tilde{u}(t-M_u+2)' \quad \ldots \quad \tilde{u}(t)']'$$

$$\tilde{Y}(t) \triangleq [\tilde{y}(t-M_y+1)' \quad \tilde{y}(t-M_y+2)' \quad \ldots \quad \tilde{y}(t)']'.$$

The aforementioned setup, in which the system (1) is unknown, covers a wide range of applications.

III. Virtual Sensor

The proposed virtual sensor is referred to as $\mathcal{V}_\alpha(w)$ and is obtained by estimating z in the following way:

$$\hat{z}(t) = \sum_{k=1}^{N} w_k \alpha_k \left(\tilde{U}(t), \tilde{Y}(t)\right) \triangleq f_\alpha \left(\tilde{U}(t), \tilde{Y}(t); w\right) \tag{2}$$

where w is a vector of parameters to be determined, $f_\alpha : \mathbb{R}^{n_\xi} \rightarrow \mathbb{R}$ (for fixed w), with $\xi = \left[\begin{smallmatrix}\tilde{U} \\ \tilde{Y}\end{smallmatrix}\right]$ and $n_\xi \triangleq M_u n_u + M_y n_y$, and $\{\alpha_k\}_{k=1}^{N}$ is a basis of functions that is described hereinafter. Vector $w \in D_w \subset \mathbb{R}^N$ collects the corresponding weights, with D_w being a compact set, determined by solving the least squares problem

$$w^* = \arg\min_{w} \left\{ \sum_{t=M}^{T} \left[\tilde{z}(t) - f_\alpha\left(\tilde{U}(t), \tilde{Y}(t); w\right)\right]^2 \right\} \tag{3}$$

where $M = \max(M_u, M_y)$, $M \ll T$. The results in [12] show that, by properly choosing the basis functions, the estimation error $z(t) - \hat{z}(t)$ converges toward zero in a "better way" than those obtainable with a two-step procedure (see Theorem 1 hereinafter).

For the efficient circuit implementation of the virtual observer (2), we consider a class of continuous and regular PWA basis functions, defined over a regular partition of the (closed and convex) hyperrectangular domain

$$S = \{\xi \in \mathbb{R}^{n_\xi} : \xi_{minj} \leq \xi_j \leq \xi_{maxj}, \quad = 1, \ldots, n_\xi\} \tag{4}$$

into a set of regular *simplices*. The functions that can be obtained by combining the elements of this basis are called *PWAS functions*. Each basis function α_k is defined over the domain S.

Definition 1 (Simplex): Given a set of $n_\xi + 1$ points $\xi_i^0, \xi_i^1, \ldots, \xi_i^{n_\xi} \in \mathbb{R}^{n_\xi}$, called *vertices*, a *simplex* S_i in $\mathbb{R}^{n_\xi}$ is their convex combination

$$S_i = \left\{ \xi \in \mathbb{R}^{n_\xi} : \xi = \sum_{j=0}^{n_\xi} \mu_j \xi_i^j, \ 0 \leq \mu_j \leq 1, \ \sum_{j=0}^{n_\xi} \mu_j = 1 \right\}.$$

For $n_\xi = 1$, a simplex is a segment, for $n_\xi = 2$ a triangle, for $n_\xi = 3$ a tetrahedron, and, more generally, a n_ξ-dimensional hypertriangle. The domain S is partitioned into simplices as follows. Every dimensional component $\xi_j \in [\xi_{minj}, \xi_{maxj}]$ of

[1] Vectors $\tilde{U}(t)$ and $U(t)$ are empty vectors in case the system is autonomous, i.e., for $n_u = M_u = 0$.

S is divided into p_j subintervals of length $(\xi_{maxj} - \xi_{minj})/p_j$. Consequently, the domain is divided into $\prod_{j=1}^{n_\xi} p_j$ hyperrectangles and contains $N = \prod_{j=1}^{n_\xi}(p_j + 1)$ vertices v_k. Each rectangle is further partitioned into $n_\xi! = \prod_{i=1}^{n_\xi} i$ simplices by resorting to the algorithm proposed in [20]; thus, S contains $L = n_\xi! \prod_{j=1}^{n_\xi} p_j$ simplices S_i such that $S = \cup_{i=0}^{L-1} S_i$ and $\overset{\circ}{S}_i \cap \overset{\circ}{S}_j = \emptyset, \forall i, j = 0, \ldots, L-1, i \neq j$, with $\overset{\circ}{S}_i$ denoting the interior of S_i. The resulting partition is called *simplicial partition* or *type-1 triangulation* and is univocally identified by vector p. The corresponding class of continuous functions that are affine over each simplex constitutes an N-dimensional linear space $PWAS_p[S] \subset C^0[S]$ [21], where $C^0[S]$ denotes the space of continuous functions defined over S. Therefore, it is possible to define different bases, made up of N linearly independent functions belonging to $PWAS_p[S]$. By choosing some (arbitrary) ordering of the functions of any of these bases, we can regard them as an N-length vector. Then, a scalar PWAS function $f_\alpha \in PWAS_p[S]$ is defined as a linear combination of the basis functions as in (2). The coefficients w_k, collected into the vector w, determine uniquely f_α for each given $\xi \in S$.

Different types of basis functions can be defined; in this paper, we refer to the so-called α-basis [22]. The function $\alpha_k(\xi)$ is a PWAS hyperpyramid (a pyramid if $n_\xi = 2$), which takes the value 1 at v_k and 0 at all the other vertices. Hence, every element of the basis has a local nature, is affine over each simplex, and satisfies the conditions, $0 \leq \alpha_k(\xi) \leq 1$, $\forall \xi \in S$, and

$$\alpha_k(v_h) = \delta_{h,k} = \begin{cases} 1 & \text{if } h = k \\ 0 & \text{if } h \neq k \end{cases}$$

where $\delta_{h,k}$ is the Kronecker's delta. Of course, other bases can be defined and used: The interested reader is referred to [23] and [24] for some examples.

A. Circuit Implementation

The circuit implementation of the proposed virtual sensor is made up of two blocks: a bank of registers to store past values of $\tilde{u}(t)$ and $\tilde{y}(t)$ (i.e., the vectors $\tilde{U}(t)$ and $\tilde{Y}(t)$) and an arithmetic unit to calculate the value of the PWAS function f_α.

Whatever basis one chooses, a PWAS function can be implemented in a circuit by using linear interpolators. Indeed, given a n_ξ-dimensional input vector ξ, the value of a PWAS function can be obtained by linearly interpolating only the $n_\xi + 1$ values assumed by the function at the vertices defining the simplex that the input vector belongs to. The algorithm usually adopted to locate such a simplex is based on Kuhn's lemmas [20] and is optimal with respect to the number of inputs [25]. Some examples of digital circuit solutions for fast piecewise-linear interpolation can be found in [13], [26], and [27].

We remark that speed is just one of the key features of the proposed circuit implementation. Other important features are related to the possibility of implementing the proposed architectures in embedded systems, with low-cost hardware resources.

IV. Convergence Analysis of the Estimation Error

This section proves the convergence properties of the proposed virtual sensor. After recalling from [5] the properties that a generic virtual sensor must enjoy to ensure convergence, we prove that the PWAS virtual sensor (2) satisfies such properties.

Consider the minimum variance filter $\mathcal{K}(\theta_\mathcal{M})$ obtained by using a two-step procedure: 1) by estimating a state-space model of (1) from a set of available measurements and 2) by designing an observer based on such a model. The filter $\mathcal{K}(\theta_\mathcal{M})$ is based on the parameter set $\theta_\mathcal{M} \in D_{\theta_\mathcal{M}}$, where $D_{\theta_\mathcal{M}}$ is a compact set, and designed relying on a class of models $\mathcal{M}(\theta_\mathcal{M})$ of system (1). In particular, consider the model $\mathcal{M}(\theta_\mathcal{M}^*)$, identified in accordance with a prediction error method from a set of available data $\tilde{u}(t)$, $\tilde{y}(t)$, and $\tilde{z}(t)$ and the corresponding filter realization $\mathcal{K}(\theta_\mathcal{M}^*)$. The estimate produced by $\mathcal{K}(\theta_\mathcal{M}^*)$ can be represented in regression form as

$$\hat{z}_\mathcal{K}(t) = f_\mathcal{K}\left(\theta_\mathcal{M}, \tilde{U}(t), \tilde{Y}(t), \hat{Z}(t)\right)$$

where

$$\hat{Z}(t) \triangleq [\hat{z}(t - M_z)' \quad \hat{z}(t - M_z + 1)' \quad \ldots \quad \hat{z}(t-1)']'$$

with $t > T$, $M_z \geq 0$ ($M_z = 0$ means that $\hat{Z}(t)$ is an empty vector).

The estimates $\hat{z}_\mathcal{V}$ produced by a generic virtual sensor $\mathcal{V}(\theta_\mathcal{V})$ (which is obtained by applying the direct approach) are

$$\hat{z}_\mathcal{V}(t) = f_\mathcal{V}\left(\theta_\mathcal{V}, \tilde{U}(t), \tilde{Y}(t), \hat{Z}(t)\right) \tag{5}$$

where $t > T$, $\theta_\mathcal{V} \in D_{\theta_\mathcal{V}}$, where $D_{\theta_\mathcal{V}}$ is a compact set. An optimal realization $\mathcal{V}(\theta_\mathcal{V}^*)$ of the virtual sensor is obtained by minimizing a quadratic criterion similar to (3)

$$\theta_\mathcal{V}^* = \arg\min_{\theta_\mathcal{V}} \left\{ \sum_{t=M}^{T} \left[\tilde{z}(t) - f_\mathcal{V}\left(\theta_\mathcal{V}, \tilde{U}(t), \tilde{Y}(t), \hat{Z}(t)\right)\right]^2 \right\} \tag{6}$$

with $M \ll T$. Note that the PWA virtual sensor (2) is a special case of (5), based on PWAS basis functions and with $M_z = 0$.

The following result describes the convergence properties of $\mathcal{V}(\theta_\mathcal{V}^*)$ compared to $\mathcal{K}(\theta_\mathcal{M}^*)$.

Theorem 1 ([5] and [12]): Consider system (1), and assume that (g, h_y) is observable. Consider a minimum variance filter $\mathcal{K}(\theta_\mathcal{M}^*)$ and a virtual sensor $\mathcal{V}(\theta_\mathcal{V}^*)$ whose parameters are obtained by solving problem (6). Assume that there exist two scalars $C > 0$ and λ, $0 < \lambda < 1$, such that the following conditions hold.

1) The estimate is limited at origin, i.e.,

$$\left| f_\mathcal{V}\left(\theta_\mathcal{V}, \tilde{U}_0(t), \tilde{Y}_0(t), \hat{Z}_0(t)\right)\right| \leq C \tag{7}$$

where $\tilde{U}_0$, $\tilde{Y}_0$, and $\hat{Z}_0$ are zero vectors of dimensions $M_u n_u$, $M_y n_y$, and M_z, respectively.

2) The virtual sensor (5) has exponential fading memory, i.e., considering two different realizations of vectors

$\tilde{U}(t)$, $\tilde{Y}(t)$, and $\hat{Z}(t)$ named $\tilde{U}_1(t)$, $\tilde{Y}_1(t)$, and $\hat{Z}_1(t)$ and $\tilde{U}_2(t)$, $\tilde{Y}_2(t)$, and $\hat{Z}_2(t)$, respectively, it yields

$$\begin{aligned}\left| f_{\mathcal{V}}\left(\theta_{\mathcal{V}}, \tilde{U}_1(t), \tilde{Y}_1(t), \hat{Z}_1(t)\right) - f_{\mathcal{V}}\left(\theta_{\mathcal{V}}, \tilde{U}_2(t), \tilde{Y}_2(t), \hat{Z}_2(t)\right)\right| \\ \leq C \sum_{s=0}^{t} \lambda^{t-s} \left[\|\tilde{u}_1(s) - \tilde{u}_2(s)\|_1 + \|\tilde{y}_1(s) - \tilde{y}_2(s)\|_1 \right. \\ \left. + \|\hat{z}_1(s) - \hat{z}_2(s)\|_1\right] \qquad (8)\end{aligned}$$

where $\|\cdot\|_1$ denotes the 1-norm in the Euclidean space. This condition requires that the remote past inputs of the virtual sensor are forgotten at an exponential rate.

3) The function $f_{\mathcal{V}}(\theta_{\mathcal{V}}, \tilde{U}(t), \tilde{Y}(t), \hat{Z}(t))$ is differentiable with respect to $\theta_{\mathcal{V}}$ for all $\theta \in D_{\theta_{\mathcal{V}}}$, and its gradient $\nabla_{\theta_{\mathcal{V}}} f_{\mathcal{V}}(\theta_{\mathcal{V}}, \tilde{U}(t), \tilde{Y}(t), \hat{Z}(t))$ fulfills the following exponential fading property:

$$\begin{aligned}\left\| \nabla_{\theta_{\mathcal{V}}} f_{\mathcal{V}}\left(\theta_{\mathcal{V}}, \tilde{U}_1(t), \tilde{Y}_1(t), \hat{Z}_1(t)\right)\right. \\ \left. - \nabla_{\theta_{\mathcal{V}}} f_{\mathcal{V}}\left(\theta_{\mathcal{V}}, \tilde{U}_2(t), \tilde{Y}_2(t), \hat{Z}_2(t)\right)\right\|_1 \\ \leq C \sum_{s=0}^{t} \lambda^{t-s} \left[\|\tilde{u}_1(s) - \tilde{u}_2(s)\|_1 + \|\tilde{y}_1(s) - \tilde{y}_2(s)\|_1 \right. \\ \left. + \|\hat{z}_1(s) - \hat{z}_2(s)\|_1\right]. \qquad (9)\end{aligned}$$

Then, denoting the statistical expectation by $E[\cdot]$, the following results hold with probability 1 as $T \to \infty$.

i) The vector of the parameters defined in (6) leads to the minimization of the estimation error variance among all the virtual sensors with the same structure, i.e., $\mathcal{V}(\theta^*_{\mathcal{V}}) = \arg\min_{\mathcal{V}(\theta_{\mathcal{V}})} E[(z(t) - \hat{z}_{\mathcal{V}}(t))^2]$.

ii) If $\mathcal{K}(\theta^*_{\mathcal{M}}) \in \mathcal{V}(\theta_{\mathcal{V}})$ (i.e., it is possible to express the two-step observer in regression form as a particular realization of the virtual sensor), one obtains that $E[(z(t) - \hat{z}_{\mathcal{K}}(t))^2] \geq E[(z(t) - \hat{z}_{\mathcal{V}}(t))^2]$ which means that the virtual sensor has a performance that is better than or equal to that of the two-step observer.

iii) If there exists $\theta^o_{\mathcal{M}} \in D_{\theta_{\mathcal{M}}}$ such that $\mathcal{S} = \mathcal{M}(\theta^o_{\mathcal{M}})$ (i.e., there exists a set of parameters of the two-step observer that describes exactly the system dynamics) and $\mathcal{K}(\theta^o_{\mathcal{M}}) \in \mathcal{V}(\theta_{\mathcal{V}})$, then $\mathcal{V}(\theta^*_{\mathcal{V}})$ is a minimum variance filter.

For *ii*) and *iii*) to be applied, it is necessary that $\mathcal{K}(\theta^*_{\mathcal{M}})$ has fading memory. □

In conclusion, Theorem 1 proves that, in the favorable situation of no modeling error (and assuming $\mathcal{K}(\theta^*_{\mathcal{M}})$ is computable), the two-step procedure performs no better than the direct approach. Also, in the presence of modeling errors, the virtual sensor $\mathcal{V}(\theta^*_{\mathcal{V}})$, although may be suboptimal, is the minimum variance estimator among the selected approximating class of filters. Such a result cannot be ensured using the two-step procedure, where modeling errors can determine large performance deteriorations. Finally, note that minimum variance filters are, in general, very difficult to derive (or to implement) for nonlinear systems; as a consequence, approximations like the extended Kalman filter are often used with no guarantees even on the boundedness of the estimation error.

Remark 1: To obtain the convergence results of Theorem 1, two conditions that are not explicitly stated are needed. The first assumption is on the data set, the second one is on the criterion used to choose the parameter vector (these are labeled as conditions S3 and C1 in [12], respectively). As also highlighted in [5], these conditions are automatically fulfilled if system (1) is observable and if a quadratic criterion as in (6) is used. As observability and a quadratic criterion are required in Theorem 1, we conclude that the results shown in [12] still hold. The three additional requirements (7)–(9) correspond to condition M1 in [12] and are exclusive properties of the structure of the virtual sensor.

The rest of this section is devoted to show that the proposed PWAS virtual sensor $\mathcal{V}_\alpha(w)$ fulfills the aforementioned requirements and, hence, the convergence properties of Theorem 1.

Lemma 1: For the PWAS virtual sensor $\mathcal{V}_\alpha(w)$ defined in (2), there exist two scalars $C > 0$ and λ, $0 < \lambda < 1$ such that the following conditions hold.

1) The estimate is limited at origin, i.e.,

$$\left| f_\alpha\left(\tilde{U}_0(t), \tilde{Y}_0(t); w\right)\right| \leq C \qquad (10)$$

for $\tilde{U}_0(t) \equiv 0 \in \mathbb{R}^{M_u n_u}$ and $\tilde{Y}_0(t) \equiv 0 \in \mathbb{R}^{M_y n_y}$.

2) The virtual sensor (2) has exponential fading memory

$$\begin{aligned}\left| f_\alpha\left(\tilde{U}_1(t), \tilde{Y}_1(t); w\right) - f_\alpha\left(\tilde{U}_2(t), \tilde{Y}_2(t); w\right)\right| \\ \leq C \sum_{s=0}^{t} \lambda^{t-s} \left[\|\tilde{u}_1(s) - \tilde{u}_2(s)\|_1 + \|\tilde{y}_1(s) - \tilde{y}_2(s)\|_1\right] \qquad (11)\end{aligned}$$

for any $(\tilde{U}_1(t), \tilde{Y}_1(t))$ and $(\tilde{U}_2(t), \tilde{Y}_2(t))$.

3) Function f_α is differentiable with respect to w for all $w \in D_w$, and the following exponential fading property is satisfied:

$$\begin{aligned}\left\| \nabla_w f_\alpha\left(\tilde{U}_1(t), \tilde{Y}_1(t); w\right) - \nabla_w f_\alpha\left(\tilde{U}_2(t), \tilde{Y}_2(t); w\right)\right\|_1 \\ \leq C \sum_{s=0}^{t} \lambda^{t-s} \left[\|\tilde{u}_1(s) - \tilde{u}_2(s)\|_1 + \|\tilde{y}_1(s) - \tilde{y}_2(s)\|_1\right] \qquad (12)\end{aligned}$$

for any $(\tilde{U}_1(t), \tilde{Y}_1(t))$ and $(\tilde{U}_2(t), \tilde{Y}_2(t))$. □

Proof: Let $n_\xi = M_u n_u + M_y n_y$, and define

$$\Xi(t) \triangleq \left[\tilde{U}'(t) \quad \tilde{Y}'(t)\right]' \in \mathbb{R}^{n_\xi}.$$

1) Let $\Xi_0(t) \equiv [\tilde{U}'_0(t) \quad \tilde{Y}'_0(t)]' = 0 \in \mathbb{R}^{n_\xi}$. Then, by recalling that $0 \leq \alpha_k(\cdot) \leq 1$, for any k, with a slight abuse of notation, we get $|f_\alpha(\Xi_0(t); w)| = |\sum_{k=1}^{N} w_k \alpha_k(\Xi_0(t))| \leq \sum_{k=1}^{N} |w_k| \triangleq C_1 > 0$.

2) Take two vectors $\Xi_1(t)$ and $\Xi_2(t)$, and consider the left-hand side of (11)

$$\begin{aligned}&|f_\alpha\left(\Xi_1(t); w\right) - f_\alpha\left(\Xi_2(t); w\right)\| \\ &\quad = \left|\sum_{k=1}^{N} w_k\left(\alpha_k\left(\Xi_1(t)\right) - \alpha_k\left(\Xi_2(t)\right)\right)\right| \\ &\quad \leq \sum_{k=1}^{N} \left|w_k\left[\alpha_k\left(\Xi_1(t)\right) - \alpha_k\left(\Xi_2(t)\right)\right]\right| \\ &\quad = \sum_{k=1}^{N} |w_k| \left|\alpha_k\left(\Xi_1(t)\right) - \alpha_k\left(\Xi_2(t)\right)\right| \\ &\quad \leq C_1 \sum_{k=1}^{N} \left|\alpha_k\left(\Xi_1(t)\right) - \alpha_k\left(\Xi_2(t)\right)\right|.\end{aligned}$$

Due to the structure of the chosen α-basis, each basis function α_k is Lipschitz continuous. Hence, there exists a coefficient $\beta > 0$ such that

$$|\alpha_k(\Xi_1(t)) - \alpha_k(\Xi_2(t))| \leq \beta \|\Xi_1(t) - \Xi_2(t)\|_1 . \tag{13}$$

Then

$$C_1 \sum_{k=1}^{N} |\alpha_k(\Xi_1(t)) - \alpha_k(\Xi_2(t))| \leq C_1 \beta N \|\Xi_1(t) - \Xi_2(t)\|_1 .$$

Consider now the right-hand side of (11). Let λ be any scalar such that $0 < \lambda < 1$, and let $C_2 > 0$ be a constant to be determined. Then, recalling that $M = \max(M_u, M_y)$, we have

$$\begin{aligned}
&C_2 \sum_{s=0}^{t} \lambda^{t-s} \left[\|\tilde{u}_1(s) - \tilde{u}_2(s)\|_1 + \|\tilde{y}_1(s) - \tilde{y}_2(s)\|_1\right] \\
&\geq C_2 \left(\sum_{s=t-M_u+1}^{t} \lambda^{t-s} \|\tilde{u}_1(s) - \tilde{u}_2(s)\|_1 \right. \\
&\qquad \left. + \sum_{s=t-M_y+1}^{t} \lambda^{t-s} \|\tilde{y}_1(s) - \tilde{y}_2(s)\|_1 \right) \\
&\geq C_2 \lambda^{M-1} \left(\sum_{s=t-M_u+1}^{t} \|\tilde{u}_1(s) - \tilde{u}_2(s)\|_1 \right. \\
&\qquad \left. + \sum_{s=t-M_y+1}^{t} \|\tilde{y}_1(s) - \tilde{y}_2(s)\|_1 \right) \\
&= C_2 \lambda^{M-1} \|\Xi_1(t) - \Xi_2(t)\|_1
\end{aligned}$$

where $\sum_{s=t-M_u+1}^{t} \lambda^{t-s} \|\tilde{u}_1(s) - \tilde{u}_2(s)\|_1$ is zero for $M_u = 0$. By defining $C_2 = C_1 \beta N / \lambda^{M-1}$, we get

$$C_1 \beta N \|\Xi_1(t) - \Xi_2(t)\|_1 \leq C_2 \lambda^{M-1} \|\Xi_1(t) - \Xi_2(t)\|_1$$

which proves that (11) is satisfied.

3) Function f_α is clearly differentiable with respect to w, as

$$\begin{aligned}
\nabla_w f_\alpha(\Xi(t); w) &= \nabla_w \sum_{k=1}^{N} w_k \alpha_k(\Xi(t)) \\
&= [\alpha_1(\Xi(t)) \quad \ldots \quad \alpha_N(\Xi(t))] .
\end{aligned}$$

Consider the left-hand side of (12). From (13), we get

$$\begin{aligned}
&\|\nabla_w f_\alpha(\Xi_1(t); w) - \nabla_w f_\alpha(\Xi_2(t); w)\|_1 \\
&= \sum_{k=1}^{N} |\alpha_k(\Xi_1(t)) - \alpha_k(\Xi_2(t))| \leq \beta N \|\Xi_1(t) - \Xi_2(t)\|_1 .
\end{aligned}$$

Since the right-hand side of (12) coincides with that of (11), by setting $C_3 = \beta N / \lambda^{M-1}$, we get

$$\beta N \|\Xi_1(t) - \Xi_2(t)\|_1 \leq C_3 \lambda^{M-1} \|\Xi_1(t) - \Xi_2(t)\|_1 .$$

In conclusion, for any choice of λ, $0 < \lambda < 1$, by taking $C = \max(C_1, C_2, C_3)$, the theorem is proven. ■

Referring to Remark 1, the following result follows as a corollary of the main result of [12].

Corollary 1: Consider system (1), and assume that (g, h_y) is observable. Consider a minimum variance filter $\mathcal{K}(\theta^*_{\mathcal{M}})$ and the proposed PWAS virtual sensor $\mathcal{V}_\alpha(w^*)$, whose parameters are obtained by solving problem (3). Then, denoting the statistical expectation by $E[\cdot]$, the following results hold with probability 1 as $T \to \infty$.

1) $\mathcal{V}_\alpha(w^*) = \arg\min_{\mathcal{V}_\alpha(w)} E[(z(t) - \hat{z}_{\mathcal{V}}(t))^2]$.
2) If $\mathcal{K}(\theta^*_{\mathcal{M}}) \in \mathcal{V}_\alpha(w)$, then $E[(z(t) - \hat{z}_{\mathcal{K}}(t))^2] \geq E[(z(t) - \hat{z}_{\mathcal{V}}(t))^2]$.
3) If there exists $\theta^o_{\mathcal{M}} \in D_{\theta_{\mathcal{M}}}$ s.t. $\mathcal{S} = \mathcal{M}(\theta^o_{\mathcal{M}})$ and $\mathcal{K}(\theta^o_{\mathcal{M}}) \in \mathcal{V}_\alpha(w)$, then $\mathcal{V}_\alpha(w^*)$ is a minimum variance filter. □

Proof: We need to show that conditions S3, C1, and M1 in [12] hold. Condition S3 is an assumption on the data set and is automatically fulfilled by having assumed that (g, h_y) is observable. Condition C1 is on the choice of the parameter vector w and is fulfilled by having adopted the quadratic criterion in (3) for the PWAS virtual sensor. Finally, since $\mathcal{V}_\alpha(w^*)$ is a PWAS virtual sensor, Lemma 1 guarantees that condition M1 is satisfied. ■

V. Implementation Issues

A. RLS Problem

A number of numerical issues have to be faced to design a practical implementation of the proposed PWAS virtual sensor.

By directly solving problem (3), it is possible to obtain an ill-conditioned problem, i.e., the solution is sensitive to small changes in the data. A practical work-around is to rely on the so-called Tikhonov regularization, which consists of calculating the weights w_k by solving the *regularized least squares* (RLS) problem

$$\min_w \left\{ \sigma w' \Gamma w + \sum_{t=M}^{T} \left[\tilde{z}(t) - f_\alpha\left(\tilde{U}(t), \tilde{Y}(t); w\right)\right]^2 \right\} \tag{14}$$

where σ is the Tikhonov regularization parameter. The second term in the sum can be reformulated as a quadratic function of w and takes into account the square error between the output of the PWAS model and the actual data $\tilde{z}(t)$. The first term performs a Tikhonov regularization that depends on the structure of Γ [28]. 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_α (or higher order derivatives) in constructing Γ; see [28] for details.

The RLS problem (14) can be rewritten as

$$\min_w \left\{ \sigma w' \Gamma w + \|\tilde{z} - H w\|_2^2 \right\} \tag{15}$$

where

$$\tilde{z} = \begin{bmatrix} \tilde{z}_M \\ \vdots \\ \tilde{z}_T \end{bmatrix} H = \begin{bmatrix} \alpha'\left(\tilde{U}(M), \tilde{Y}(M)\right) \\ \vdots \\ \alpha'\left(\tilde{U}(T), \tilde{Y}(T)\right) \end{bmatrix} .$$

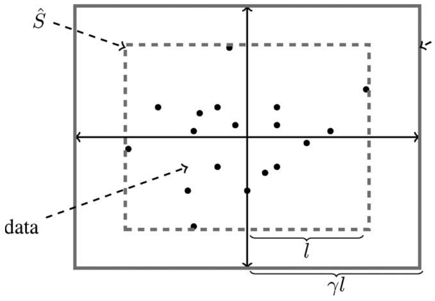

Fig. 1. Definition of the domain in a 2-D case.

Matrix H has $T - M + 1$ rows, each one corresponding to a data sample, and N columns, each one corresponding to a basis function, while $\tilde{z} \in \mathbb{R}^{T-M+1}$ is the data vector and $\Gamma \in \mathbb{R}^{N \times N}$ is the Tikhonov regularization matrix. Assume that σ has been fixed. Then, problem (15) has the unique solution

$$w_\sigma = (H'H + \sigma\Gamma)^{-1} H'\tilde{z} = H^* \tilde{z} \qquad (16)$$

where H^* is the (regularized) pseudoinverse matrix of H. As $\Gamma = \Gamma'$, $\det\Gamma \neq 0$, matrix $(H'H + \sigma\Gamma)$ is always invertible. A singular value decomposition of matrix H provides some information for selecting a good value for σ [28]. An alternative way that is suitable also for large matrices consists of cross-validation, a method to ensure model prediction ability [28].

B. Domain Definition and Basis Function Selection

The choice of the domain of definition $S \subset \mathbb{R}^{n_\xi}$ of the PWAS function also requires some care. Since no *a priori* information is available, the boundary of S must be also estimated before solving (14). As the circuits proposed in [13] are able to evaluate PWAS functions defined over hyperrectangular domains S, we assume such a form, and we identify the upper and lower limits for each dimension.

Let $\hat{S}$ be the hyperrectangle that exactly contains all data. The choice $S = \hat{S}$ is not safe enough, since some trajectory could fall outside $\hat{S}$ for $t > T$ (i.e., when the virtual sensor is operating) and PWAS circuits produce an incorrect output when the input lies outside the domain boundaries. Then, we calculate S as an expansion of $\hat{S}$ with respect to its center by a constant factor $\gamma > 1$. Fig. 1 shows an example for data scattered in a 2-D space. Such a heuristic criterion for choosing S must be validated *a posteriori* by running simulations or experimental tests. This procedure can be skipped when the value of γ is imposed by physical constraints on the system.

Once the domain S is chosen, many basis functions can be adopted among those proposed in the literature [22]–[24]; in particular, we can use the α-basis functions described previously. The resulting PWAS function is not affected by this choice because each basis spans the same space S, although the numerical complexity of problem (14) might depend on the chosen basis. In particular, as the α-basis has a local nature, each basis function is nonzero only on a limited region of the domain S, which makes the matrices in (16) amenable for sparse matrix computations.

TABLE I
PARAMETERS INFLUENCING A PWA VIRTUAL SENSOR

Parameter	Definition	Effects on
M_u	Number of past inputs to $\mathcal{S}$	N, model precision
M_y	Number of past outputs from $\mathcal{S}$	N, model precision
N	Number of basis functions	RLS complexity, circuit memory size
Γ	Tikhonov regularization matrix	numerical stability
σ	Tikhonov regularization weight	numerical stability
T	Number of data	RLS algorithm speed
γ	Domain expansion factor	model precision

C. Parameters and Settings

The performance of the virtual sensor depends on a set of parameters that must be fixed before solving problem (14). In this section, we provide a short description of their effects on the structure of the virtual sensor and highlight the tradeoffs between opposite requirements when choosing such parameters. The description is summarized in Table I.

1) The parameters M_u and M_y establish the dimension of the domain of definition of the PWAS function, given by $n_\xi = n_u M_u + n_y M_y$. On the one hand, this quantity must be as small as possible because the number of coefficients w_k grows exponentially with the dimension of the PWAS function; on the other hand, usually, a more complicated model has the ability to better capture the dynamics hidden in the data, thus reducing the discrepancy between $\hat{z}(t)$ and $z(t)$. Finally, we remark that M_u and M_y can be estimated through statistical techniques such as the mutual information [29], [30].
2) The number of vertices N depends on the chosen partition S and is the number of coefficients w_k, which, in turn, influences the complexity of the optimization problem (14) and the dimension of the memory required by the circuit implementation.
3) About the Tikhonov regularization parameter σ, there is a vast literature dealing with the problem of its optimal choice. In Section VI, we employ a cross-validation method. The choice of σ depends on the order of the Tikhonov regularization, i.e., on the structure of Γ. The reader is referred to [28] for further details.
4) The observation time interval T plays an important role in the identification of $\mathcal{V}_\alpha$, but also, the distribution of the data $y(t)$ and $z(t)$ must be considered. Indeed, all the main dynamics of $\mathcal{S}$ must lie inside the window $[0, T]$, including transient responses. The model estimated by solving (14) is reliable only in a neighborhood of the data samples. Therefore, the input $u(t)$ and the outputs $y(t)$ and $z(t)$ should span uniformly the input and output spaces for $t = 0, \ldots, T$ [31]. For this condition to hold true, the input $u(t)$ must be driven to stimulate all the possible behaviors of $\mathcal{S}$. If this control action on $u(t)$ is not feasible (for instance, because $\mathcal{S}$ is autonomous or because of input constraints), the system $\mathcal{S}$ can be observed starting from different initial conditions x_0, recording both transient and steady-state solutions [31]. In this case, the data set is a collection of different trajectories starting from different initial conditions. Finally, a large value of

T can help in capturing the dynamics of $\mathcal{S}$ but has the drawback that matrix H in (16) grows linearly with T, so the memory requirements and the time needed to solve problem (14) increase.

In the examples described in the next section, the choice of some parameters is done as a tradeoff between the accuracy of the estimation and the complexity and speed of the resulting circuit, on the basis of trials with different parameter sets.

VI. Simulations and Experiments

We test the performance of the proposed approach on a simulation example and on an experimental one. In both cases, the resulting PWAS functions defining the virtual sensor were implemented on a FPGA Xilinx Spartan 3 xc3s200, coding all circuit inputs with a 12-b precision and adopting architecture B in [13]. Moreover, the relatively low slice occupation for the considered cases allows one to embed more complex systems, such as a controller or a soft microprocessor, on the same FPGA together with the virtual sensor.

Simulations have been carried out using the root mean square estimation error (RMSEE) calculated over a test set as a measure of the accuracy of the estimation

$$\text{RMSEE} = \sqrt{\frac{1}{T_s}\sum_{t=1}^{T_s}\left(\hat{z}(t) - z(t)\right)^2} \tag{17}$$

where T_s is the number of samples in the test set. Standard international units are used for parameters unless specified differently.

A. Lorenz System

The discrete-time version of the Lorenz system (consisting in simplified equations of convection rolls arising in atmospheric dynamics) allows a comparison of the proposed PWAS virtual sensor with the results obtained in [5], where a virtual sensor is implemented using a one-hidden-layer neural network (HLNN). Consider the discrete-time Lorenz system

$$\begin{cases} x_1(t+1) = (1-\tau s)x_1(t) + \tau s x_2(t) \\ x_2(t+1) = (1-\tau)x_2(t) - \tau x_1(t)x_3(t) + \tau\rho x_1(t) \\ x_3(t+1) = (1-\tau\beta)x_3(t) + \tau x_1(t)x_2(t) \\ \tilde{y}(t) = x_1(t) + \eta_y(t) \\ \tilde{z}(t) = x_2(t)x_3(t) + \eta_z(t) \end{cases} \tag{18}$$

where $\tau = 0.01$ is the sampling time, $s = 10$, $\beta = 8/3$ and $\rho = 28$ are fixed parameters, and $\eta_y(t)$ and $\eta_z(t)$ are Gaussian processes with zero mean and standard deviations of 0.02 and 20, respectively. With this set of parameters, system (18) exhibits a chaotic behavior.

A PWAS virtual sensor has been derived from a set of $T = 6000$ samples of $\tilde{z}(t)$ and $\tilde{y}(t)$. We selected a uniform partition with seven subdivisions along each dimension and a zero-order Tikhonov regularization. The parameters used for the virtual sensors are reported in Table II, except for σ, which has been estimated by a cross-validation technique. The Lorenz system is autonomous ($n_u = 0$), so M_u can be ignored. With $M_y = 4$ and $n_y = 1$, we work with functions defined over 4-D domains.

TABLE II
Parameters and Simulation Results for a PWAS Virtual Sensor Applied to the Lorenz System

Method	N	σ	M_y	γ	RMSEE
PWAS	4096	$5.57\,10^{-4}$	4	1.2	22.45
HLNN [5]	/	/	/	/	24

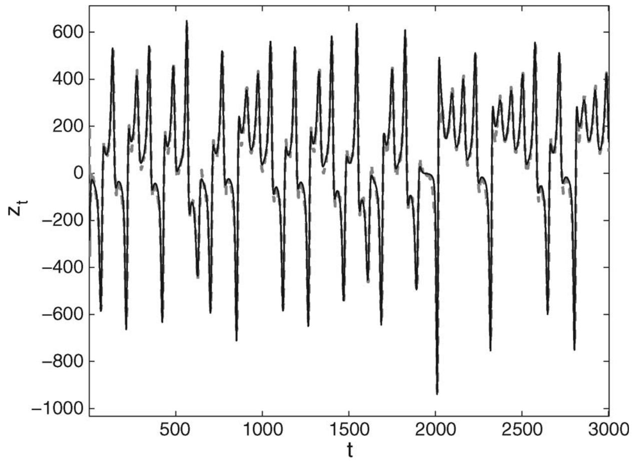

Fig. 2. PWA virtual sensor estimation example: Time evolution of (thin black line) the actual output $z(t)$ and (dashed gray line) the estimate $\hat{z}(t)$.

TABLE III
Effects of M_y on a PWAS Virtual Sensor

N	σ	$\mathbf{M_y}$	γ	RMSEE	L(ns)	SO
512	$5.18\,10^{-4}$	**3**	1.2	25.55	65	32%
4096	$5.57\,10^{-4}$	4	1.2	22.45	65	28%
32768	$6.41\,10^{-4}$	**5**	1.2	20.03	70	33%

Table II also shows the RMSEE, calculated over $T_s = 2000$ samples, for the PWA virtual sensor (simulated in Matlab with double precision) and for the HLNN used in [5]. The PWAS virtual sensor has performances comparable to those of the HLNN. The time evolutions of the actual and estimated outputs are shown in Fig. 2.

The latency, i.e., the time required to calculate the estimate $\hat{z}(t)$, is approximatively 65 ns, whereas the slice occupation, i.e., the percentage of occupied resources, is 28%.

The limited numerical precision used on the FPGA has little effects on the performance of the virtual sensor: The RMSEE value obtained using a 12-b precision is 27.05. Typically, by increasing the number of bits, the estimation error decreases, at the price of an increased latency. The reader is referred to [13] for some examples on the effects of the limited numerical precision on the implementation of PWAS functions.

The same simulation was tested under different parameter configurations. The parameters were changed one at a time with respect to the values listed in Table II. The results are shown in Tables III–V, where the data of Table II are repeated for ease of comparison. In each table, the modified parameter is marked in bold text. In particular, Table III shows the effects of varying M_y, showing that the number of basis functions changes exponentially with respect to this parameter. Table IV shows the effects of the variation of N, which was changed by setting the number of partitions along each dimension to seven (first row), three (middle row), or one (last row); as expected, the RMSEE

TABLE IV
EFFECTS OF N ON A PWAS VIRTUAL SENSOR

N	σ	M_y	γ	RMSEE	L(ns)	SO
4096	$5.57\,10^{-4}$	4	1.2	22.45	65	28%
256	$2.76\,10^{-4}$	4	1.2	29.16	70	20%
16	$5\,10^{-6}$	4	1.2	50.99	66	19%

TABLE V
EFFECTS OF γ ON A PWAS VIRTUAL SENSOR

N	σ	M_y	γ	RMSEE	L(ns)	SO
4096	$2.64\,10^{-4}$	4	**1.1**	22.36	65	28%
4096	$5.57\,10^{-4}$	4	1.2	22.45	65	27%
4096	$4.04\,10^{-4}$	4	**1.5**	24.67	65	27%
4096	$3.79\,10^{-4}$	4	**2**	30.13	65	28%

increases by decreasing N. Finally, Table V is related to the variation of γ, showing the corresponding increase of RMSEE.

The chosen parameter set (see Table II) is a good tradeoff between the accuracy of the estimation and the complexity and speed of the resulting circuit.

B. Laboratory Helicopter Experiment

We test the proposed estimation approach on the educational laboratory helicopter CE150 (Humusoft) of the telelaboratory [32]. The model is nonlinear and has two degrees of freedom (see [33] and the references therein for a detailed description). The two control variables are generated by PID controllers, and the two measured outputs are the elevation ψ and azimuth φ of the helicopter. The continuous-time dynamics of the elevation angle can be approximately described by the following torque balance

$$I\ddot{\psi} = \tau_1 - \tau_{\dot{\psi}} - \tau_{f_1} - \tau_m - \tau_G$$

where I is the moment of inertia of the helicopter around the horizontal axis, $\tau_1 = k_{\tau_1}\omega_1^2$ is the elevation driving torque given by the main propeller (ω_1 is the angular velocity of this latter, while k_{τ_1} is a constant coefficient), $\tau_{\dot{\psi}} = ml\dot{\varphi}^2 \sin\psi\cos\psi$ is the centrifugal torque (m is the helicopter mass, and l is the distance of the center of mass from the support), $\tau_{f_1} = c_\psi \text{sign}\dot{\psi} - b_\psi\dot{\psi}$ is the friction torque (b_ψ and c_ψ are constant coefficients), $\tau_m = mgl\sin\psi$ is the gravitational torque (g is the gravity acceleration), and $\tau_G = k_G\dot{\psi}\tau_1\cos\psi$ is the gyroscopic torque (k_G is a constant coefficient). The approximate continuous-time dynamics of the azimuth angle is

$$I\sin\psi\ddot{\varphi} = \tau_2 - \tau_{f_2} - \tau_r$$

where $\tau_2 = k_{\tau_2}\sin\psi\omega_2^2$ is the stabilizing motor driving torque given by the side propeller (ω_2 is the angular velocity of this latter, and k_{τ_2} is a constant coefficient), $\tau_{f_2} = c_\varphi\text{sign}\dot{\varphi} - b_\varphi\dot{\varphi}$ is the friction torque (b_φ and c_φ are constant coefficients), and τ_r is the main motor reaction torque. The full model of the experiment includes also the model of the dc motors actuating the propellers and the equations of the PID controllers [34].

Experiments were carried out according to the closed-loop configuration of Fig. 3. The reference trajectories ψ_d (elevation) and φ_d (azimuth) are the inputs to the PID controllers, the azimuth angle φ is the measured output, and the elevation angle ψ (although available to the PID controller) is assumed here to be unavailable and is estimated by the virtual sensor. The presence of unmeasured stochastic disturbances η_φ and η_ψ on the sensors is also assumed.

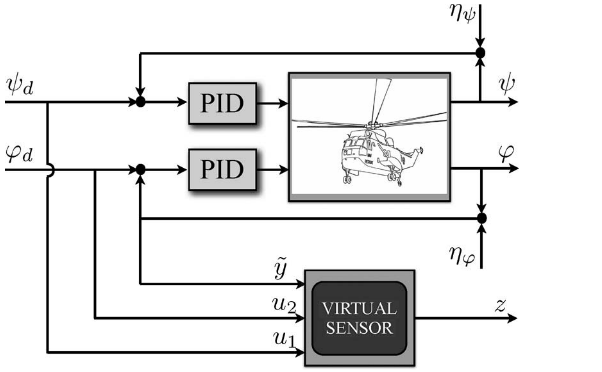

Fig. 3. Schematics of setup for estimating the elevation angle via the PWAS virtual sensor.

TABLE VI
PARAMETERS AND SIMULATION RESULTS FOR A PWAS VIRTUAL SENSOR APPLIED TO THE HELICOPTER MODEL

Method	N	σ	M_u	M_y	γ	RMSEE
PWAS	4096	50	2	2	1.1	2.29
HLNN	/	/	/	/	/	2.62

A two-step procedure to design an observer would require to first obtain a model in state-space form: For example, the model of the closed-loop system in [33] has six state variables.

To construct the virtual sensor, data are acquired with sample time $\tau = 0.01$ s. Let $u = [u_1\ \ u_2]' \triangleq [\psi_d\ \ \varphi_d]'$ be the input vector, $\tilde{y} = \varphi + \eta_\varphi$ be the measured output, and $z = \psi$ be the estimated output. Note that there are no disturbances affecting the signal u, so $\tilde{u} = u$.

A PWAS virtual sensor has been derived from a set of $T = 25\,000$ samples of $\tilde{z}(t)$, $\tilde{y}(t)$, and $\tilde{u}(t)$. We selected a partition with three subdivisions along each dimension and a first-order Tikhonov regularization. Table VI shows the values of the parameters, the resulting RMSEE (for the PWA virtual sensor simulated in Matlab with double precision), and the implementation results. Once again, the chosen parameter set is a good tradeoff between the accuracy of the estimation and the complexity and speed of the resulting circuit. Actual and estimated outputs are plotted in Fig. 4, together with the elevation reference input signal. For comparison purposes, different HLNNs, analogous to that proposed in [5], have been trained on the same data set. The lowest RMSEE has been obtained with an HLNN with six neurons in the hidden layer and is close to that obtained with the PWAS virtual sensor (see Table VI).

The resulting latency of the PWAS virtual sensor implemented on the selected FPGA is approximatively 74 ns, whereas the slice occupation is 29%. The RMSEE value obtained with the 12-b precision is 2.32.

VII. CONCLUSION

This paper has investigated the use of PWAS functions to construct virtual sensors. In addition to proving interesting theoretical properties of the filter, the main practical advantage of the PWAS structure is its implementation in low-cost digital

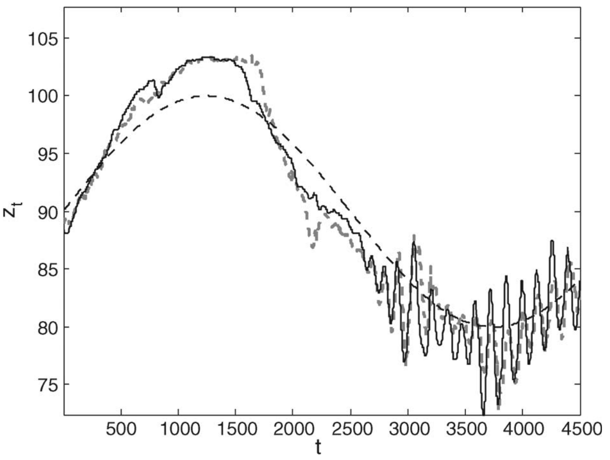

Fig. 4. Time evolution of (thin black line) the actual output elevation $z(t)$, (dashed gray line) the estimate $\hat{z}(t)$, and (dashed black line) the reference input signal.

circuits (FPGA) at a very fast rate, constituting a valid solution for state estimation in many practical applications.

Acknowledgment

The authors would like to thank Prof. M. Milanese and Prof. C. Novara from Politecnico di Torino for the useful discussions and S. Triboli, a Ph.D. student at the University of Trento, for providing data for the laboratory helicopter experiment.

References

[1] G. Welch and G. Bishop, "An introduction to the Kalman filter," Univ. North Carolina, Chapel Hill, NC, Tech. Rep. TR95-041, 1995.

[2] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[3] G. Evensen, "The ensemble Kalman filter: Theoretical formulation and practical implementation," *Ocean Dyn.*, vol. 53, no. 4, pp. 343–367, Nov. 2003.

[4] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[5] M. Milanese, C. Novara, K. Hsu, and K. Poolla, "Filter design from data: Direct vs. two-step approach," in *Proc. Amer. Control Conf.*, Minneapolis, MN, 2006, pp. 1606–1611.

[6] J. Stephant, A. Charara, and D. Meizel, "Virtual sensor: Application to vehicle sideslip angle and transversal forces," *IEEE Trans. Ind. Electron.*, vol. 51, no. 2, pp. 278–289, Apr. 2004.

[7] M. Milanese, C. Novara, K. Hsu, and K. Poolla, "The filter design from data (FD2) problem: Nonlinear set membership approach," *Automatica*, vol. 45, no. 10, pp. 2350–2357, Oct. 2009.

[8] M. Canale, L. Fagiano, F. Ruiz, and M. Signorile, "A study on the use of virtual sensors in vehicle control," in *Proc. Conf. Decision Control*, Cancun, Mexico, 2008, pp. 4402–4407.

[9] P. Borodani, "Virtual sensors: An original approach for dynamic variables estimation in automotive control systems," in *Proc. Int. Symp. Adv. Vehicle Control*, Kobe, Japan, 2008.

[10] M. Canale, L. Fagiano, and C. Novara, "Vehicle side-slip angle estimation using a direct MH estimator," in *Proc. Int. Conf. Control Appl.*, Yokohama, Japan, 2010, pp. 167–172.

[11] C. Novara, F. Ruiz, and M. Milanese, "Direct identification of optimal SM-LPV filters and application to vehicle yaw rate estimation," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 1, pp. 5–17, Jan. 2011.

[12] L. Ljung, "Convergence analysis of parametric identification methods," *IEEE Trans. Autom. Control*, vol. AC-23, no. 5, pp. 770–783, Oct. 1978.

[13] M. Storace and T. Poggi, "Digital architectures for the circuit implementation of PWL multi-variate functions: Two FPGA implementations," *Int. J. Circuit Theory Appl.*, vol. 39, pp. 1–15, 2011.

[14] A. Rosado-Munoz, M. Bataller-Mompean, E. Soria-Olivas, C. Scarante, and J. Guerrero-Martinez, "FPGA implementation of an adaptive filter robust to impulsive noise: Two approaches," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 860–870, Mar. 2011.

[15] J. Weber, E. Oruklu, and J. Saniie, "FPGA-based configurable frequency-diverse ultrasonic target-detection system," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 871–879, Mar. 2011.

[16] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.

[17] P. Tøndel, T. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, May 2003.

[18] A. Germani and C. Manes, "State observers for nonlinear systems with slowly varying inputs," in *Proc. Conf. Decision Control*, San Diego, CA, Dec. 1997, pp. 5054–5059.

[19] J. C. Vivalda, "On the genericity of the observability of uncontrolled discrete nonlinear systems," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1509–1522, 2003.

[20] H. Kuhn, "Some combinatorial lemmas in topology," *IBM J. Res. Develop.*, vol. 4, no. 5, pp. 518–524, Nov. 1960.

[21] P. Julián, A. Desages, and O. Agamennoni, "High-level canonical piecewise linear representation using a simplicial partition," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 46, no. 4, pp. 463–480, Apr. 1999.

[22] P. Julián, A. Desages, and B. D'Amico, "Orthonormal high level canonical PWL functions with applications to model reduction," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 47, no. 5, pp. 702–712, May 2000.

[23] M. Storace, L. Repetto, and M. Parodi, "A method for the approximate synthesis of cellular non-linear networks—Part 1: Circuit definition," *Int. J. Circuit Theory Appl.*, vol. 31, no. 3, pp. 277–297, May/Jun. 2003.

[24] L. Repetto, M. Storace, and M. Parodi, "A method for the approximate synthesis of cellular non-linear networks—Part 2: Circuit reduction," *Int. J. Circuit Theory Appl.*, vol. 31, no. 3, pp. 299–313, May/Jun. 2003.

[25] R. Rovatti, M. Borgatti, and R. Guerrieri, "A geometric approach to maximum-speed n-dimensional continuous linear interpolation in rectangular grids," *IEEE Trans. Comput.*, vol. 47, no. 8, pp. 894–899, Aug. 1998.

[26] R. Rovatti, C. Fantuzzi, and S. Simani, "High-speed DSP-based implementation of piecewise-affine and piecewise-quadratic fuzzy systems," *Signal Process.*, vol. 80, no. 6, pp. 951–963, Jun. 2000.

[27] P. Echevarria, M. Martínez, J. Echanobe, I. del Campo, and J. Tarela, "Digital hardware implementation of high dimensional fuzzy systems," in *Applications of Fuzzy Sets Theory*. Berlin, Germany: Springer-Verlag, 2007, ser. Lecture Notes in Computer Science, pp. 245–252.

[28] R. Aster, B. Borchers, and C. Thurber, *Parameters Estimation and Inverse Problems*. New York: Academic, 2004.

[29] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[30] M. Small, *Applied Nonlinear Time Series Analysis: Applications in Physics, Physiology and Finance*. Singapore: World Scientific, 2005.

[31] O. De Feo and M. Storace, "Piecewise-linear identification of nonlinear dynamical systems in view of their circuit implementations," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 7, pp. 1542–1554, Jul. 2007.

[32] M. Casini, D. Prattichizzo, and A. Vicino, "Operating remote laboratories through a bootable device," *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3134–3140, Dec. 2007.

[33] H. F. Ho, Y. K. Wong, and A. Rad, "Direct adaptive fuzzy control for a nonlinear helicopter system," *Aircr. Eng. Aerosp. Technol.*, vol. 80, no. 2, pp. 124–128, 2008.

[34] *CE150 Helicopter Model: User's Manual*, Humusoft, Prague, Czech Republic, 2002.

Tomaso Poggi was born in Varazze, Italy, in 1982. He received the M.Sc. degree in electronic engineering and the Ph.D. degree in mathematical and simulation engineering from the University of Genoa, Genova, Italy, in 2006 and 2010, respectively.

He has been a Research Assistant with the Department of Biophysical and Electronic Engineering, University of Genoa. His main research interests include the piecewise-affine approximation/identification of nonlinear functions and systems and the circuit implementation of nonlinear circuits. He is also interested in the circuit implementation of explicit model predictive control and in the design of control systems for radio-frequency applications, particularly particle accelerators.

Matteo Rubagotti (S'07–M'11) was born in Voghera (PV), Italy, in 1982. He received the B.Sc. and M.Sc. degrees (both with highest honors) in computer engineering and the Ph.D. degree in electronic, computer science, and electrical engineering from the University of Pavia, Pavia, Italy, in 2004, 2006, and 2010, respectively.

As a Ph.D. student, he held visiting positions at the Ohio State University Center for Automotive Research, Columbus, from October 2008 to April 2009 and at the Automatic Control Laboratory, Eidgenössische Technische Hochschule Zurich, Zurich, Switzerland, in July and August 2009. Since January 2010, he has been a Postdoctoral Fellow with the Department of Mechanical and Structural Engineering, University of Trento, Trento, Italy. His research interests are in the area of nonlinear control systems (hybrid and piecewise-affine dynamical systems, sliding mode control, and model predictive control), with applications in mobile robotics and aircraft avionics.

Alberto Bemporad (S'95–M'99–SM'06–F'10) received the M.S. degree in electrical engineering and the Ph.D. degree in control engineering from the University of Florence, Florence, Italy, in 1993 and 1997, respectively.

He spent the academic year 1996/97 at the Center for Robotics and Automation, Department of Systems Science and Mathematics, Washington University, St. Louis, as a Visiting Researcher. In 1997–1999, he held a postdoctoral position at the Automatic Control Laboratory, Eidgenössische Technische Hochschule Zurich, Zurich, Switzerland, where he collaborated as a Senior Researcher in 2000–2002. In 1999–2009, he was with the Department of Information Engineering of the University of Siena, Siena, Italy, where he became an Associate Professor in 2005. In 2010–2011, he was with the Department of Mechanical and Structural Engineering, University of Trento, Trento, Italy. Since 2011, he has been a Full Professor with the IMT Institute for Advanced Studies, Lucca, Italy. He has published more than 220 papers in the areas of model predictive control, hybrid systems, automotive control, multiparametric optimization, computational geometry, robotics, and finance. He is the author or coauthor of various MATLAB toolboxes for model predictive control design, including the Model Predictive Control Toolbox (The Mathworks, Inc.).

Dr. Bemporad was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL during 2001–2004 and the Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society in 2002–2010.

Marco Storace (M'01) was born in Genoa, Italy, in 1969. He received the "Laurea" (M.Sc.) five-year degree (*summa cum laude*) in electronic engineering and the Ph.D. degree in electrical engineering from the University of Genoa, Genova, Italy, in 1994 and 1998, respectively.

Since 2004, he has been an Associate Professor with the Department of Biophysical and Electronic Engineering, University of Genoa. He was a visitor to École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 1998 and 2002. His main research interests are in the area of nonlinear circuit theory and applications, with emphasis on circuit models of nonlinear systems (e.g., hysteresis and biological neurons), methods for the piecewise-linear approximation of nonlinear systems and for the consequent circuit synthesis, and nonlinear dynamics. He is the author or coauthor of more than 90 scientific papers, more than half of which have been published in international journals.

Dr. Storace served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II (2008–2009) and is a member of the IEEE Technical Committee on Nonlinear Circuits and Systems. He is the Coordinator of the MOBY-DIC European Project (FP7-IST-248858).