Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008



# Moving-horizon State Estimation for Nonlinear Systems using Neural Networks

A. Alessandri, M. Baglietto, G. Battistelli, and R. Zoppoli

*Abstract*— **In recent results, a moving-horizon state estimation problem has been addressed for a class of nonlinear discrete-time systems with bounded noises acting on the system and measurement equations. For the resulting estimator, suboptimal solutions can be addressed for which a certain error is allowed in the minimization of the cost function. Building on such results, in this paper the use of nonlinear parameterized functions is studied to obtain suitable state estimators with guaranteed performance. Thanks to the off-line optimization of the parameters, the estimates can be generated on line almost instantly. A new technique based on the approximation of the cost value (and not of its argument) is proposed and the properties of such a scheme are studied. Simulation results are presented to show the effectiveness of the proposed approach in comparison with the extended Kalman filter.**

## I. Introduction

The idea of estimating the state of a system by a moving-horizon (MH) approach dates back to the sixties and was originally motivated by its intrinsic robustness, which makes the approach well-suited in the presence of modelling uncertainties and/or numerical errors [1]. Recently, researches have focused on the application of such techniques to linear systems [2], [3], [4], [5], hybrid systems [6], [7], and nonlinear systems [8], [9], [10], [11], [12], [13], [14]. In [8] an asymptotic state observer is described that results from the numerical solution of a sequence of nonlinear algebraic equations via the Newton's method. Similar optimization-based solution techniques are employed in [9], [10] to construct stable estimators for continuous-time dynamic systems. In [11], a MH observer for nonlinear continuous-time systems was proposed that performs estimation at discrete-time instants by approximately minimizing an integral error defined on the preceding time window. In [12], a MH estimation scheme was presented that allows one to explicitly take into account possible constraints on the system and requires the solution of a nonlinear programming problem at each time step. Moreover, a sufficient condition for the non-divergence of the estimation error in the presence of bounded noises was provided. All the aforementioned methods require that either the exact minimization of a nonlinear error cost function or the exact solution of a system of nonlinear equations is obtained on line. Such a requirement may lead to heavy calculations, thus reducing the applicability of such approaches. In order to overcome this drawback, a method was proposed in [13], where the possibility of committing a certain error in the minimization of the cost function is considered and the computation required to design the approximate filter may be carried out off line.

These results were furtherly extended in [14] were the simultaneous presence of both system and measurement noises was accounted for, the conditions that guarantee the stability of the estimation error were relaxed and the essentially local results of [13] were extended to regional stability. The stability analysis was based on quite a standard observability notion.

In this paper, we rely on the results in [14], with particular attention to the possibility of making a certain error in the minimization of the cost function. Such a result can be obtained by deriving off line an approximate state estimation function by means of suitable fixed-structure nonlinear approximators, in which a fixed number of parameters have to be optimized and that are particularly convenient in terms of structural complexity [15], [16]. With respect to [13] (where a similar technique was proposed), the approximate solution is referred to the value of the cost function and not to its argument, thus leading to less restrictive assumptions.

Various fixed-structure nonlinear approximators can be used, e.g., feedforward neural networks, radial basis functions, linear combinations of sinusoidal functions with variable frequencies, etc. In general, nonlinear approximators benefit from better approximation capabilities than those of traditional linear approximators (i.e., approximating functions made up of linear combination of fixed algebraic basis functions). However, how to choose, among various possible nonlinear approximators, a particular structure for solving a given functional optimization problem is a fundamental but still unsettled issue.

In this paper, we chose one-hidden-layer feedforward neural networks. Such a choice is motivated both by the nice properties of such class of approximators in solving approximately complex functional optimization problems (for a thorough discussion on this issue, see [17] and the references therein) and by the satisfactory experimental results obtained in solving highly nonlinear optimal control and estimation problems [18], [13], [19].

The paper is organized as follows. In Section II, the approximate MH state estimation algorithms is formulated and the results presented in [14] are summarized. In Section

A. Alessandri is with the Department of Production Engineering, Thermoenergetics, and Mathematical Models, DIPTEM–University of Genoa, P.le Kennedy Pad. D, 16129 Genova, Italy, alessandri@diptem.unige.it

M. Baglietto and R. Zoppoli are with the Department of Communications, Computer and System Sciences, DIST–University of Genoa, Via Opera Pia 13, 16145 Genova, Italy, {mbaglietto,rzop}@dist.unige.it

G. Battistelli is with the Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze, Via di S. Marta 3, 50139 Firenze, Italy, battistelli@dsi.unifi.it

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

III, nonlinear parameterized functions are introduced to solve off line the approximate estimation problem. A minimax method is presented for the determination of the optimal parameter vector characterizing such functions. Moreover, the approximating properties of the proposed technique are studied in connection with the use of a suitable projection operator. A simulation example is reported in Section IV.

## II. Problem statement and preliminary results

Let us consider a dynamic system described by the discrete-time equations

$$x_{t+1} = f(x_t, u_t) + \xi_t \,, \tag{1a}$$
$$y_t = h(x_t) + \eta_t \,, \tag{1b}$$

for $t = 0, 1, \ldots$, where $x_t \in \mathbb{R}^n$ is the state vector (the initial state $x_0$ is unknown) and $u_t \in \mathbb{R}^m$ is the control vector. The vector $\xi_t \in \mathbb{R}^n$ is an additive disturbance affecting the system dynamics. The state vector is observed through the measurement equation (1b) where $y_t \in \mathbb{R}^p$ is the observation vector and $\eta_t \in \mathbb{R}^p$ is a measurement noise vector. We assume the statistics of $x_0$, $\xi_t$, and $\eta_t$ to be unknown, and consider them as deterministic variables of unknown character that take their values from known compact sets.

We adopt the estimation scheme described in [14], which is bases on a MH strategy: at any time $t = N, N+1, \ldots$, the estimates $\hat{x}_{t-N,t}, \ldots, \hat{x}_{t,t}$ of the state vectors $x_{t-N}, \ldots, x_t$ are obtained on the basis of a *prediction* $\bar{x}_{t-N}$ of the state $x_{t-N}$ and of the *information vector*

$$I_t \stackrel{\triangle}{=} \text{col}\,(y_{t-N}, \ldots, y_t, u_{t-N}, \ldots, u_{t-1}) \tag{2}$$

where $N+1$ measurements and $N$ input vectors are collected within a "sliding window" $[t-N, t]$. The estimates $\hat{x}_{t-N+1,t}, \ldots, \hat{x}_{t,t}$ are generated by $\hat{x}_{t-N,t}$ through the noise-free dynamics, that is,

$$\hat{x}_{i+1,t} = f\left(\hat{x}_{i,t}, u_i\right) \,, \quad i = t-N, \ldots, t-1. \tag{3}$$

Hence it follows that at time $t$ only the estimate $\hat{x}_{t-N,t}$ has to be determined. The prediction $\bar{x}_{t-N}$ is obtained from the estimate $\hat{x}_{t-N-1,t-1}$ via the application of the function $f$, that is,

$$\bar{x}_{t-N} = f(\hat{x}_{t-N-1,t-1}, u_{t-N-1}), \quad t = N+1, N+2, \ldots .$$

The vector $\bar{x}_0$ denotes an a-priori prediction of $x_0$.

In the lines of [14],we make the following assumptions.

**A1.** The sets $\Xi$, $H$, and $U$ where $\xi_t$, $\eta_t$ and $u_t$ (respectively) take their values are compact sets, with $0 \in \Xi$ and $0 \in H$.

**A2.** The initial state $x_0$ and the control sequence $\{u_t\}$ are such that, for any possible sequence of disturbances $\{\xi_t\}$, the system trajectory $\{x_t\}$ lies in a compact set $X$.

Since, under assumption A2, at every time step $t = 0, 1, \ldots$, the state $x_t$ falls within the set $X$, the condition $\hat{x}_{t-N,t} \in X$ could be considered as a further constraint in our estimation scheme. In general, fulfilling such a constraint when applying a mathematical programming procedure results to be a very hard task. In order to mitigate this problem, a compact convex outer-approximation of $X$ will be considered and denoted by $\mathcal{X}$. Then, the constraint

$$\hat{x}_{t-N,t} \in \mathcal{X} \tag{4}$$

will be enforced. Of course, the fulfillment of such a constraint, together with assumption A3, ensures that the prediction $\bar{x}_{t-N}$ belongs to the compact set $\overline{\mathcal{X}} \stackrel{\triangle}{=} f(\mathcal{X}, U)$ for every $t = N, N+1, \ldots$ (the a-priori prediction $\bar{x}_0$ is chosen inside the set $\overline{\mathcal{X}}$). The following assumption is also needed:

**A3.** The functions $f$ and $h$ are $\mathcal{C}^2$ functions with respect to $x$ on $\mathcal{X}$ for every $u \in U$.

The sets $Y$ and $\mathcal{I}$ from which the vectors $y_t$ and $I_t$ take their values, respectively, can be expressed as

$$\begin{aligned} Y &\stackrel{\triangle}{=} \{y \in \mathbb{R}^p : \, y = h(x) + \eta, \, x \in X, \, \eta \in H\} \,, \\ \mathcal{I} &\stackrel{\triangle}{=} Y^{N+1} \times U^N \,. \end{aligned}$$

Summing up, the estimation scheme takes on the form

$$\hat{x}_{t-N,t} = a(\bar{x}_{t-N}, I_t), \; t = N, N+1, \ldots \tag{5a}$$
$$\bar{x}_{t-N+1} = f(\hat{x}_{t-N,t}, u_{t-N}), \; t = N, N+1, \ldots \tag{5b}$$
$$\bar{x}_0 \in \overline{\mathcal{X}} \,.$$

where $\hat{x}_{t-N,t} = a(\bar{x}_{t-N}, I_t)$ will be denoted as the *state-estimation function*.

Following a least-squares approach, in [14] the minimization of the following cost function[1] is addressed at any time $t = N, N+1, \ldots$:

$$J\left(\hat{x}_{t-N,t}, \bar{x}_{t-N}, I_t\right) = \mu \, \| \, \hat{x}_{t-N,t} - \bar{x}_{t-N} \, \|^2 + \sum_{i=t-N}^{t} \| \, y_i - h(\hat{x}_{i,t}) \, \|^2 \tag{6}$$

where $\mu$ is a positive scalar by which we express our belief in the prediction $\bar{x}_{t-N}$ with respect to the observation model.

In order to explicitly account for a possible error in the minimization of the cost (6), given a generic pair $(\bar{x}_{t-N}, I_t)$ with $\bar{x}_{t-N} \in \overline{\mathcal{X}}$ and $I_t \in \mathcal{I}$, let us denote by $J^\circ(\bar{x}_{t-N}, I_t)$ the cost associated with the exact minimization of cost (6), i.e.,

$$J^\circ(\bar{x}_{t-N}, I_t) \stackrel{\triangle}{=} \min_{\hat{x}_{t-N,t} \in \mathcal{X}} J(\hat{x}_{t-N,t}, \bar{x}_{t-N}, I_t) \,.$$

The following algorithm can be stated.

**Approximate MH estimator (Algorithm $\mathbf{E}^\varepsilon$).** *Given an a-priori prediction $\bar{x}_0^\varepsilon \in \overline{\mathcal{X}}$ and a positive scalar $\varepsilon$, at any time $t = N, N+1, \ldots$, find an estimate $\hat{x}_{t-N,t}^\varepsilon$ such that $\hat{x}_{t-N,t}^\varepsilon \in \mathcal{X}$ and*

$$J\left(\hat{x}_{t-N,t}^\varepsilon, \bar{x}_{t-N}^\varepsilon, I_t\right) - J^\circ\left(\bar{x}_{t-N}^\varepsilon, I_t\right) \le \varepsilon \,. \tag{7}$$

[1] Throughout the paper, for a generic vector $v$, we denote its Euclidean norm by $\|v\|$. For a generic matrix $M$, we shall denote its induced Euclidean norm by $\|M\|$.

*The prediction is propagated as*

$$\bar{x}^{\varepsilon}_{t-N+1} = f(\hat{x}^{\varepsilon}_{t-N,t}, u_{t-N}) \, . \tag{8}$$

□

In order to state the stability properties of Algorithm $\mathrm{E}^{\varepsilon}$, let us first define the function

$$F\left(x_{t-N}, u_{t-N}^{t-1}\right) \triangleq \begin{bmatrix} h\left(x_{t-N}\right) \\ h \circ f^{u_{t-N}}(x_{t-N}) \\ \vdots \\ h \circ f^{u_{t-1}} \circ \cdots \circ f^{u_{t-N}}(x_{t-N}) \end{bmatrix},$$

for $t = N, N+1, \ldots$, where $u_{t-N}^{t-1} \triangleq \mathrm{col}(u_{t-N}, \ldots, u_{t-1})$, "$\circ$" denotes function composition, and $f^{u_i}(x_i) \triangleq f\left(x_i, u_i\right)$.

The following assumption is made.

**A4.** System (1) is $\mathcal{X}$-observable in $N+1$ steps; i.e., there exists a $K$-function[2] $\varphi(\cdot)$, such that $\varphi\left(\|x_1 - x_2\|^2\right) \leq \left\|F\left(x_1, \bar{u}\right) - F\left(x_2, \bar{u}\right)\right\|^2, \forall\, x_1,\, x_2 \in \mathcal{X}, \forall\, \bar{u} \in U^N$.

Let us consider a sequence of approximate estimates $\hat{x}^{\varepsilon}_{t-N,t}\,, t = N, N+1, \ldots$ obtained by applying Algorithm $\mathrm{E}^{\varepsilon}$ for a certain $\varepsilon \geq 0$ and denote by $e^{\varepsilon}_{t-N} \triangleq x_{t-N} - \hat{x}^{\varepsilon}_{t-N,t}$ the related estimation error. Finally, let us denote by $k_f$ an upper bound on the Lipschitz constant of $f(\cdot, u)$ with respect to $x_t$ on $\mathcal{X}$ for every $u \in U$.

Then the following result can be stated (see [14] for the proof).

**Theorem 1:** *Suppose that assumptions A1, A2, A3, and A4 are satisfied. Moreover suppose that the $K$-function $\varphi$, defined in assumption A4, satisfies the following condition*

$$\delta \triangleq \inf_{x_1, x_2 \in \mathcal{X};\, x_1 \neq x_2} \frac{\varphi\left(\|x_1 - x_2\|^2\right)}{\|x_1 - x_2\|^2} > 0 \, . \tag{9}$$

*Then suitable scalars $\alpha, \bar{\beta}_0$, and $\bar{\beta}$ exist such that the square norm of the estimation error is bounded as*

$$\|e^{\varepsilon}_{t-N}\|^2 \leq \bar{\zeta}_{t-N}$$

*where $\{\bar{\zeta}_t\}$ is a sequence generated by*

$$\begin{aligned} &\bar{\zeta}_0 = \bar{\beta}_0 \\ &\bar{\zeta}_t = \alpha\, \bar{\zeta}_{t-1} + \bar{\beta}\,, \quad t = 1, 2, \ldots \, . \end{aligned} \tag{10}$$

*Moreover, if $\mu$ is selected such that*

$$\frac{8\, k_f^2\, \mu}{\mu + \delta} < 1\,, \tag{11}$$

*the bounding sequence $\{\bar{\zeta}_t\}$ has the following properties:*

*(a) the sequence $\{\bar{\zeta}_t\}$ converges exponentially to the asymptotic value $e^{\varepsilon}_{\infty}(\mu) \triangleq \bar{\beta}/(1-\alpha)$;*

*(b) if $\bar{\zeta}_t > e^{\varepsilon}_{\infty}(\mu)$ then $\bar{\zeta}_{t+1} < \bar{\zeta}_t\,, \quad t = 0, 1, \ldots$.* □

[2]Recall that a function $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$ is a $K$-function if it is continuous, strictly monotone increasing, and such that $\varphi(0) = 0$.

## III. Off-line design of an approximate MH estimator

In this section, a method is developed for the off–line solution of the minimizations involved in Algorithm $\mathrm{E}^{\varepsilon}$ that is based on the so-called *nonlinear approximators*. Such a method consists in constraining the estimation function to take on a fixed structure of the form

$$\hat{a}(\bar{x}_{t-N}, I_t, w),\; t = N, N+1, \ldots$$

where $w$ is a finite-dimensional vector of parameters to be optimized off line in order to ensure the fulfillment of condition (7) at any time instant. Once the parameters vector $w$ has been determined, the approximate estimation function $\hat{a}(\bar{x}_{t-N}, I_t, w)$ can be used on line to generate state estimates with a small computational effort.

As motivated in the introduction, among various fixed-structure nonlinear approximators that can be used, in this paper we chose one-hidden-layer feedforward neural networks. Clearly, the input variables of such a network are the components of the prediction $\bar{x}_{t-N}$ and of the information vector $I_t$. The output variables correspond to the components of the neural estimate $\tilde{x}_{t-N,t}$. This means that the $j$-th component $\tilde{x}^{(j)}_{t-N,t}$ of the output vector takes on the form

$$\begin{aligned} \tilde{x}^{(j)}_{t-N,t} &= \tilde{a}^{(j)}(\bar{x}_{t-N}, I_t, w) \\ &= \sum_{p=1}^{\nu} c_{pj}\, g\left(\bar{w}_p^{\top} \bar{x}_{t-N} + w_p^{\top} I_t + w_{0p}\right) + c_{0j}\,, \end{aligned}$$

where $\nu$ is the number of parameterized basis functions of the neural network, $g(\cdot)$ is a sigmoidal activation function, and the coefficients $c_{pj}\,, c_{0j}\,, \bar{w}_p\,, w_p$, and $w_{0p}$, for $p = 1, 2, \ldots, \nu$ and $j = 1, 2, \ldots, n$, are the components of the vector $w$ to be determined, i.e.,

$$\begin{aligned} w \triangleq \mathrm{col} \quad &(c_{pj}\,, c_{0j}\,, \bar{w}_p\,, w_p\,, w_{0p}\,; \\ &p = 1, 2, \ldots, \nu;\; j = 1, 2, \ldots, n) \, . \end{aligned}$$

It is important to note that the integer $\nu$ is sufficient to characterize the complexity of the network since the number of free parameters grows linearly with $\nu$.

Unfortunately, when the state estimates are generated by means of a nonlinear approximator, one cannot guarantee that such estimates belong to the set $\mathcal{X}$, where the trajectory of the system lies. In order to constrain the estimates inside the set $\mathcal{X}$, a possibility consists in projecting the outputs of the neural network onto the set $\mathcal{X}$. With this respect, by defining as $p_S(v)$ an operator that provides a projection of a vector $v$ onto a compact set $S$, i.e.,

$$p_S(v) \in \arg\min_{v' \in S} \|v - v'\|\,, \tag{12}$$

the approximate state estimate $\hat{x}_{t-N,t}$ can be obtained as

$$\hat{x}_{t-N,t} = \hat{a}(\bar{x}_{t-N}, I_t, w) = p_{\mathcal{X}}\left[\tilde{a}(\bar{x}_{t-N}, I_t, w)\right] . \tag{13}$$

In the following, the above mapping made up of the composition of the neural approximator with the projection operator will be called *neural state estimation function*. Furthermore, we shall always explicitly indicate the dependence of the structure of the approximate estimation function on the number $\nu$ of units in the hidden layer by means of a superscript. Then the approximate state estimator takes on the form

$$\begin{aligned}&\hat{x}_{t-N,t} = \hat{a}^{\nu}(\bar{x}_{t-N}, I_t, w), \quad t = N, N+1, \dots\\ &\bar{x}_{t-N+1} = f(\hat{x}_{t-N,t}, u_{t-N}), \quad t = N, N+1, \dots\\ &\bar{x}_0 \in \overline{\mathcal{X}}\,.\end{aligned}$$

Clearly, the latter is a suboptimal estimator and at each time $t = N, N+1, \dots$, the following error is made in the minimization of the cost

$$\begin{aligned}\varepsilon_{t-N} &\triangleq J(\hat{x}_{t-N}, \bar{x}_{t-N}, I_t) - J^{\circ}(\bar{x}_{t-N}, I_t)\\ &= J\left[\hat{a}^{\nu}(\bar{x}_{t-N}, I_t, w), \bar{x}_{t-N}, I_t)\right] - J^{\circ}(\bar{x}_{t-N}, I_t)\,.\end{aligned}$$

If one can guarantee, for a given $\varepsilon$, that

$$\varepsilon_{t-N} \le \varepsilon, \quad t = N, N+1, \dots\,, \tag{14}$$

then, at any time $t = N, N+1, \dots$, the estimate $\hat{x}_{t-N,t} = \hat{a}^{\nu}(\bar{x}_{t-N}, I_t, w)$ satisfies condition (7) in Algorithm $\mathrm{E}^{\varepsilon}$. Hence, as shown in the previous section (see Theorem 1), it is possible to determine an upper bound on the norm of the estimation error $e_{t-N} \triangleq x_{t-N} - \hat{x}_{t-N,t}$.

It is important to point out that, given the output of the neural networks $\tilde{x}_{t-N,t}$ the computation of the projection $p_{\mathcal{X}}(\tilde{x}_{t-N,t})$ requires, in general, the solution of a minimization problem (see (12)). If no a-priori assumption is made on the form of the compact set $\mathcal{X}$, this may lead to considerable complications. With few exceptions, a closed form for $p_{\mathcal{X}}(\cdot)$ is not available; then, in general, such a computation has to be performed on–line, stage after stage, as soon as the vector $\tilde{x}_{t-N,t}$ becomes available. This can be a serious drawback, since the calculation of the projection can be very computationally demanding. However, as the set $\mathcal{X}$ is convex, the computation of the projection $p_{\mathcal{X}}(\tilde{x}_{t-N,t})$ becomes a convex optimization problem for which many computationally efficient numerical solvers exist [20]. Moreover, in this case, the projection $p_{\mathcal{X}}(\tilde{x}_{t-N,t})$ is always unique and depends continuously on the vector $\tilde{x}_{t-N,t}$.

These considerations suggest to choose the outer approximation $\mathcal{X}$ of $X$ among some particular kinds of sets (e.g., polytopes or ellipsoids), the computation of $p_{\mathcal{X}}(\tilde{x}_{t-N,t})$ can be considerably simplified and speeded up [21], [22]. Note that, even when the set $X$ is convex, such an outer approximation may be useful in order to reduce the computational burden associated with the calculation of the projection.

The need for fulfilling condition (14) enforces us to use an approach to derive the optimal parameter vector that turns out to be quite different from most neural–network learning algorithms (e.g., backpropagation). In fact, such methods would not offer us any guarantee that the error due to the replacement of the optimal cost with the approximate cost associated with the neural state estimation function $\hat{a}^{\nu}(\bar{x}_{t-N}, I_t, w)$ can be uniformly bounded by a given positive scalar $\varepsilon$, or, more specifically, that inequality (14) can be verified. This leads us to determine a minimax neural estimator by solving the following problem.

**Problem M**. *Find a number $\nu^*$ of neural units and a parameter vector $w^*$ such that*

$$\max_{\bar{x}\in\overline{\mathcal{X}},\ I\in\mathcal{I}} \left\{ J\left[\hat{a}^{\nu^*}(\bar{x}, I, w^*), \bar{x}, I\right] - J^{\circ}(\bar{x}, I)\right\} \le \varepsilon\,. \tag{15}$$

□

Problem M can be addressed by solving a sequence of minimax problems. One can start with a small number $\nu$ of neural units and then increasing $\nu$ until inequality (15) is satisfied. In practice, we solve the related minimax problem for each $\nu$ and verify if the left–hand side of (15) is less than or equal to $\varepsilon$. In the case this condition does not hold, we increase the number $\nu$ and repeat the procedure, which ends when the number $\nu^*$ is obtained and the solution of the minimax problem yields the vector $w^*$, hence the corresponding neural minimax estimator.

A difficulty in solving our minimax problem depends on the fact that the function $J^{\circ}(\bar{x}_{t-N}, I_t)$ is unknown. However, given a couple $(\bar{x}_{t-N}, I_t)$, the scalar $J^{\circ}(\bar{x}_{t-N}, I_t)$ can be computed by solving a nonlinear programming problem for which cost (6) is minimized. Let us remark that, in the search of the optimal parameter vector, we cannot apply well-know methods like backpropagation, as such algorithms are derived from the minimization of $\mathcal{L}_2$ cost, since in our case the solution has to satisfy an $\mathcal{L}_\infty$ condition.

Once the estimation function $\hat{a}^{\nu}(\bar{x}_{t-N}, I_t, w)$ has been given the structure (13), the ability of this function to provide approximate estimates that satisfy condition (14) at any time instant becomes a crucial point of the method considered in the paper. In order to address this issue, the following assumption is now needed.

**A5.** Cost (6) has a unique global minimizer $\hat{x}^{\circ}_{t-N,t}$ for any $\bar{x}_{t-N} \in \overline{\mathcal{X}}$ and any $I_t \in \mathcal{I}$.

Assumption A5 together with the well-known density properties of one-hidden-layer feedforward neural networks (see, for example, [23], [24] and the references therein) allows one to state the following result.

**Theorem 2:** *Suppose that assumptions A1-A5 are verified. Then, for every $\varepsilon > 0$, there exist an integer $\nu$ and a parameter vector $w$, i.e., a neural approximation function $\hat{a}^{\nu}(\bar{x}_{t-N}, I_t, w)$ and hence a neural state estimation function $\tilde{a}^{\nu}(\bar{x}_{t-N}, I_t, w)$, such that*

$$\begin{aligned}&J\left[\hat{a}^{\nu}(\bar{x}_{t-N}, I_t, w), \bar{x}_{t-N}, I_t\right] - J^{\circ}(\bar{x}_{t-N}, I_t)\\ &= J\left\{p_{\mathcal{X}}\left[\tilde{a}^{\nu}(\bar{x}_{t-N}, I_t, w)\right], \bar{x}_{t-N}, I_t\right\} - J^{\circ}(\bar{x}_{t-N}, I_t) \le \varepsilon\end{aligned}$$

*for any $\bar{x}_{t-N} \in \overline{\mathcal{X}}$ and any $I_t \in \mathcal{I}$.*

*Proof.* Let us denote by $a^{\circ}(\bar{x}_{t-N}, I_t)$ the optimal estimation function that, given a couple $(\bar{x}_{t-N}, I_t)$ with $\bar{x}_{t-N} \in \overline{\mathcal{X}}$ and $I_t \in \mathcal{I}$ provides the unique (by assumption A5) minimum of the cost (6) over the set $\mathcal{X}$. Under assumptions A1-A3, cost (6) depends continuously on the estimate $\hat{x}_{t-N,t}$, on the prediction $\bar{x}_{t-N}$, and on the information vector $I_t$. Hence, recalling that the set $\mathcal{X}$ is compact, one may invoke the *Theorem of the Maximum* (see [25]) and conclude that the optimal estimate $a^\circ(\bar{x}_{t-N}, I_t)$ depends continuously on the prediction $\bar{x}_{t-N}$ and on the information vector $I_t$.

In addition, note that assumptions A1, A2, and A3 ensure the compactness of the sets $\overline{\mathcal{X}}$ and $\mathcal{I}$ to which the prediction $\bar{x}_{t-N}$ and the information vector $I_t$, respectively, belong. Then, by exploiting the density properties of one-hidden-layer feedforward neural networks (see [23], [24]), one may conclude that for every scalar $\vartheta > 0$, there exist an integer $\nu$ and a parameter vector $w$, [i.e., a neural approximating function $\tilde{a}^\nu(\bar{x}_{t-N}, I_t, w)$ ] such that

$$\|\tilde{a}^\nu(\bar{x}_{t-N}, I_t, w) - a^\circ(\bar{x}_{t-N}, I_t)\| \leq \vartheta \tag{16}$$

for any $\bar{x}_{t-N} \in \overline{\mathcal{X}}$ and any $I_t \in \mathcal{I}$.

Now note that, as the set $\mathcal{X}$ is convex, the function $p_{\mathcal{X}}(\tilde{x}_{t-N})$ is continuous over $\mathbb{R}^n$. Thus, the function $\tilde{J}(\tilde{x}_{t-N}, \bar{x}_{t-N}, I_t) \triangleq J[p_{\mathcal{X}}(\tilde{x}_{t-N}), \bar{x}_{t-N}, I_t]$ obtained as composition of continuous functions is also continuous over $\mathbb{R}^n \times \overline{\mathcal{X}} \times \mathcal{I}$. Moreover, the images of the compact set $\overline{\mathcal{X}} \times \mathcal{I}$ via the continuous functions $a^\circ(\bar{x}_{t-N}, I_t)$ and $\tilde{a}^\nu(\bar{x}_{t-N}, I_t, w)$ (for fixed values of $\nu$ and $w$) are compact. Hence, by using the *Heine-Cantor theorem*, it follows that the function $\tilde{J}(\tilde{x}_{t-N}, \bar{x}_{t-N}, I_t)$ is uniformly continuous. Then, for every $\varepsilon > 0$, there exists a scalar $\bar{\vartheta} > 0$ such that, if

$$\|\tilde{a}^\nu(\bar{x}_{t-N}, I_t, w) - a^\circ(\bar{x}_{t-N}, I_t)\| \leq \bar{\vartheta}$$

for any $\bar{x}_{t-N} \in \overline{\mathcal{X}}$ and any $I_t \in \mathcal{I}$, then

$$\Big| \tilde{J}[\tilde{a}^\nu(\bar{x}_{t-N}, I_t, w), \bar{x}_{t-N}, I_t] - \tilde{J}[a^\circ(\bar{x}_{t-N}, I_t), \bar{x}_{t-N}, I_t] \Big| \leq \varepsilon \tag{17}$$

for any $\bar{x}_{t-N} \in \overline{\mathcal{X}}$ and any $I_t \in \mathcal{I}$.

Furthermore, since the optimal estimate $a^\circ(\bar{x}_{t-N}, I_t)$ always belongs to the set $\mathcal{X}$, we have $p_{\mathcal{X}}[a^\circ(\bar{x}_{t-N}, I_t)] = a^\circ(\bar{x}_{t-N}, I_t)$ for any $\bar{x}_{t-N} \in \overline{\mathcal{X}}$ and any $I_t \in \mathcal{I}$, and so $\tilde{J}[a^\circ(\bar{x}_{t-N}, I_t), \bar{x}_{t-N}, I_t] = J^\circ(\bar{x}_{t-N}, I_t)$. As a consequence, inequality (17) can be rewritten as

$$\Big| J[\hat{a}^\nu(\bar{x}_{t-N}, I_t, w), \bar{x}_{t-N}, I_t] - J^\circ(\bar{x}_{t-N}, I_t) \Big| \leq \varepsilon\,. \tag{18}$$

The proof is completed by combining (16) with (18). □

Theorem 2 shows that the errors due to the introduction of the neural approximators and of the projection operators can be made arbitrarily small, provided that a sufficiently large number of neural units $\nu$ is used, thus ensuring the theoretical solvability of Problem M. The reader is referred to [17] for an extensive discussion on the capability of neural approximators to be characterized by a moderate number of parameters according to Barron's theorem [15].

## IV. NUMERICAL RESULTS

In this section, a simple example is considered to illustrate the performance of the proposed estimation approach.

Let us first consider the following discrete-time system

$$\begin{aligned} x_{t+1} &= A\, x_t + B\, u_t + \xi_t \\ y_t &= C\, x_t + \rho \sin\left[2\left(x_t^{(2)} - x_t^{(1)}\right)\right] + \eta_t \end{aligned} \tag{19}$$

where $x_t = [x_t^{(1)} \quad x_t^{(2)}]^\top \in \mathbb{R}^2$ and

$$A \triangleq \begin{bmatrix} 0.998 & 0.009 \\ -0.241 & -0.930 \end{bmatrix}, \quad B \triangleq \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix},$$
$$C \triangleq \begin{bmatrix} -100 & 100 \end{bmatrix}, \quad \rho \triangleq 40 \quad .$$

In the following, we shall consider $x_0$, $\xi_t$, and $\eta_t$ as uniformly distributed, independent random variables with $p(x_0) = \Pi\left(\left[1 \;\; 1\right]^\top, \left[\sigma_x^2 \;\; \sigma_x^2\right]^\top\right)$, $p(\xi_t) = \Pi\left(0, \left[\sigma_\xi^2 \;\; \sigma_\xi^2\right]^\top\right)$, and $p(\eta_t) = \Pi\left(0, \sigma_\eta^2\right)$, where $\Pi(m, v)$ represents the probability density function of a component-wise independent uniform distribution with mean value $m$ and covariance $\mathrm{diag}(v)$. The control $u_t$ was chosen equal to $u_t = 10^4 \sin(0.1\, t - \psi)$, where $\psi$ is a random variable uniformly distributed in $[0, 2\pi]$.

Let us consider the performance indices given by the Root Mean Square Error (RMSE) and the Asymptotic Root Mean Square Error (ARMSE). As additional performance index, we considered also the convergence time, i.e., the number of time instants that guarantees an RMSE smaller than $1.5$ times the ARMSE.

In the following of this section, for the sake of brevity, we shall denote the filter obtained by applying Algorithm $\mathrm{E}^\varepsilon$ where each approximate minimization is carried out by means of a mathematical programming tool (the Matlab Optimization Toolbox) as the On-line Moving Horizon Filter (OMHF) and the filter obtained solving Problem M as the Neural Moving Horizon Filter (NMHF). The NMHF was implemented by means of a one hidden layer feedforward neural network with 100 sigmoidal activation functions in the hidden layer. The values of $\mu$, $N$, and $\varepsilon$ were chosen equal to $10^5$, $5$, and $10^{-6}$, respectively.

In order to evaluate the effectiveness of the proposed MH estimation scheme, such filters were compared with the Extended Kalman Filter (EKF). Fig. 1 shows the behavior of the true values and the estimates of the two state components for a randomly chosen simulation. The influence of the

TABLE I

PERFORMANCE OF THE CONSIDERED FILTERS FOR $\sigma_x = 4 \cdot 10^3$, $\sigma_\eta = 10$, AND DIFFERENT $\sigma_\xi$ VALUES.

| | convergence time | | | ARMSE | | |
|---|---|---|---|---|---|---|
| $\sigma_\xi$ | OMHF | NMHF | EKF | OMHF | NMHF | EKF |
| 0.1 | 52 | 161 | 207 | 0.14 | 0.19 | 0.12 |
| 0.5 | 47 | 111 | 202 | 0.73 | 0.74 | 0.55 |
| 1 | 46 | 60 | 160 | 1.40 | 1.39 | 4.10 |
| 5 | 37 | 37 | 172 | 7.29 | 7.27 | 51.79 |
| 10 | 36 | 36 | 110 | 14.52 | 14.50 | 125.68 |
| 50 | 25 | 25 | 43 | 70.62 | 70.60 | 1284.40 |

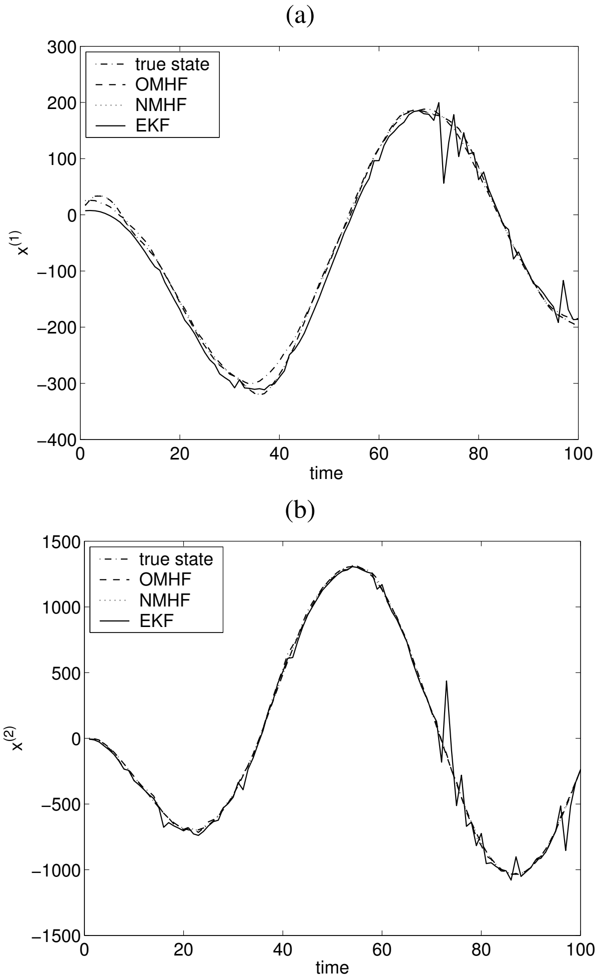


Fig. 1. True values and estimates of the first (a) and the second (b) component of the state for a randomly chosen simulation of system (19) with $\sigma_x = 40,\ \sigma_\xi = 20,\ \sigma_\eta = 10$.

system disturbance on the performance of the considered filters is illustrated in Table I. First, note that in most of the considered settings, the OMHF and the NMHF have an almost coincident behavior: only in the presence of low system noise, the NMHF shows a certain decay in performance with respect to the OMHF. This is not surprising as, in order to use the *same* neural network in all the simulations, the training has been performed in the least favorable conditions (i.e., in the presence of high system noise). However, it is important to point out once more that the OMHF requires heavy on-line computations, while the NMHF provides the estimates of the state almost instantly. As to the comparison with the EKF, while, on the one hand, in the presence of a low system disturbance, all the filters show a similar asymptotic behavior, on the other hand in the presence of a high system disturbance the asymptotic performance of the proposed filters turns out to be much better than those of the EKF. Furthermore, in all the considered frameworks, regardless of the value of $\sigma_\xi$, both the OMHF and the NMHF provide a better transient behavior than the EKF.

## References

[1] A. H. Jazwinski, "Limited memory optimal filtering," *IEEE Transactions on Automatic Control*, vol. 13, no. 5, pp. 558–563, 1968.

[2] C. V. Rao, J. B. Rawlings, and J. H. Lee, "Constrained linear estimation–a moving horizon approach," *Automatica*, vol. 37, pp. 1619–1628, 2001.

[3] A. Alessandri, M. Baglietto, and G. Battistelli, "Receding-horizon estimation for discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 473–478, 2003.

[4] ——, "On estimation error bounds for receding-horizon filters using quadratic boundedness," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1350–1355, 2004.

[5] ——, "Robust receding-horizon state estimation for uncertain discrete-time linear systems," *Systems & Control Letters*, vol. 54, no. 7, pp. 627–643, 2005.

[6] G. Ferrari-Trecate, D. Mignone, and M. Morari, "Moving-horizon estimation for hybrid systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 10, pp. 1663–1676, 2003.

[7] A. Alessandri, M. Baglietto, and G. Battistelli, "Receding-horizon estimation for switching discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 11, pp. 1736–1748, 2005.

[8] P. E. Moraal and J. W. Grizzle, "Observer design for nonlinear systems with discrete–time measurements," *IEEE Transactions on Automatic Control*, vol. 40, pp. 395–404, 1995.

[9] G. Zimmer, "State observation by on-line minimization," *International Journal of Control*, vol. 60, no. 4, pp. 595–606, 1994.

[10] M. Alamir, "Optimization based non-linear observers revisited," *International Journal of Control*, vol. 72, no. 13, pp. 1204–1217, 1999.

[11] H. Michalska and D. Q. Mayne, "Moving horizon observers and observer–based control," *IEEE Transactions on Automatic Control*, vol. 6, no. 6, pp. 995–1006, 1995.

[12] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, "Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations," *IEEE Transactions on Automatic Control*, vol. 48, no. 2, pp. 246–257, 2003.

[13] A. Alessandri, M. Baglietto, T. Parisini, and R. Zoppoli, "A neural state estimator with bounded errors for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 2028–2042, 1999.

[14] A. Alessandri, M. Baglietto, and G. Battistelli, "Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes," *Automatica*, vol. 44, no. 7, pp. 1753–1765, 2008.

[15] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Transactions on Information Theory*, vol. 39, pp. 930–945, 1993.

[16] F. Girosi and G. Anzellotti, "Rates of convergence for Radial Basis Functions and neural networks," in *Artificial Neural Networks for Speech and Vision*, M. R. J., Ed. Chapman & Hall, 1993, pp. 97–113.

[17] R. Zoppoli, M. Sanguineti, and T. Parisini, "Approximating networks and the extended Ritz method for the solution of functional optimization problems," *Journal of Optimization Theory and Applications*, vol. 112, no. 2, pp. 403–439, 2002.

[18] A. Alessandri, T. Parisini, and R. Zoppoli, "Neural approximators for finite–memory state estimation," *International Journal of Control*, vol. 67, no. 2, pp. 275–302, 1997.

[19] M. Baglietto, T. Parisini, and R. Zoppoli, "Distributed–information neural control: the case of dynamic routing in traffic networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 3, pp. 485–502, 2001.

[20] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2nd edition, Belmont, MA, 1999.

[21] T. Coleman and J. Liu, "An interior Newton method for quadratic programming," *Math. Program., Ser. A*, vol. 85, pp. 491–523, 1999.

[22] L. H. An, "An efficient algorithm for globally minimizing a quadratic function under convex quadratic constraints," *Math. Program., Ser. A*, vol. 87, pp. 401–426, 2000.

[23] M. S. K. Hornik and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[24] M. Leshno, V. Ya, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, pp. 861–867, 1993.

[25] C. Berge, *Espaces topologiques et fonctions multivoques*. Paris: Dunod, 1959.