

# Active State Estimation for Nonlinear Systems: A Neural Approximation Approach

Luca Scardovi, *Member, IEEE*, Marco Baglietto, *Associate Member, IEEE*, and  
Thomas Parisini, *Senior Member, IEEE*

**Abstract**—In this paper, we consider the problem of actively providing an estimate of the state of a stochastic dynamic system over a (possibly long) finite time horizon. The active estimation problem (AEP) is formulated as a stochastic optimal control one, in which the minimization of a suitable uncertainty measure is carried out. Toward this end, the use of the Renyi entropy as an information measure is proposed and motivated. A neural control scheme, based on the application of the extended Ritz method (ERIM) and on the use of a Gaussian sum filter (GSF), is then presented. Simulation results show the effectiveness of the proposed approach.

**Index Terms**—Active estimation, entropy, neural networks (NNs).

## I. INTRODUCTION

**I**N this paper, we address the problem of actively estimating the state of a stochastic dynamic system. By active estimation we mean the problem of finding a feedback control law that aims at “maximizing the amount of information” on the state of the system. In particular, we formalize the problem within the framework of stochastic optimal control where a suitable uncertainty measure is added to the process cost to be minimized.

We do not assume that the classical linear Gaussian (LG) hypotheses are verified. Indeed, if these hypotheses are satisfied, the well-known separation principle states that the choice of the control law does not affect the estimation process, that is, any control law is “equally informative” (see, for instance, [1] and the references therein). Moreover, in the case of linear systems, it is well known that, under the hypothesis of Gaussian noises, the *a posteriori* probability density function (pdf) of the state of the system is Gaussian as well. The mean value and the covariance matrix can be propagated by the usual Kalman filter equations and the control action does not influence the propagation of the covariance matrix.

Manuscript received December 1, 2005; revised September 20, 2006; accepted February 5, 2007. This work was supported in part by the Italian Ministry for University and Research, the Regione Friuli Venezia Giulia under Grant L. 11/03, and the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office.

L. Scardovi is with the Department of Electrical Engineering and Computer Science, University of Liège, B-4000 Liège, Belgium (e-mail: lscardovi@ulg.ac.be).

M. Baglietto is with the Department of Communications, Computer and System Sciences, DIST-University of Genoa, 16145 Genova, Italy (e-mail: mbaglietto@dist.unige.it).

T. Parisini is with the Department of Electrical, Electronic, and Computer Engineering University of Trieste, 34127 Trieste, Italy (e-mail: parisini@units.it).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2007.899251

In statistics, a similar problem is the optimal experiment design (OED), where the objective is to design an experiment in order to infer about an unknown parameterized system [2]. Also, in machine learning, a similar problem arises when one can choose the input patterns for the training process (active learning) [3], [4]. In robotics, the problem of environment exploration can be formulated as a particular case of the active identification problem, and it has been studied from a heuristic point of view [5]–[7]. Some researchers have used information theoretical concepts to study control problems (see [1] and [8]) and in [9] and [10] the problem of actively identifying a set of unknown parameters in a linear regression setting has been addressed.

In this paper, we formulate the problem in an information theoretical setting by using the Renyi entropy [11] as a measure of information about the state of the system. This choice is motivated by the possibility of deriving a closed-form expression for the amount of information, thus avoiding resorting to computationally demanding nonlinear programming techniques.

In general, solving an  $N$ -stage finite-horizon (FH) stochastic optimal control problem requires the knowledge of the conditional pdf  $p(x_k|I_k)$ ,  $k = 0, 1, \dots, N - 1$ , where  $x_k$  is the state vector of the controlled system and  $I_k$  is the information vector consisting of all the measures taken by the controller up to stage  $k$  and of all the control actions applied to the system up to stage  $k - 1$ . Dynamic programming could be an effective tool to be applied, at least in principle. This technique, however, entails the recursive computation of the state conditional pdf. Unfortunately, such a computation can be accomplished analytically only in very few cases, typically, under the classical LQG hypotheses, i.e., when the system is linear, the cost is quadratic, and the random variables are Gaussian.

The conditional probability function is explicitly needed to calculate the measure of uncertainty (which is part of the cost function), but, in general, no analytical expression for it is available. This leads us to look for a suitable approximation. Toward this end, a Gaussian sum filtering (GSF) approach is adopted [12].

The approximating technique considered in this paper to solve the optimal control problem consists in assigning a given structure to the control laws, where a fixed number of parameters have to be determined in order to minimize the cost function. Multilayer feedforward neural networks (NNs) have been chosen for their good approximating properties. Actually, this family of NNs is characterized by the ability of approximating nonlinear functions (in our case, the optimal control functions) by using a number of parameters that may be surprisingly smaller than the one required by traditional

expansions, like the polynomial and trigonometric ones (this applies to a class of functions to be approximated that are characterized by suitable smoothness assumptions). Such a property, proved by Barron [13], should explain to some extent the successful practical results achieved by feedforward NNs in solving many application problems.

Constraining the control functions to take on fixed structures enables us to reduce the problem of finding the optimal control laws (which is a functional optimization problem) to a nonlinear programming one. Such a technique has been used successfully to solve non-LQG deterministic and stochastic optimal control problems (in finite, infinite, and receding-horizon (RH) cases; see [14] and references cited therein). In the lines of [15], the technique proposed in this paper to optimize the free parameters relies on the so-called stochastic approximation methodology (see [16]). Here, the possibility of differentiating the expressions for the Renyi entropy and for the GSF plays a central role.

This paper is organized as follows. In Section II, the active estimation problem (AEP) is formulated. In Section III, the GSF technique as an approximate solution to the estimation problem is addressed. In Section IV, two possible measures of information are introduced and the choice of the Renyi entropy is justified. An approximate version of the AEP is then presented. In Section V, the neural approximation approach to the solution of such a problem is discussed, and an RH extension of the proposed methodology is considered to address the problem over a long (possibly unknown) time horizon. Finally, in Section VI, simulation results are reported showing the effectiveness of the proposed neural active estimation technique.

## II. PROBLEM FORMULATION

Let us consider a discrete-time stochastic dynamic system (in general, nonlinear)

$$x_{k+1} = f(x_k, u_k) + \xi_k, \quad k = 0, 1, \dots \quad (1a)$$

$$y_k = h(x_k) + \eta_k, \quad k = 0, 1, \dots \quad (1b)$$

where  $x_k \in \mathbb{R}^n$ ,  $y_k \in \mathbb{R}^q$ , and  $u_k \in \mathbb{R}^p$  are the state vector, the observation vector, and the control vector, respectively, and where  $\xi_k \in \mathbb{R}^n$  and  $\eta_k \in \mathbb{R}^q$  are two independent white noise processes. The initial state  $x_0$  is unknown; however, we assume that its pdf  $p(x_0)$  is known.<sup>1</sup>

Let us consider an FH stochastic optimal control problem characterized by the following additive cost function (in general, nonquadratic) over a finite number  $N$  of temporal stages:

$$G = \sum_{k=0}^{N-1} g_k(x_k, u_k) + g_N(x_N) \quad (2)$$

where  $g_k(x_k, u_k)$  is the transition cost incurred at any stage  $k = 0, 1, \dots, N-1$ , and  $g_N(x_N)$  denotes the final cost.

Clearly, the unavailability of perfect information on the system state makes the aforesaid stochastic optimal control problem much more difficult than the one in which perfect measures on  $x_t$  are available. In the latter case, the state vector

$x_t$  “summarizes the whole history of the system” up to stage  $t$  and contains all the information needed to make a decision  $u_t$ . As usual (see, for instance, [17]), by “history of the system” we mean all the measures taken up to the current stage  $t$  and all the control vectors applied up to stage  $t-1$ . Then, we define the information vector  $I_k$ ,  $k = 0, 1, \dots$ , as

$$I_0 \triangleq y_0$$

$$I_k \triangleq \text{col}(y_0, \dots, y_k, u_0, \dots, u_{k-1}), \quad k = 1, 2, \dots$$

It follows that the control functions take on the closed-loop form:

$$u_k = \gamma_k(I_k), \quad k = 0, 1, \dots \quad (3)$$

As is well known, the classical FH stochastic optimal control problem involves the minimization of the expected value of a cost function of the form (2). However, in this paper, we address a different problem. More specifically, in qualitative terms, our goal is to design a control policy for the system (1) so that, at the final time instant  $N$ , the “maximum amount of information” on the state vector  $x_N$  is achieved, while, at the same time, the process cost (2) is kept at the smallest possible value. More precisely, we introduce a measure of information about the final state  $x_N$  as a function of the information vector  $I_N$ . We denote such a quantity by  $U(I_N)$ . We are now able to state the following optimal control problem.

*Problem 2.1 (AEP):* Given a scalar  $\rho \in [0, 1]$ , find the optimal control functions  $u_0^o = \gamma_0^o(I_0), \dots, u_{N-1}^o = \gamma_{N-1}^o(I_{N-1})$  that minimize the expected value of the cost functional

$$J = \rho \left[ \sum_{k=0}^{N-1} g_k(x_k, \gamma_k(I_k)) + g_N(x_N) \right] + (1-\rho)U(I_N) \quad (4)$$

subject to (1a) and (1b).  $\square$

Problem 2.1 will be referred to as the AEP. The scalar  $\rho$  appearing in (4) expresses the tradeoff between the process cost between brackets and the term  $U(I_N)$  denoting an uncertainty measure penalizing the lack of knowledge of  $x_N$  at the final time instant  $k = N$ . In Section IV, the choice of the function  $U(I_N)$  will be addressed in greater detail.

Some remarks about Problem 2.1 are now in place. In particular, it is important to emphasize again that the control law solving Problem 2.1 aims at the following two objectives simultaneously: 1) to provide an estimate of the state of the system at the final time instant and 2) to minimize the process cost. In this connection, it is well known that, unlike the LQG framework, the aforementioned stochastic optimal control problem is, in general, a dual control problem [1], [8], [18], and an optimal feedback control law has two different, usually conflicting, attributes known as probing (or learning) and regulating. The probing aspect of a feedback control law is critical in the dual control problem. By acting on the tradeoff parameter  $\rho$ , it is possible to address a classical stochastic optimal control problem ( $\rho = 1$ ) and an information retrieval problem ( $\rho = 0$ ) as well. If the only objective of the controller is to acquire information (i.e.,  $\rho = 0$ ), the problem is strictly related to optimal experiment design [2].

<sup>1</sup>Here and throughout this paper, given a generic random variable  $v$ , we denote by  $p(v)$  its pdf.

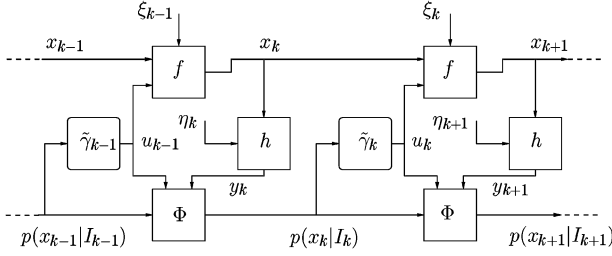


Fig. 1. Block diagram showing the sufficient statistics propagation according to the Bayes rule.

As is well known, unless particular assumptions are verified, solving Problem 2.1 analytically is an impossible task for several conceptual and practical reasons. One of these reasons turns out to be the increasing dimension of the information vector  $I_k$ ,  $k = 0, \dots, N - 1$  [see (3)].

Therefore, in the following, an equivalent problem will be formulated in which the arguments of the optimal control strategies will be characterized by a time invariant structure. In this respect, it is important to recall the concept of sufficient statistics  $p(x_k|I_k)$  [17].

The sufficient statistics  $p(x_k|I_k)$  summarizes all the available information about the system at time  $k = 0, 1, \dots$  and can be used (in principle) to determine the optimal control functions. Then, (3) can be replaced by

$$u_k = \tilde{\gamma}_k [p(x_k|I_k)], \quad k = 0, 1, \dots, N - 1 \quad (5)$$

where now we restrict the time index  $k$  to take on integer values less than  $N$  as we consider an FH problem. Comparing the structures of the control functions (3) and (5), we emphasize that in (5) the arguments of  $\tilde{\gamma}_k$  are functions (hence, infinite-dimensional entities), whereas in (3) the arguments of  $\gamma_k$  are finite-dimensional vectors. On the other hand, in (5), the structure of  $\tilde{\gamma}_k$  does not change over time, whereas the use of the functions (3) would involve argument vectors whose dimensions grow in time.

The conditional pdf  $p(x_k|I_k)$  can be generated recursively by the Bayes law (see Fig. 1, where a block diagram is depicted showing the evolution of the conditional pdf) and can be viewed as the state of a controlled discrete-time dynamic system [17]

$$p(x_{k+1}|I_{k+1}) = \Phi [p(x_k|I_k), u_k, y_{k+1}], \quad k = 0, 1, \dots, N - 1 \quad (6)$$

where  $\Phi$  represents the recursive Bayes updating law

$$\Phi [p(x_k|I_k), u_k, y_{k+1}] = \frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|I_k, u_k)}{p(y_{k+1}|I_k, u_k)} \quad (7)$$

with

$$p(x_{k+1}|I_k, u_k) = \int_{\mathbb{R}^n} p(x_{k+1}|x_k, u_k)p(x_k|I_k)dx_k$$

$$p(y_{k+1}|I_k, u_k) = \int_{\mathbb{R}^n} p(y_{k+1}|x_{k+1})p(x_{k+1}|I_k, u_k)dx_{k+1}.$$

We are now able to state the following optimal control problem, where we denote by  $H[p(x_N|I_N)]$  a measure of uncertainty corresponding to  $U(I_N)$  and expressed as a functional of the conditional pdf.

**Problem 2.2 (AEP):** Given a scalar  $\rho \in [0, 1]$ , find the optimal control functions  $u_0^o = \hat{\gamma}_0^o[p(x_0|I_0)], \dots, u_{N-1}^o = \hat{\gamma}_{N-1}^o[p(x_{N-1}|I_{N-1})]$  that minimize the expected value of the cost function

$$\tilde{J} = \rho \left[ \sum_{k=0}^{N-1} g_k(x_k, \tilde{\gamma}_k[p(x_k|I_k)]) + g_N(x_N) \right] + (1 - \rho)H[p(x_N|I_N)] \quad (8)$$

subject to (1a) and (1b) and to the updating rule (6).  $\square$

Thanks to the concept of ‘‘sufficient statistics,’’ Problem 2.1 is equivalent to Problem 2.2. Here, the same terms are present that make use of the conditional pdf  $p(x_k|I_k)$  instead of the information vector  $I_k$ . Consistently, the generic measure of uncertainty  $U(I_k)$  is replaced by the corresponding term  $H[p(x_k|I_k)]$ .

### III. PROPAGATION OF THE PDFS: GSF

The updating rule (6) plays a central role in Problem 2.2. Unfortunately, an explicit form for it is (in general) not available. An important exception is when the system is linear and the random variables are normally distributed; in this case, the conditional probability is also normal and (7) takes on the form of the well-known Kalman filter equations. Since we are interested in addressing the problem under general assumptions and the conditional pdf is needed to compute the measure of uncertainty, we resort to an approximate approach.

More specifically, we approximate the true conditional pdfs  $p(x_k|I_k)$  by means of suitable fixed-structure functions of the form  $\hat{p}(x_k, \hat{S}_k)$ , where the parameter vector  $\hat{S}_k$  is to be determined. The *a priori* pdf  $p(x_0)$  can be approximated by  $\hat{p}(x_0, \hat{S}_{-1})$  by determining the corresponding parameter vector  $\hat{S}_{-1}$ . Toward this end, one can resort to a broad class of methods among which the least squares criterion is the best known.

Clearly, a computational methodology for the propagation over time of the parameter vectors  $\hat{S}_k$  has to be designed with the following structure:

$$\hat{S}_0 = \hat{\Phi}_0(\hat{S}_{-1}, y_0)$$

$$\hat{S}_{k+1} = \hat{\Phi}_{k+1}(\hat{S}_k, u_k, y_{k+1}), \quad k = 0, 1, \dots, N - 1 \quad (9)$$

where the vector functions  $\hat{\Phi}_k$ ,  $k = 0, \dots, N$ , are defined in an implicit way.

One type of approximation consists in considering the low-order moments of  $p(x_k|I_k)$  as a sufficient statistics. This approach leads to estimators that are near optimal when the estimation error is small and the nonlinearities are mild. A classical example is the extended Kalman filter (EKF), where only the first- and second-order moments are considered. One way to refine this idea would be to use higher moments but this would result in approximations of the pdf in the proximity of its mean.

We adopt a different kind of approximation that involves collections of first- and second-order moments that do not concentrate only on one part of the pdf. This can be done by approximating the density  $p(x_k|I_k)$  as a sum of Gaussian densities where the covariance of each Gaussian is sufficiently small to allow the propagation of its mean and covariance by using the EKF algorithm [12].

Let us first introduce the following notation: given  $a \in \mathbb{R}^{\bar{q}}$ ,  $B \in \mathbb{R}^{\bar{q} \times \bar{q}}$ ,  $\mathcal{N}(a, B) \triangleq ((2\pi)^{-n/2}(\det B)^{-1/2})e^{-(1/2)a^T B^{-1}a}$

(irrespective of the dimension  $\bar{q}$  of  $a$ ). Then, if  $v$  is a normally distributed (Gaussian) random variable with the mean value  $\bar{v}$  and the covariance matrix  $Q$ , its pdf is denoted by  $\mathcal{N}(v - \bar{v}, Q)$ .

Let

$$\hat{p}(x, \hat{S}) = \sum_{i=1}^m \alpha_i \mathcal{N}(x - \mu_i, P_i) \quad (10)$$

$$\sum_{i=1}^m \alpha_i = 1, \quad \alpha_i \geq 0; \quad i = 1, \dots, m$$

be a Gaussian sum where  $\mu_i$ ,  $P_i$ , and  $\alpha_i$  are the mean, the covariance, and the weight of each Gaussian, respectively; the vector  $\hat{S}$  suitably collects all the parameters  $\alpha_i, \mu_i, P_i, i = 1, \dots, m$ , and  $m$  is a positive integer denoting the number of Gaussian terms. The effectiveness of the GSF is essentially based on the good approximating properties of the Gaussian sum density function (see Lemma 7.1 in the Appendix).

By using the parametric representation (10) to approximate  $p(x_k | I_k)$ , we obtain

$$\hat{p}(x_k, \hat{S}_k) = \sum_{i=1}^m \alpha_{ki} \mathcal{N}(x_k - \mu_{ki}, P_{ki}) \quad (11)$$

where the parameter vectors  $\hat{S}_k, k = 0, 1, \dots, N$ , have the same structure as  $\hat{S}$  thus involving the parameters  $\alpha_{ki}, \mu_{ki}$ , and  $P_{ki}, i = 1, \dots, m, k = 0, 1, \dots, N$ .

To iterate (11) from step  $k$  to step  $k + 1$  (after the acquisition of the new measure), we resort to the GSF equations (see the Appendix and [12] for details).

Along the lines of [12], two cases must be taken in account to iterate (11). The first is faced when the system noise covariance  $Q_k$  is “comparable” with that of the terms of the Gaussian sum  $P_{ki}$ . In this case, the GSF updating rule (see the Appendix) is sufficient to approximate the true (*a posteriori*) pdf to any degree of accuracy (see [12] and [19]).

The second case corresponds to the situation where the covariance  $Q_k$  is “large” as compared with  $P_{ki}$ . In this case, it may be necessary to introduce a Gaussian sum representation for  $p(\xi_k)$  to prevent all approximating Gaussians from collapsing into a single term and the GSF from reducing to a single EKF [12].

*Remark 3.1:* In the following, it will be clearer that the neural approximation method that we will apply to solve the AEP requires the differentiability of the filter; therefore, we will assume that the noises’ covariances “are not too large” so that we do not need to reapproximate the pdf. The differentiability assumption is also the main reason why we use a GSF and do not opt for more sophisticated Monte Carlo sampling techniques, like particle filters. Numerical results show that the proposed method works satisfactorily also for “nonnegligible” noises (see Section VI).

#### IV. INFORMATION MEASURES AND STATEMENT OF THE APPROXIMATE AEP

Let us address Problem 2.2. As mentioned in Section II, a quantitative measure of the information acquired on the state of the system is needed in order to specify the function  $H[p(x_N | I_N)]$  in (8). In this respect, several options are available. We focus our attention on two information measures: the

differential Shannon entropy and the differential Renyi entropy [20].

It is worth noting that the use of such information measures is widespread not only in information theory and statistics but, more recently, also in control theory and other applied-mathematics fields [1], [8]. The problem of active state estimation is strongly related to optimal experiment design [2]–[4]. All these references address, to some degree, the motivations for various different design criteria, of which the information theoretical criterion is a well-motivated one.

For the reader’s convenience, let us recall the following definitions.

*Definition 4.1:* The differential Shannon entropy  $H_1[p(x)]$  of a continuous random variable  $x \in \mathcal{X}$  is defined as

$$H_1[p(x)] \triangleq - \int_{\mathcal{X}} p(x) \log p(x) dx. \quad (12)$$

□

*Definition 4.2:* The differential Renyi entropy  $H_r[p(x)]$  of a continuous random variable  $x \in \mathcal{X}$  is defined as

$$H_r[p(x)] \triangleq \frac{1}{1-r} \log \int_{\mathcal{X}} [p(x)]^r dx \quad (13)$$

with  $r \in \mathbb{R}^+ \setminus \{1\}$ . □

The previous measures are scalar functionals with the pdf  $p(x)$  as argument. Moreover, it is worth noting that [20]

$$\lim_{r \rightarrow 1} H_r[p(x)] = H_1[p(x)].$$

Therefore, the differential Renyi entropy can be considered as a generalization of the differential Shannon entropy. If  $r = 2$ , (13) is called the quadratic Renyi entropy.

The functionals  $H_1[p(x)]$  and  $H_r[p(x)]$  have been introduced axiomatically by Shannon [21] and Renyi [11], respectively. In qualitative terms, a measure of uncertainty about a random variable must decrease if the associate pdf becomes more “concentrated” (see [20] and [22] or the seminal papers [11] and [21] for a complete treatment of the subject). Loosely speaking, maximally informative control functions should “shrink” the state pdf as much as possible, thus minimizing the measure of uncertainty.

Let us consider the expressions of the Shannon entropy (12) and of the Renyi entropy (13). By substituting (10) into (12) and (13), we obtain

$$H_1[\hat{p}(x, \hat{S})] = - \int_{\mathcal{X}} \hat{p}(x, \hat{S}) \log \hat{p}(x, \hat{S}) dx \quad (14)$$

and

$$H_r[\hat{p}(x, \hat{S})] = \frac{1}{1-r} \log \int_{\mathcal{X}} [\hat{p}(x, \hat{S})]^r dx \quad (15)$$

respectively. By inspection of (14), it becomes evident that the analytical integration of the Shannon entropy for a Gaussian sum is far from being simple. To the best of the authors’ knowledge, a closed-form expression of (14) is not known (see [20] for a table of closed-form expressions of entropies associated with some pdf). On the contrary, the quadratic Renyi entropy can be

calculated in closed form, as stated by the following simple result.

*Proposition 4.1:* Let  $x$  be a random variable such that

$$p(x) = \sum_{i=1}^m \alpha_i \mathcal{N}(x - \mu_i, P_i).$$

Then, the quadratic Renyi entropy  $H_2[p(x)]$  can be expressed in closed form by the expression

$$H_2[p(x)] = -\log[\alpha^T C \alpha]$$

where  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_m]^T$  and  $C$  is the symmetric matrix of elements  $c_{ji} = \mathcal{N}(\mu_j - \mu_i, P_j + P_i)$ .  $\square$

*Proof:* Let us recall the following result, valid for the product of two Gaussians:

$$\mathcal{N}(x - \mu_i, P_i) \mathcal{N}(x - \mu_j, P_j) = c_{ij} \mathcal{N}(x - \mu_{ij}, P_{ij})$$

where

$$\begin{aligned} c_{ij} &= c_{ji} = \mathcal{N}(\mu_i - \mu_j, P_i + P_j) \\ \mu_{ij} &= (P_i^{-1} + P_j^{-1})^{-1} (P_i^{-1} \mu_i + P_j^{-1} \mu_j) \\ P_{ij} &= (P_i^{-1} + P_j^{-1})^{-1}. \end{aligned}$$

Then,  $H_2[p(x)]$  can be computed as follows:

$$\begin{aligned} H_2[p(x)] &= -\log \int_{\mathbb{R}^n} \left[ \sum_{i=1}^m \alpha_i \mathcal{N}(X - \mu_i, P_i) \right]^2 dX \\ &= -\log \left[ \sum_{i=1}^m \sum_{j=1}^m \alpha_i c_{ij} \alpha_j \right] = -\log[\alpha^T C \alpha]. \end{aligned}$$

The approximation of  $p(x_k|I_k)$ ,  $k = 0, 1, \dots, N$ , in terms of suitable Gaussian sums [see (10)] enables us to reformulate Problem 2.2 in an approximate way. Moreover, the possibility of evaluating analytically the quadratic Renyi entropy for a Gaussian sum, and then to explicitly differentiating it (see also Remark 3.1) leads us to choose it as the measure of uncertainty to be adopted in the formulation of the next problem.

*Problem 4.1 (Approximate AEP):* Find the optimal control functions  $u_0^o = \bar{\gamma}_0^o(\hat{S}_0), \dots, u_{N-1}^o = \bar{\gamma}_{N-1}^o(\hat{S}_{N-1})$  that minimize the expected value of the cost function

$$\begin{aligned} \bar{J} &= \rho \left[ \sum_{k=0}^{N-1} g_k(x_k, \bar{\gamma}_k(\hat{S}_k)) + g_N(x_N) \right] \\ &\quad + (1 - \rho) H_2[\hat{p}(x_N, \hat{S}_N)] \end{aligned} \quad (16)$$

subject to (1a), (1b), and to the updating rule for  $\hat{S}_k$ .  $\square$

In the formulation of Problem 4.1 and in the following, for the sake of notational simplicity, we do not use a different symbol for the control variables generated by the proposed approximation schemes, that is, the control variables will always be denoted by  $u_k$ .

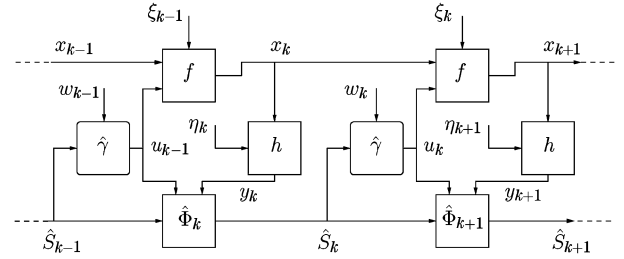


Fig. 2. Neural computational scheme.

Problem 4.1 corresponds to Problem 2.2 where the infinite dimensional sufficient statistic  $p(x_k|I_k)$  has been replaced by the approximate (finite-dimensional) sufficient statistic  $\hat{S}_k$ . It is evident that using a finite-dimensional vector instead of the pdf represents, in general, an approximation,<sup>2</sup> but enables us to address a “tractable solution” of the original AEP. The Renyi entropy of the approximate conditional pdf  $H_2[\hat{p}(x_N, \hat{S}_N)]$  (which is computable in closed form as shown in Proposition 4.1) is used instead of the generic term  $H[p(x_N|I_N)]$  in the cost function. Moreover, at each time index  $k = 0, 1, \dots, N - 1$ , the control vectors are generated as functions of  $\hat{S}_k$  (not of  $p(x_k|I_k)$ ), thus making it possible to implement the control functions. We want to remark here that the approximate sufficient statistics  $\hat{S}_k$  (used in Problem 4.1 as inputs to the control functions) can be chosen to have a time-invariant dimension. This makes the problem tractable even for very large values of  $N$  and will lead to the possibility of applying the proposed technique within the infinite-horizon framework (see Section V-B).

## V. NEURAL SOLUTION TO THE APPROXIMATE FUNCTIONAL OPTIMIZATION PROBLEM

Although Problem 4.1 has been obtained as an approximation of Problem 2.1, deriving its solution is still a very hard task. Hence, to face the functional optimization Problem 4.1, we resort to the extended Ritz method (ERIM) described in [14], that is, we constrain the control functions  $\bar{\gamma}_k(\hat{S}_k)$ ,  $k = 0, 1, \dots, N - 1$ , to take on fixed structures in which a certain number of parameters have to be determined in order to minimize the expected value of the cost function  $\bar{J}$ . More specifically, we replace the control functions  $\bar{\gamma}_k(\hat{S}_k)$  with

$$u_k = \hat{\gamma}(\hat{S}_k, w_k), \quad k = 0, 1, \dots, N - 1 \quad (17)$$

where  $\hat{\gamma}$  are nonlinearly parametrized approximating functions, and  $w_k$ ,  $k = 0, 1, \dots, N - 1$ , are the vectors of parameters to be determined. As fixed structures, we choose multilayer feedforward NNs (this choice is motivated by the nice approximating properties of such networks—see, again, [14]). In Fig. 2, the neural computational scheme is shown where  $\hat{\gamma}$  are the input/output mappings of multilayer feedforward NNs, and  $w_k$ ,  $k = 0, 1, \dots, N - 1$ , are the vectors of the synaptic weights.

<sup>2</sup>As already noted, the LG case is an exception. We want to recall here that an AEP would not make sense in the linear context, due to the “separation principle” holding in this case.

Without loss of generality, in the following we will focus on one-hidden-layer feedforward NNs (for the general multilayer case, the reader is referred, for instance, to [15]) and assume that the  $j$ th component  $\hat{\gamma}_j$  of the function  $\hat{\gamma}$  has the structure

$$\hat{\gamma}_j(\hat{S}_k, w_k) = \sum_{i=1}^{\nu^k} c_{ij}^k \varphi_i(\hat{S}_k, \kappa_i^k), \quad c_{ij}^k \in \mathbb{R}; \quad \kappa_i^k \in \mathbb{R}^{n_S} \quad (18)$$

for  $j = 1, \dots, p$ , where  $\varphi_i$  are given parametrized basis functions and  $n_S$  denotes the dimension of the argument vector  $\kappa_i^k$ . The scalars  $c_{ij}^k$  and the components of the vectors  $\kappa_i^k$  are the parameters to be determined in (17), i.e.,  $w_k \triangleq \text{col}(c_{ij}^k, \kappa_i^k : i = 1, \dots, \nu^k, j = 1, \dots, p)$ .

Let us now address the minimization of the expected value of the cost  $\bar{J}$  as assessed in Problem 4.1. Let us replace the control functions  $\bar{\gamma}_k(\hat{S}_k)$  in the cost function (16) with the neural control functions  $\hat{\gamma}(\hat{S}_k, w_k)$ , and substitute repeatedly (1a), (1b), and the updating rule  $\hat{\Phi}_k$  for  $\hat{S}_k$  so as to “eliminate” the vectors  $x_k, k = 1, 2, \dots, N$ , and  $\hat{S}_k, k = 0, 1, \dots, N$ . Then, the cost function takes on the form

$$\hat{J}(\hat{S}_{-1}, x_0, \xi, \eta, w) \quad (19)$$

where  $\xi \triangleq \text{col}(\xi_0, \dots, \xi_{N-1}), \eta \triangleq \text{col}(\eta_0, \dots, \eta_N)$ , and  $w \triangleq \text{col}(w_0, \dots, w_{N-1})$ .

Then, we have to solve the following problem.

**Problem 5.1 (Neural Approximate AEP):** Let  $\hat{S}_{-1}$  be the parameters of the Gaussian sum representation of the given  $p(x_0)$ . Find the optimal vector  $w^o$  that minimizes the expected value of the cost (19).  $\square$

It follows that the functional optimization Problem 4.1 has been reduced to the unconstrained nonlinear programming Problem 5.1, which is well suited to being solved by means of nonlinear programming techniques.

Owing to the smoothness of the mapping describing the GSF algorithm, to solve Problem 5.1, we can resort to gradient algorithms that are also very appealing for their simplicity (as we will show, this will enable us to introduce the “stochastic approximation” technique in a straightforward way).

We assume the functions  $f, h$ , and  $g_k, k = 0, 1, \dots, N$ , to have continuous first-order partial derivatives. To solve our problem, for a given  $\hat{S}_{-1}$ , the gradient algorithm can be defined as

$$w(l+1) = w(l) - \delta(l) \nabla_w \mathbb{E}_{x_0, \xi, \eta} \hat{J}[\hat{S}_{-1}, x_0, \xi, \eta, w(l)] \quad (20)$$

for  $l = 0, 1, \dots$ , where  $l$  denotes the iteration step of the descent procedure and  $\delta(l)$  are positive step-sizes. Due to the general statement of the problem, we are unable to express the average cost  $\mathbb{E}_{x_0, \xi, \eta} \hat{J}[\hat{S}_{-1}, x_0, \xi, \eta, w(l)]$  in explicit form. This leads us to compute the “realization”  $\nabla_w \hat{J}[\hat{S}_{-1}, x_0(l), \xi(l), \eta(l), w(l)]$  instead of the gradient appearing in (20). The sequence  $\{x_0(l), \xi(l), \eta(l), l = 0, 1, \dots\}$  is generated randomly, according to the pdfs of the vectors  $x_0, \xi, \eta$ . Then, in lieu of (20), we consider the following updating algorithm:

$$w(l+1) = w(l) - \delta(l) \nabla_w \hat{J}[\hat{S}_{-1}, x_0(l), \xi(l), \eta(l), w(l)] \quad (21)$$

for  $l = 0, 1, \dots$ . The probabilistic algorithm (21) is related to the concept of “stochastic approximation.” Sufficient conditions for the algorithm convergence can be found, for instance, in [16]. Some of such conditions are related to the behavior of the time-dependent step-size  $\delta(l)$ , the others to the shape of the cost surface  $\mathbb{E}_{x_0, \xi, \eta} \hat{J}[\hat{S}_{-1}, x_0, \xi, \eta, w]$ . As to the step-size  $\delta(l)$ , we have the following conditions for the algorithm convergence:

$$\delta(l) > 0 \quad \sum_{l=0}^{\infty} \delta(l) = \infty \quad \sum_{l=0}^{\infty} \delta(l)^2 < \infty. \quad (22)$$

In the simulation examples given in the following, we take  $\delta(l) = c_1/(c_2 + l)$ ,  $c_1, c_2 > 0$ , which satisfies the conditions (22) (we do not consider more sophisticated forms for the step-size as the issue of convergence speed is beyond the scope of this paper). Concerning the cost surface  $\mathbb{E}_{x_0, \xi, \eta} \hat{J}[\hat{S}_{-1}, x_0, \xi, \eta, w]$ , to verify if the conditions for the convergence of the weight vector to a global minimum are fulfilled is clearly a hard task, due to the high complexity of such a surface. Actually, how to avoid getting stuck in local minima still remains an open issue in optimizing NNs, even though a vast body of experimental works presented in the literature confirms the robustness of simple descent algorithms based on the gradient method. Some recent theoretical results tend to support these empirically established characteristics (and also the close relationship between stochastic approximation and the well-known backpropagation procedure described in the following; see, for instance, [23]).

Now, we have to determine the gradient

$$\nabla_w \hat{J}[\hat{S}_{-1}, x_0(l), \xi(l), \eta(l), w(l)] = \text{col} \left( \left[ \frac{\partial \hat{J}}{\partial w_k} \right]^T \right)$$

where  $k = 0, 1, \dots, N-1$ .

#### A. Forward–Backward Computation of the Partial Derivatives

We have

$$\frac{\partial \hat{J}}{\partial w_k} = \frac{\partial \hat{J}}{\partial u_k} \frac{\partial}{\partial w_k} \hat{\gamma}(\hat{S}_k, w_k).$$

Following the computational procedure reported, for instance, in [15], it can be shown that such a computation corresponds to the application of a backpropagation procedure initialized with the vector  $\partial \hat{J} / \partial u_k$ , that is, with the sensitivity of the cost function with respect to the output of the  $k$ th neural control function.

To determine the partial derivatives  $\partial \hat{J} / \partial u_k, k = 0, 1, \dots, N-1$ , we can write

$$\frac{\partial \hat{J}}{\partial u_k} = \rho \frac{\partial}{\partial u_k} g_k(x_k, u_k) + \rho \frac{\partial}{\partial u_k} \left[ \sum_{s=k+1}^{N-1} g_s(x_s, u_s) + g_N(x_N) \right] + \frac{\partial}{\partial u_k} \left\{ (1-\rho) H_2 \left[ \hat{p}(x_N, \hat{S}_N) \right] \right\}. \quad (23)$$

Note that the second and third terms in (23) represent the derivative of the so called “cost-to-go” at stage  $k+1$  with respect to  $u_k$ . As can be noticed from the scheme depicted in Fig. 2, this cost is influenced by  $u_k$  through the state vector  $x_{k+1}$  generated

by the state equation, and through the mapping  $\hat{\Phi}_{k+1}$ , which has  $u_k$  as its argument. Then, we rewrite (23) as follows:

$$\begin{aligned} \frac{\partial \hat{J}}{\partial u_k} &= \rho \frac{\partial}{\partial u_k} g_k(x_k, u_k) + \frac{\partial \hat{J}}{\partial x_{k+1}} \frac{\partial x_{k+1}}{\partial u_k} \\ &\quad + \frac{\partial \hat{J}}{\partial \hat{S}_{k+1}} \frac{\partial}{\partial u_k} \hat{\Phi}_{k+1}(\hat{S}_k, u_k, y_{k+1}) \end{aligned}$$

and hence, by letting

$$\lambda_k \triangleq \left( \frac{\partial \hat{J}}{\partial x_k} \right)^T \quad (24)$$

and

$$\psi_k \triangleq \left( \frac{\partial \hat{J}}{\partial \hat{S}_k} \right)^T \quad (25)$$

we obtain

$$\begin{aligned} \frac{\partial \hat{J}}{\partial u_k} &= \rho \frac{\partial}{\partial u_k} g_k(x_k, u_k) + \lambda_{k+1}^T \frac{\partial}{\partial u_k} f(x_k, u_k) \\ &\quad + \psi_{k+1}^T \frac{\partial}{\partial u_k} \hat{\Phi}_{k+1}(\hat{S}_k, u_k, y_{k+1}). \end{aligned}$$

To determine the vectors  $\lambda_k^T$ , from Fig. 2, we see that  $x_k$  influences the cost directly through  $g_k(x_k, u_k)$  and the remaining part of the cost through  $x_{k+1}$  and  $y_k$ . Then, we can write

$$\begin{aligned} \lambda_k^T &= \rho \frac{\partial}{\partial x_k} g_k(x_k, u_k) + \lambda_{k+1}^T \frac{\partial}{\partial x_k} f(x_k, u_k) \\ &\quad + \psi_k^T \frac{\partial}{\partial y_k} \hat{\Phi}_k(\hat{S}_{k-1}, u_{k-1}, y_k) \frac{\partial}{\partial x_k} h(x_k, \eta_k) \quad (26) \end{aligned}$$

for  $k = N-1, N-2, \dots, 0$ . As to the determination of the vectors  $\psi_k^T$ , from Fig. 2, we observe that  $\hat{S}_k$  influences the cost through the  $k$ th neural control function (which has  $\hat{S}_k$  as its argument) and through the mapping  $\hat{\Phi}_{k+1}$  (which, again, has  $\hat{S}_k$  as its argument). Then, from the definition (25), it follows that:

$$\psi_k^T = \frac{\partial \hat{J}}{\partial u_k} \frac{\partial}{\partial \hat{S}_k} \hat{\gamma}(\hat{S}_k, w_k) + \psi_{k+1}^T \frac{\partial}{\partial \hat{S}_k} \hat{\Phi}_{k+1}(\hat{S}_k, u_k, y_{k+1}) \quad (27)$$

for  $k = N-1, N-2, \dots, 0$ . The term  $(\partial \hat{J} / \partial u_k) (\partial / \partial \hat{S}_k) \hat{\gamma}(\hat{S}_k, w_k)$  represents the sensitivity of the cost function with respect to the actual input of the  $k$ th neural control function. As shown in [15], it can be computed quite easily by exploiting the backpropagation procedure through the neural control functions  $\hat{\gamma}$  at stage  $k$ .

The recursive equation (27) is initialized as follows:

$$\psi_N^T = (1 - \rho) \frac{\partial H_2 \left[ \hat{p}(x_N, \hat{S}_N) \right]}{\partial \hat{S}_N}.$$

As to the initialization of (26), note that

$$\begin{aligned} \lambda_N^T &= \rho \frac{\partial}{\partial x_N} g_N(x_N) + (1 - \rho) \frac{\partial}{\partial x_N} H_2 \left[ \hat{p}(x_N, \hat{S}_N) \right] \\ &= \rho \frac{\partial}{\partial x_N} g_N(x_N) + (1 - \rho) \frac{\partial}{\partial \hat{S}_N} H_2 \left[ \hat{p}(x_N, \hat{S}_N) \right] \\ &\quad \times \frac{\partial}{\partial y_N} \hat{\Phi}_N(\hat{S}_{N-1}, u_{N-1}, y_N) \frac{\partial}{\partial x_N} h(x_N, \eta_N) \end{aligned}$$

and, hence

$$\begin{aligned} \lambda_N^T &= \rho \frac{\partial}{\partial x_N} g_N(x_N) \\ &\quad + \psi_N^T \frac{\partial}{\partial y_N} \hat{\Phi}_N(\hat{S}_{N-1}, u_{N-1}, y_N) \frac{\partial}{\partial x_N} h(x_N, \eta_N). \end{aligned}$$

The computation of the partial derivatives  $(\partial / \partial \hat{S}_k) \hat{\Phi}_{k+1}(\hat{S}_k, u_k, y_{k+1})$ ,  $(\partial / \partial y_k) \hat{\Phi}_k(\hat{S}_{k-1}, u_{k-1}, y_k)$ , and  $(\partial / \partial u_k) \hat{\Phi}_{k+1}(\hat{S}_k, u_k, y_{k+1})$ , though conceptually simple, is rather cumbersome because it involves the differentiation of the mapping (9) and is not reported here for lack of space. The reader is referred to [24] for more details.

Finally, let us briefly describe the learning mechanism. The following two ‘‘passes’’ alternate up to convergence.

#### Forward pass.

The initial state  $x_0(l)$  and the random vectors  $\xi_k(l)$ ,  $k = 0, 1, \dots, N-1$ , and  $\eta_k(l)$ ,  $k = 0, 1, \dots, N$ , are randomly generated according to their pdfs. Then, the state and control trajectories are computed by means of the neural control functions characterized by the current weight vectors  $w_0(l), \dots, w_{N-1}(l)$ .

#### Backward pass.

The components of  $\nabla_w \hat{J}[\hat{S}_{-1}, x_0(l), \xi(l), \eta(l), w(l)]$  are computed backwards using the aforementioned recursive relations. Then, the new weight vector  $w(l+1)$  is generated by (21).

### B. Extension to the RH Framework

Up to this point, we have considered the AEP in an FH framework. This is consistent with many practical situations where a ‘‘manoeuvre’’ must be performed within a fixed time in order to acquire information about the final state of the system. In many other situations, when the system has to be controlled over a very long time horizon (or when the time horizon is not *a priori* fixed), an optimal control problem is usually stated as an infinite-horizon one.

When controlling the system online, the approximate sufficient statistic  $\hat{S}_k$  can be propagated for  $k = 0, 1, \dots$  by means of the GSF described in Section III.

An approximate technique that is often adopted to obtain time-invariant control functions to be applied on line consists in the so-called RH approach (see, for instance, [25]). Within such a framework, the technique presented in this paper can be suitably applied.

One could think of using the control function  $\hat{\gamma}(\cdot, w_0)$  corresponding to  $k = 0$  as a time-invariant control function in order to generate the control vectors online for  $k = 0, 1, \dots$  as  $u_k = \hat{\gamma}(\hat{S}_k, w_0)$ .

Of course if, in the (offline) optimization phase, the initial conditions (i.e., the parameters  $\hat{S}_{-1}$  characterizing the initial approximate pdf  $\hat{p}(x_0, \hat{S}_{-1})$ ) are generated to approximate the *a priori* pdf  $p(x_0)$ , then the NN  $\hat{\gamma}(\cdot, w_0)$  will turn out to be “specialized” for inputs typically corresponding to “wide” distributions (large values of the corresponding entropy). When controlling the system, the approximate pdf will become sharper and sharper, whereas the entropy will (hopefully) decrease and the control function  $\hat{\gamma}(\cdot, w_0)$  will not be able to generate appropriate control vectors. In order to overcome such a drawback, the parameter vector  $\hat{S}_{-1}$  will not be regarded as being fixed, but dependent on the optimization step  $l$ . Consequently, we will denote it by  $\hat{S}_{-1}(l)$ , and the optimization procedure will be suitably modified in the following way.

Let us consider a fixed length  $N$  of the control window and an integer  $L$  characterizing the number of successive optimization steps.

At the first step (i.e., step  $l = 0$ ),  $\hat{S}_{-1}(0)$  is initialized in order to approximate the *a priori* pdf (e.g., it can be set to the fixed value  $\hat{S}_{-1}$ ). Then, the first optimization step (consisting in a forward and a backward pass) is performed as described in Section V-A. An integer  $\bar{k}(0) \in [1, N]$  is randomly chosen and the corresponding approximate sufficient statistic  $\hat{S}_{\bar{k}(0)}(0)$  generated in the forward pass is used to approximate  $p(x_0)$  at step  $l = 1$  (i.e., we set  $\hat{S}_{-1}(1) = \hat{S}_{\bar{k}(0)}(0)$ ). Then, the corresponding forward and backward passes are performed for step  $l = 1$ . An integer  $\bar{k}(1) \in [1, N]$  is randomly chosen, and we set  $\hat{S}_{-1}(2) = \hat{S}_{\bar{k}(1)}(1)$ . The same mechanism is applied for  $l = 2, 3, \dots, L - 1$  where  $L$  is suitably large.

This mechanism allows the generation, in the offline optimization phase, of a sequence of  $L$  optimization steps with initial conditions corresponding to time instants  $\bar{t}(0) = 0$  (for  $l = 0$ ),  $\bar{t}(1) = \bar{k}(0)$  (for  $l = 1$ ),  $\bar{t}(2) = \bar{k}(0) + \bar{k}(1)$  (for  $l = 1$ ),  $\dots$ ,  $\bar{t}(L - 1) = \sum_{i=0, \dots, L-2} \bar{k}(i)$  (for  $l = L - 1$ ).

At step  $l = L$ , the initial pdf is generated once again to approximate the *a priori* knowledge of  $x_0$  (e.g., by letting  $\hat{S}_{-1}(L) = \hat{S}_{-1}$ ). Then, the previous procedure is iterated at steps  $L + 1, \dots, 2L - 1$  and for successive batches of  $L$  steps.

Since, in our RH framework, we will apply a time-invariant control function when controlling the system online, we force such a behavior also in the offline optimization phase, and we use  $N$  identical neural control functions in the control window, i.e., we set

$$w_k = w, \quad k = 0, 1, \dots, N - 1. \quad (28)$$

As a result, we have a single parameter vector entering all the neural control functions at stages  $k = 0, 1, \dots, N - 1$ .<sup>3</sup> In the following, we will go on denoting by  $w_k$  the parameter vectors entering the neural control function at each stage  $k = 0, 1, \dots, N - 1$  (see Fig. 3). Thanks to this “redefinition,” the forward pass remains formally unchanged with respect to that described in Section V-A.

As to the backward pass, the same computations as in Section V-A have to be carried out, in order to obtain the

<sup>3</sup>Note that, with a little abuse of notation, here,  $w$  is not defined as  $\text{col}(w_k, k = 0, 1, \dots, N - 1)$  but as the “unique” parameter vector characterizing all the neural control functions.

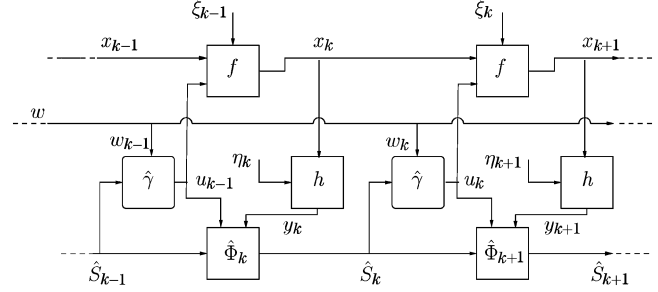


Fig. 3. Neural computational scheme used in the RH framework.

partial derivatives  $\partial \hat{J} / \partial w_k$ . According to (28), the gradient  $\nabla_w \hat{J}[\hat{S}_{-1}(l), x_0(l), \xi(l), \eta(l), w(l)]$  is obtained as follows:

$$\nabla_w \hat{J}[\hat{S}_{-1}(l), x_0(l), \xi(l), \eta(l), w(l)] = \left[ \sum_{k=0, 1, \dots, N-1} \frac{\partial \hat{J}}{\partial w_k} \right]^T$$

and the updating algorithm (21) can be applied at each step  $l$ .

## VI. NUMERICAL RESULTS

In this section, we provide two numerical examples to show the effectiveness of the active estimation scheme proposed in this paper. The two examples are “active” versions of nonlinear tracking problems known as bearings only measurements problems. The “passive” versions (i.e., the problems where the control sequence is not chosen to maximize the information on the state vector) have been adopted by many authors as benchmark examples to propose new filtering techniques. For these problems, the EKF often exhibits an unstable behavior [12], [26]. The active versions presented here are generalizations of the aforementioned problems where the objective is to design a control strategy to improve the filtering performance.

### A. Example 1

Consider the following discrete-time dynamic system:

$$\begin{aligned} x_{k+1} &= x_k + u_k + \xi_k, & k=0, 1, \dots \\ y_k &= \tan^{-1} \left( \frac{P_y - r \sin x_k}{P_x - r \cos x_k} \right) + \eta_k, & k=0, 1, \dots \end{aligned} \quad (29)$$

where  $\xi_k$  and  $\eta_k$  are two white, independent, Gaussian noises with zero mean and variances  $\sigma_\xi^2 = \sigma_\eta^2 = 0.1$ . The initial state  $x_0$  is a random variable uniformly distributed in the interval  $[0, 2\pi]$ . The system represents an observer moving on a circle of radius  $r$ . The observer’s objective is to estimate its angular position  $x_k$  by means of the measurements of the angle  $y_k$ ,  $k = 0, 1, \dots$ . In Fig. 4, a geometric representation of the problem is given.

The following FH cost was considered:

$$J_N = \rho \sum_{k=0}^{N-1} u_k^2 + (1 - \rho) H_2 \left[ \hat{p}(x_N, \hat{S}_N) \right] \quad (30)$$

where  $N = 18$  and  $\rho = 0.1$ . The control functions were implemented by means of one-hidden-layer feedforward NNs with 20 sigmoidal units.

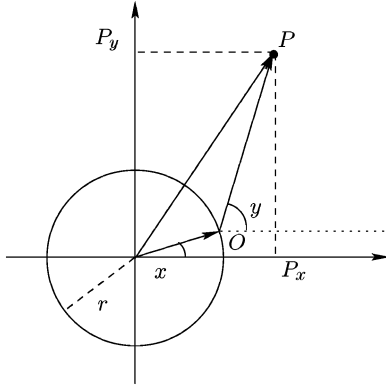


Fig. 4. Geometric representation of the problem.

For the sake of comparison, the active estimation technique proposed in this paper (i.e., the combination of a GSF with the approximation of the control functions by means of NNs) was compared with two different strategies. The first consisted in the same control technique but the EKF was used instead of the GSF to approximate the conditional pdf. The second corresponded to the application of a random control action coupled with the GSF.

Two quantities were considered in order to evaluate the performance of the aforementioned techniques: the mean square estimation error (mse) and the mean value of the quadratic Renyi entropy. The mse is defined as

$$\text{mse}(k) = 1/S \sum_{s=1}^S [x_k^{(s)} - \hat{x}_k^{(s)}]^2, \quad k = 0, 1, \dots$$

where  $S$  is the number of simulation runs and  $x_k^{(s)}$  and  $\hat{x}_k^{(s)}$  are the values of the state vector at time  $k$  and of its estimate, respectively. The mean value corresponding to the approximate pdf  $\hat{p}(x_k, \hat{S}_k)$  was chosen as an estimate  $\hat{x}_k$  of the state vector at each time  $k$ . Similarly, the mean value of the quadratic Renyi entropy is defined as

$$\bar{H}_2(\hat{S}_k) = 1/S \sum_{s=1}^S H_2 \left[ \hat{p} \left( x_k, \hat{S}_k^{(s)} \right) \right], \quad k = 0, 1, \dots$$

To show the effectiveness of the approach,  $S = 2 \cdot 10^4$  simulation runs were performed by extracting the initial state randomly according to a uniform pdf in the interval  $[0, 2\pi]$ .

In Fig. 5(a) and (b), the evolution of the mse and of the Renyi entropy, respectively, are plotted for the three different control strategies.

It can be seen that the neural control technique coupled with the GSF outperforms the others in terms of both entropy values and mse. If the EKF is applied, even if the entropy decreases the mse turns out to be almost constant. This happens because, in this example, the EKF fails to give an accurate estimate. This does not happen if the GSF is used instead. This fact points out that, in the proposed technique, both the controller and the estimator play a central role, and, if an unsuitable estimator is used, the method may yield unsatisfactory results.

Two examples of trajectories followed by the system are depicted in Fig. 6 as polar plots, where the angle represents

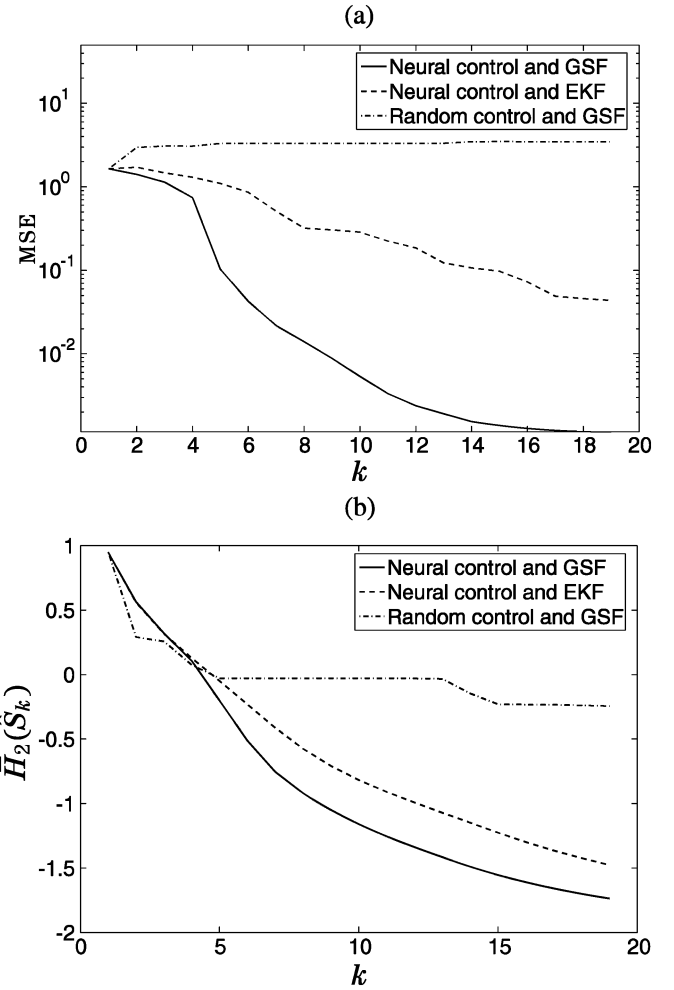


Fig. 5. Evolution of the mse and of the mean value of the Renyi entropy on a finite horizon ( $N = 18$ ) (the mse is plotted with a logarithmic scale).

the state  $x_k$  and the radial distance increases over time. The dashed lines show the angles at which the measurement channel presents an extremum.

In Fig. 7, an example of evolution of the approximate pdf is presented.

The methodology was tested also in an RH framework. The same cost function as before was considered in the optimization phase. Also in this case,  $2 \cdot 10^4$  simulation runs were performed by extracting the initial state according to the initial pdf. In Fig. 8(a) and (b), the evolutions of the mse and of the mean value of the Renyi entropy are shown.

Here, the RH optimization procedure described in Section V-B was applied. The same comparisons were performed as in the FH case. Even in this case, the methodology described in the paper outperformed both the random control approach and the neural approach combined with a standard EKF (similar considerations as before can be done).

## B. Example 2

Consider the following AEP. A target, whose position is denoted by  $x_{Bk} \in \mathbb{R}^2$ , moves in the plane at constant velocity  $v_B$ . An observer  $A$ , whose position is denoted by  $x_{Ak} \in \mathbb{R}^2$ , has to

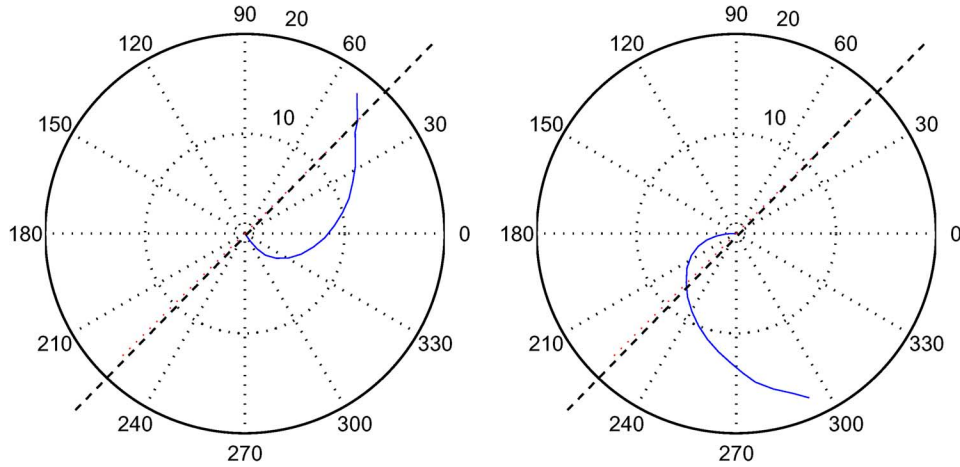


Fig. 6. Two examples of trajectories.

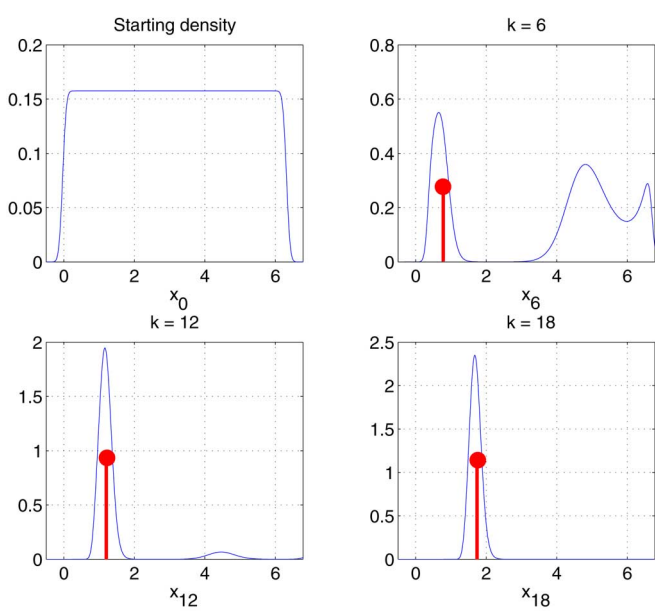


Fig. 7. Example of the evolution of the approximate pdf.

estimate the relative position  $x_k = [x_k^{(1)}, x_k^{(2)}]^T = x_{Bk} - x_{Ak}$ , by following the “most appropriate” trajectory, in order to track the uncertain target  $B$ . Only bearing measurements are allowed, i.e., the observer  $A$  can only measure the angle

$$\theta_k = \tan^{-1} \left( \frac{x_k^{(2)}}{x_k^{(1)}} \right)$$

through a noisy measurement channel. Such a situation is depicted in Fig. 9.

The system is described by the model

$$\begin{aligned} x_{B(k+1)} &= x_{Bk} + cv_B \\ x_{A(k+1)} &= x_{Ak} + u_k - \xi_k \\ y_k &= \theta_k + \eta_k, \quad k = 0, 1, \dots, N \end{aligned}$$

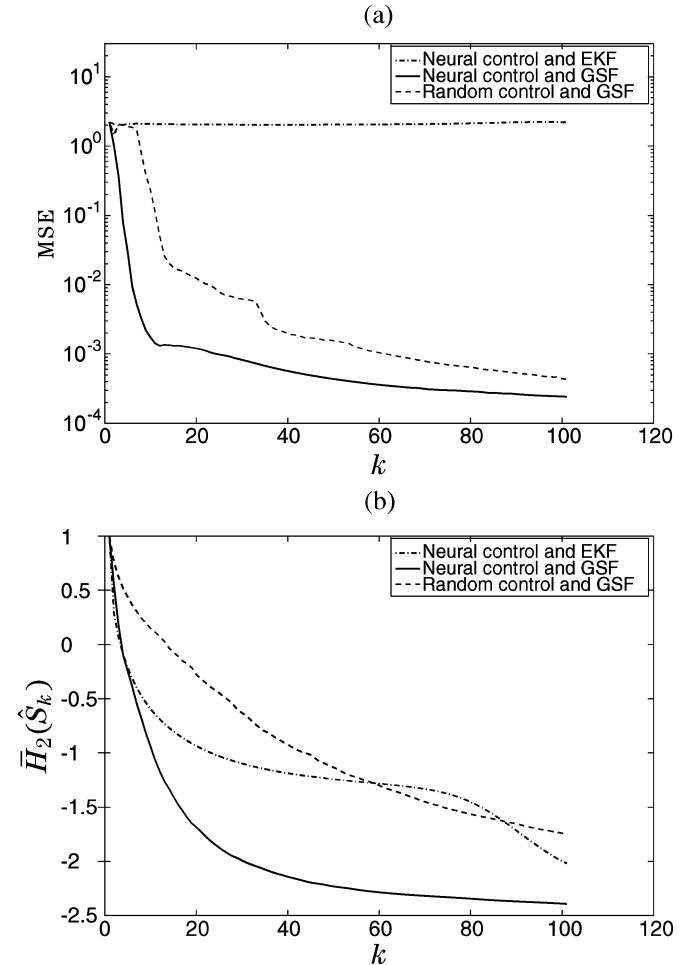


Fig. 8. Evolution of the mse and of the average Renyi entropy (the mse is plotted with a logarithmic scale).

where  $u_k \in \mathbb{R}^2$  is the control vector and  $\xi_k$  and  $\eta_k$  are two white, independent, Gaussian noises with zero mean values. The covariance matrix for  $\xi_k$  is

$$\Sigma_\xi = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

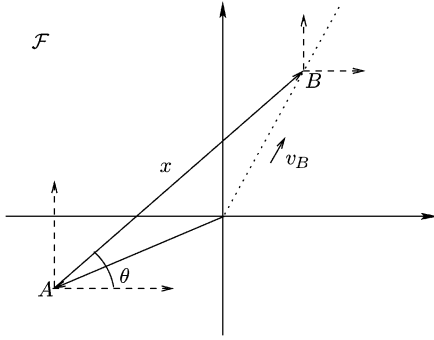


Fig. 9. Geometric representation of the problem.

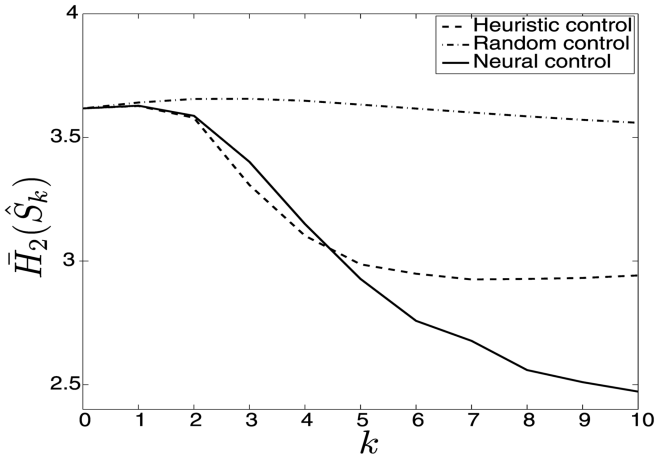


Fig. 10. Evolution of the mean value of the entropy.

and the variance of  $\eta_k$  is  $\sigma_\eta = 0.1$ . The scalar  $c \in \mathbb{R}^+$  is a discretization parameter. The model can be formulated equivalently by considering the system state as the relative position  $x_k = x_{Bk} - x_{Ak}$

$$\begin{aligned} x_{k+1} &= x_k - u_k + cv_B + \xi_k \\ y_k &= \theta_k + \eta_k, \quad k = 0, 1, \dots, N \end{aligned}$$

where the term  $cv_B + \xi_k$  can be regarded as a white Gaussian noise with mean value  $cv_B$  and covariance matrix  $\Sigma_\xi$ .

The neural approximators were trained to minimize the cost function (30) and the control horizon was set to  $N = 10$ . In Fig. 10, the entropy evolution is presented by comparing the neural technique proposed in this paper with the application of a random control and with a heuristic control strategy (aiming at maximizing the measured angles  $y_k$  along the trajectory).

In Fig. 11, the evolution of the approximate pdf is shown for a given instance of the problem. It can be seen that at the final stage the density function is almost centered on the true state of the system.

## VII. CONCLUDING REMARKS

The problem of actively estimating the state of a stochastic dynamic system has been addressed. The formulation follows from a stochastic optimal control approach where a suitable

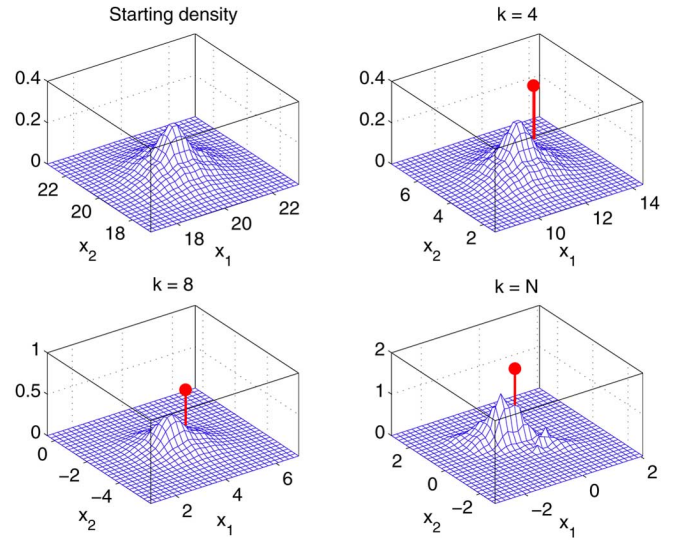


Fig. 11. Example of evolution of the approximate pdf.

functional measuring the uncertainty about the state vector has been added to the process cost. The Renyi entropy has been proposed and motivated as a “measure of information.” A neural scheme has been proposed for the approximation of the control functions. In particular, the proposed technique incorporates a GSF into the ERIM described in [14]. The problem has been formulated in a finite horizon framework and the extension to the infinite-horizon case has been addressed by using an RH methodology. The simulation results on two significant case studies confirm the effectiveness of the proposed methodology.

The extension of the technique to more general AEPs will deserve further research. Moreover, considerable efforts will be devoted to enhance the theoretical analysis, especially towards relating the performances of the estimation schemes with the complexity of the neural approximators implementing the control functions.

## APPENDIX GAUSSIAN SUM FILTERING

The following Lemma (see [12] and [27]) states the good approximating properties of the Gaussian sum density function.

*Lemma 7.1:* Any pdf  $p(x)$  can be approximated as closely as desired in the space  $L_1(\mathbb{R}^n)$  by a Gaussian sum representation  $\hat{p}(x, \hat{S})$ , as defined in (10) for some integer  $m$ , positive scalars  $\alpha_i$ , vectors  $\mu_i$ , and positive-definite matrices  $P_i$ .  $\square$

In the following sections, we review the GSF equations.

### A. Prediction Equation

First, we describe how to calculate the one-step-ahead prediction density on the basis of

$$\hat{p}(x_{k-1}, \hat{S}_{k-1}) = \sum_{i=1}^m \alpha_{(k-1)i} \mathcal{N}(x_{k-1} - \mu_{(k-1)i}, P_{(k-1)i}). \quad (31)$$

By applying the EKF theory, one obtains the expression for the one-step predicted estimate of each Gaussian distribution

$$\hat{p}(x_k, \hat{S}'_k) = \sum_{i=1}^m \alpha'_{ki} \mathcal{N}(x_k - \mu'_{ki}, P'_{ki}) \quad (32)$$

by the updating rule

$$\begin{aligned} \mu'_{ki} &= f(\mu_{(k-1)i}, u_{k-1}) \\ P'_{ki} &= F_{ki} P_{(k-1)i} F_{ki}^T + Q_{k-1} \\ \alpha'_{ki} &= \alpha_{k(i-1)} \end{aligned} \quad (33)$$

where

$$F_{ki} \triangleq \left. \frac{\partial f}{\partial x_{k-1}} \right|_{\mu_{(k-1)i}, u_{k-1}}$$

### B. Innovation Equation

Now, we establish how to calculate  $\hat{p}(x_k, \hat{S}_k)$  from  $\hat{p}(x_k, \hat{S}'_k)$  when a new measurement  $y_k$  becomes available. From

$$\hat{p}(x_k, \hat{S}'_k) = \sum_{i=1}^m \alpha'_{ki} \mathcal{N}(x_k - \mu'_{ki}, P'_{ki}) \quad (34)$$

we can calculate

$$\hat{p}(x_k, \hat{S}_k) = \sum_{i=1}^m \alpha_{ki} \mathcal{N}(x_k - \mu_{ki}, P_{ki}) \quad (35)$$

by using the Kalman filter update rule and a normalization operator (see [12] for details). This leads to

$$\begin{aligned} \mu_{ki} &= \mu'_{ki} + K_{ki} [y_k - h(\mu'_{ki})] \\ P_{ki} &= P'_{ki} - K_{ki} H_{ki} P'_{ki} \\ \alpha_{ki} &= \frac{\alpha'_{ki} \beta_{ki}}{\sum_{j=1}^m \alpha'_{kj} \beta_{kj}} \end{aligned} \quad (36)$$

where

$$\begin{aligned} H_{ki} &\triangleq \left. \frac{\partial h}{\partial x_k} \right|_{\mu'_{ki}} \\ T_{ki} &\triangleq H_{ki} P'_{ki} H_{ki}^T + R_k \\ K_{ki} &\triangleq P'_{ki} H_{ki}^T T_{ki}^{-1} \\ \beta_{ki} &\triangleq \mathcal{N}[y_k - h(\mu'_{ki}), T_{ki}]. \end{aligned}$$

### REFERENCES

- [1] K. A. Loparo, X. Feng, and Y. Fang, "Optimal state estimation for stochastic systems: An information theoretic approach," *IEEE Trans. Autom. Control*, vol. 42, no. 6, pp. 771–785, Jun. 1997.
- [2] V. V. Fedorov, *Theory of Optimal Experiments*. New York: Academic, 1972.
- [3] D. MacKay, "Information-based objective functions for active data selection," *Neural Comput.*, vol. 4, pp. 590–604, 1992.
- [4] Z. Ghahramani, D. A. Cohn, and M. I. Jordan, "Active learning with statistical models," *J. Artif. Intell. Res.*, vol. 4, pp. 129–145, 1996.
- [5] M. Baglietto, M. Paolucci, L. Scardovi, and R. Zoppoli, "Information-based multi-agent exploration," in *Proc. IEEE 3rd Int. Workshop Robot Motion Control*, Bukowy Dworek, Poland, 2002, pp. 173–179.

- [6] M. Baglietto, M. Paolucci, L. Scardovi, and R. Zoppoli, "Entropy-based environment exploration and stochastic optimal control," in *Proc. 42nd IEEE Conf. Decision Control*, Maui, HI, 2003, pp. 2938–2941.
- [7] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, Monterey, CA, 1997, pp. 146–151.
- [8] G. N. Saridis, "Entropy formulation of optimal and adaptive control," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 713–721, Aug. 1988.
- [9] M. Baglietto, G. Cannata, L. Scardovi, and R. Zoppoli, "Active identification of unknown systems," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, 2005, pp. 452–457.
- [10] M. Baglietto, L. Scardovi, and R. Zoppoli, "Active identification of unknown systems: An information theoretic approach," in *Proc. 42nd IEEE Amer. Control Conf.*, Boston, MA, 2004, pp. 3826–3830.
- [11] A. Renyi, "On measures of entropy and information," in *Proc. 4th Berkeley Symp. Math. Statist. Probab.*, Berkeley, CA, 1961, pp. 547–561.
- [12] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximation," *IEEE Trans. Autom. Control*, vol. AC-17, no. 4, pp. 439–448, Aug. 1972.
- [13] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–944, May 1993.
- [14] R. Zoppoli, M. Sanguineti, and T. Parisini, "Approximating networks and the extended Ritz Method for the solution of functional optimization problems," *J. Optim. Theory Appl.*, vol. 112, pp. 403–439, 2002.
- [15] T. Parisini and R. Zoppoli, "Neural networks for feedback feedforward nonlinear control systems," *IEEE Trans. Neural Netw.*, vol. 5, no. 3, pp. 436–449, May 1994.
- [16] H. J. Kushner and G. Yin, *Stochastic Approximation Algorithms and Applications*. New York: Springer-Verlag, 1997.
- [17] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 2001.
- [18] A. A. Feldbaum, *Optimal Control Systems*. New York: Academic, 1965.
- [19] B. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [20] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [21] C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.
- [22] I. Shunsuke, *Information Theory for Continuous Systems*. Singapore: World Scientific, 1993.
- [23] W. Finnoff, "Diffusion approximations for the constant learning rate backpropagation algorithm and resistance to local minima," *Neural Comput.*, vol. 6, pp. 285–295, 1994.
- [24] L. Scardovi, "Information-Based Control for State and Parameter Estimation" Ph.D. dissertation, Dept. Commun., Comp. Syst. Sci., DIST, Univ. Genoa, Genova, Italy, April 2005 [Online]. Available: <http://www.dist.unige.it/lscardovi/phd.pdf>
- [25] E. Camacho and C. Bordons, *Model Predictive Control*. New York: Springer-Verlag, 2005.
- [26] A. Alessandri, T. Parisini, and R. Zoppoli, "Neural approximators for nonlinear finite-memory state estimation," *Int. J. Control*, vol. 67, pp. 275–301, 1997.
- [27] F. Girosi, "Regularization theory, radial basis functions and networks," in *From Statistics to Neural Networks. Theory and Pattern Recognition Applications*, J. H. Friedman, V. Cherkassky, and H. Wechsler, Eds. Berlin, Germany: Springer-Verlag, 1993.



**Luca Scardovi** (M'06) was born in Albenga, Italy, in 1977. He received the "Laurea" degree in computer science engineering and the Ph.D. degree in electronic engineering and computer science from the University of Genoa, Genova, Italy, in 2001 and 2005, respectively.

In 2005, he was an Adjunct Professor of System Identification and Data Analysis at the University of Lecce, Italy. Since 2005, he has been a Postdoctoral Research Associate at the Department of Electrical Engineering and Computer Science, University of Liège, Liège, Belgium. His research interests include synchronization and consensus problems, collective motion, estimation and filtering, switching systems, and NNs.



**Marco Baglietto** (M'04–A'05) was born in Savona, Italy, in 1970. He received the “Laurea” degree in electronic engineering and the Ph.D. degree in electronic engineering and computer science from the University of Genoa, Genova, Italy, in 1995 and 1999, respectively.

Since 1999, he has been an Assistant Professor of Automatic Control at the Department of Communications, Computer and Systems Science (DIST), University of Genoa. His research interests include neural approximations, state estimation for linear

and nonlinear systems, switching systems, distributed-information control systems, and control of communication networks.

Dr. Baglietto is currently an Associate Editor for the IEEE Control Systems Society Conference Editorial Board. He was a Guest Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS Special Issue on Adaptive Learning Systems in Communication Networks. He is the corecipient of the 2004 Outstanding Paper Award of the IEEE TRANSACTIONS ON NEURAL NETWORKS.



**Thomas Parisini** (S'87–M'88–SM'99) was born in Genoa, Genoa, Italy, in 1963. He received the “Laurea” degree (*cum laude* and printing honors) in electronic engineering from the University of Genoa, in 1988 and the Ph.D. degree in electronic engineering and computer science in 1993.

From 1988 to 1995, he was with the Department of Communications, Computer and Systems Sciences (DIST), University of Genoa. In 1995, he joined the Department of Electrical, Electronic and Computer Engineering (DEEI), University of Trieste, Trieste,

Italy, as an Assistant Professor, and in 1998, he joined the Department of Electronic and Information Engineering (DEI), Politecnico di Milano, as an Associate Professor. In 2001, he was appointed Full Professor and Danieli Endowed Chair of Automation Engineering at DEEI, University of Trieste. His research interests include NN approximations for optimal control and filtering problems, fault diagnosis for nonlinear systems, hybrid control systems and control of distributed systems. He is involved as Project Leader in several projects funded by the European Union, by the Italian Ministry for Research, and by some major process control companies.

Dr. Parisini is the present Chair of the IEEE Control Systems Society Conference Editorial Board. He was the Chair of the Technical Committee on Intelligent Control and he is an appointed member of the Board of Governors of the IEEE Control Systems Society. He is a Distinguished Lecturer of the IEEE Control Systems Society. He is the corecipient of the 2004 Outstanding Paper Award of the IEEE TRANSACTIONS ON NEURAL NETWORKS. Currently, he serves as an Associate Editor of the *Automatica* and the *International Journal of Control* and as a Subject Editor of the *International Journal of Robust and Nonlinear Control*. He served as an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL and the IEEE TRANSACTIONS ON NEURAL NETWORKS and as a Subject Editor of the *International Journal of Adaptive Control and Signal Processing*. He was a Guest Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS Special Issue on Adaptive Learning Systems in Communication Networks. He was involved in the organization and in the technical program committees of several IEEE CSS sponsored conferences including the IEEE Conference on Decision and Control and the IEEE Conference on Control Applications. In particular, he was the Vice-Program Chair of the 2003 IEEE Conference on Decision and Control, the Program Chair of the IEEE International Symposium on Intelligent Control, Mexico City, Mexico, 2001, and the Program Chair of the Joint IEEE International Symposium on Intelligent Control and Mediterranean Control Conference, Limassol, Cyprus, June 2005. He was recently appointed the Program Chair of the 47th IEEE Conference on Decision and Control to be held in Cancun, Mexico, in 2008.