

# A Study on Uncertainty–Complexity Tradeoffs for Dynamic Nonlinear Sensor Compensation

Michele Gubian, Anna Marconato, *Student Member, IEEE*, Andrea Boni, *Member, IEEE*, and Dario Petri, *Senior Member, IEEE*

**Abstract**—In this paper, we focus on the design of reduced-complexity sensor compensation modules based on learning-from-examples techniques. A multiobjective optimization design framework is proposed, where system complexity and compensation uncertainty are considered as two conflicting costs to be jointly minimized. In addition, suitable statistical techniques are applied to cope with the variability in the uncertainty estimation arising from the limited availability of data at design time. Numerical simulations are provided on a set of synthetic models to show the validity of the proposed methodology.

**Index Terms**—Multiobjective optimization (MOO), sensor compensation, support vector machines (SVMs).

## I. INTRODUCTION

GROWING numbers and types of sensors are employed in industrial plants, vehicles, and civil infrastructures, as well as in environmental and biomedical applications. Highly integrated single-package chipsets are equipped not only with sensing and conditioning elements but also with small computational units and possibly a wired or wireless communication unit. Such integration enables sensors to be organized in smart sensor systems, such as wireless sensor networks, that are able to accomplish higher level functionalities like self-diagnostics, adaptivity, and data fusion. Applications of this kind are characterized by stringent platform resource limitations; thus, a low-power, low-cost design strategy has to be adopted [1], [2].

One of the most critical issues that needs to be dealt with in the design of sensor networks is related to the intrinsic behavior of the sensor itself. In fact, the dynamic characteristic of (cheap) sensors can be altered by nonlinearities and memory distortions, whose effects can be reduced only by employing specific compensation strategies. Usually, sensor compensation techniques based on inverse modeling criteria are considered to reconstruct a given input signal on the basis of the corresponding output.

In particular, since sensor dynamics are in most cases nonlinear, and typically unknown, nonparametric techniques are

most suitably employed. Along this line, we propose the use of learning-from-examples algorithms, which allow one to model the behavior of the sensor by inferring a general rule based only on a set of input/output measurement data. Among them, the renowned support vector machines for regression (SVR) seem to be good candidates, both from a theoretical perspective and from on-field achievements [3], [4].

However, to produce a feasible design solution for low-power, low-cost sensor compensation, stringent constraints on memory size, central processing unit capability, and power consumption have to be met. In this context, the use of simple devices such as 8- or 16-bit microcontrollers as digital interfaces between the sensing unit and the transceiver is an almost mandatory solution. Thus, system complexity optimization becomes a major concern. The design of a compensation algorithm that performs as accurately as possible while keeping a low implementation cost is a problem that can be tackled in a multiobjective optimization (MOO) framework [5] once precise indexes for compensation accuracy and for complexity are defined.

In this paper, several synthetic data sets for sensor compensation will be used as a case study to introduce the MOO design framework [6], as well as the use of SVR for sensor compensation. The *mean square error* (MSE) will be used to evaluate compensation uncertainty, whereas a complexity index related to the number of required arithmetic operations will be defined based on typical microcontroller characteristics. Design space exploration is speeded up by employing the successful technique of genetic algorithms (GAs) [5]. In this framework, results based on numerical simulations will be shown, and insights on the inherent tradeoffs will be given.

As a second contribution of this paper, we will focus our attention on the estimation uncertainty that arises from the scarce availability of sensor data of which the desired values are known, which, from hereon, will be referred to as labeled data. Such data usually come from a calibration bank or from some other time-consuming measurement process and cannot be produced in large quantity at low cost. Hence, in a real design setting, MSE values are estimated on a small sequence of labeled data. This introduces a nonnegligible variability, which could mislead the whole design process if not taken into account. A statistical analysis of MSE estimation will be applied to the proposed case study, and as a result, an estimation procedure will be proposed that reaches a compromise between estimation accuracy and design space exploration search time.

This paper is organized as follows: Section II briefly accounts for the state-of-the-art sensor compensation. Section III outlines

Manuscript received September 21, 2007; revised July 3, 2008. First published September 23, 2008; current version published December 9, 2008. The Associate Editor coordinating the review process for this paper was Prof. Alessandro Ferrero.

M. Gubian was with the Department of Information and Communication Technology, University of Trento, 38100 Trento, Italy. He is now with Radboud University Nijmegen, 6500 Nijmegen, The Netherlands (e-mail: gubian@dit.unin.it).

A. Marconato, A. Boni, and D. Petri are with the Department of Information and Communication Technology, University of Trento, 38100 Trento, Italy (e-mail: amarcon@dit.unin.it; boni@dit.unin.it; petri@dit.unin.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/TIM.2008.2004985



Fig. 1. General scheme for sensor compensation based on digital signal processing.

the SVR-based sensor compensation approach. Section IV describes the MOO methodology. Section V illustrates simulation results. Section VI integrates the previous results with useful caveats and guidelines for MSE estimation. Finally, conclusions are drawn.

## II. SENSOR COMPENSATION

Sensor compensation refers to techniques aimed at reducing nonidealities in input/output characteristics of sensors. An acquisition system is usually made up of one or more transducers, which convert the measurand  $u(t)$  into a form suitable for analog-to-digital conversion, i.e.,  $y(t)$ . Ideally, the transduced quantity should have a linear relationship with the measurand, should be insensitive to the past history of the input signal, and should be insensitive to interfering environmental factors, as well as to offset, error gain, or other noise sources. Numerical techniques are currently applied for compensation, which means that after the transduced analog signal is sampled and quantized, a digital signal processing unit transforms the sensor output signal  $y(t)$  into a corrected value  $\hat{u}(n)$ , as shown in Fig. 1, where  $n$  is a discrete time index ( $t = nT_s$ ,  $T_s$  being the sampling period).

The easiest approach to sensor compensation is to employ a lookup table, but memory requirements scale badly with problem complexity. When no prior knowledge of the sensor dynamics is assumed, both linear and nonlinear techniques have been applied, the former performing well only when nonlinearities are negligible inside the sensor operating bandwidth [7]. Probably, neural networks (NNs) [8] are currently the most popular solution, usually in the form of multilayer feedforward NNs trained with backpropagation. An NN input can be just the signal of interest  $y(n)$  [9] or can also include measurements of a disturbance signal [10], [11].

Although in some cases an NN approach brings satisfactory results, they exhibit well-known disadvantages when compared to the more recent techniques like SVR (e.g., see [12]). First, the NN training is not guaranteed to converge to the global optimum, and the presence of many local optima requires repeating the training several times with different initial guesses, with a negative impact on the computational cost. Moreover, poor generalization capabilities are found when a small training set with a highly dimensional input is used [13]. In the next section, the use of SVR is proposed as a way to overcome such drawbacks.

## III. SVR-BASED SENSOR COMPENSATION

Since they were introduced by Vapnik in the mid-1990s, support vector machines (SVMs) have proven to be an effective tool for solving many different problems, including classification and regression tasks [3], [4]. As opposed to NNs, SVMs do not suffer from the problem of local optima since the training

problem is convex (see the following discussion) [4]. Moreover, they tend to yield sparse solutions, which is a feature of great interest for a reduced-complexity platform implementation. To our knowledge, SVMs have not been proposed for sensor compensation, whereas they are popular in the close field of soft sensing [12], [14]. Recently, the authors have proposed to exploit SVRs for the compensation of the nonlinear dynamic characteristics of sensors [15].

The principle of an SVR-based compensation algorithm can be summarized as follows: The task is to build an approximation function  $\hat{f}(\cdot)$  that is able to reconstruct the original input value  $u$  on the basis of the output values  $\mathbf{y}$ , given a training set  $Z_L = \{(\mathbf{y}_i, u_i)\}_{i=1}^L$  of  $L$  samples characterizing the input/output behavior of the sensor. In particular,  $u_i \in \mathbb{R}$ , and samples  $\mathbf{y}_i \in \mathbb{R}^D$  are vectors of  $D$  features  $y_i(n)$ ,  $y_i(n-1), \dots, y_i(n-D+1)$ , which represent past output values. The compensation function is obtained by solving a constrained quadratic optimization problem, which turns out to be convex. More specifically, we will refer to the  $\nu$ -SVR problem, introduced by Scholkopf and Smola [4], whose training problem is formulated as follows:

$$\max_{\alpha, \alpha^*} \left[ \sum_{i=1}^L (\alpha_i^* - \alpha_i) u_i - \frac{1}{2} \sum_{i,j=1}^L (\alpha_i^* - \alpha_i) (\alpha_j^* - \alpha_j) k(\mathbf{y}_i, \mathbf{y}_j) \right] \quad (1)$$

subject to

$$\begin{aligned} \sum_{i=1}^L (\alpha_i^* - \alpha_i) &= 0 \\ 0 &\leq \alpha_i, \alpha_i^* \leq \frac{C}{L} \\ \sum_{i=1}^L (\alpha_i^* + \alpha_i) &\leq C\nu \end{aligned} \quad (2)$$

where  $\alpha_i$  and  $\alpha_i^*$  are the optimization parameters,  $k(\cdot, \cdot)$  is a suitable kernel function, and  $C > 0$  and  $\nu > 0$  are further parameters to be set *a priori*. The resulting estimating function takes the following expression:

$$\hat{u} = \hat{f}(\mathbf{y}, \alpha, \alpha^*, q) = \sum_{i \in \text{SVs}} (\alpha_i^* - \alpha_i) k(\mathbf{y}_i, \mathbf{y}) + q \quad (3)$$

where “SVs” (support vectors) denotes the subset of training samples associated with nonzero coefficients  $\alpha_i, \alpha_i^*$ , whereas  $q$  is a bias term that is computed using the Karush–Kuhn–Tucker conditions (see [4] for further details).

To implement  $\nu$ -SVR for sensor compensation, training is performed offline, whereas only the computation of (3) has to be carried out onboard, noting that  $\mathbf{y}_i$  and  $\alpha_i$ , which are already computed in the training phase, are simply loaded in memory. The smaller the number of SVs and the number of

features  $D$ , the less the memory occupation. Similar considerations apply for the number of operations necessary to output a single  $\hat{u}$ , which, in turn, affects both power consumption and response time.

#### IV. MOO SETTING

In this section, the SVR sensor compensation problem is presented as a MOO design problem. The design space, as well as the design objectives, will be defined. The design space exploration methodology will also be specified.

As explained in Section III, the solution of the training problem (1) is expressed in a parametric form, where  $C$  and  $\nu$  have to be set *a priori*, both of them indirectly but strongly influencing the solution quality and its sparsity, i.e., the number of SVs [4]. However, no explicit relationship describes such influence; thus, a full search has to be performed. Similar considerations apply to the choice of the kernel function  $k(\cdot, \cdot)$  and its parameter(s). We decided to use a Gaussian kernel, i.e.,

$$k(\mathbf{u}, \mathbf{v}) = \exp(-\gamma \|\mathbf{u} - \mathbf{v}\|^2) \quad (4)$$

which depends on the parameter  $\gamma > 0$ . This and other radial basis function kernels are considered to be the most appropriate when no further knowledge of the problem is available, since all features are equally weighted [16]. Finally, the number of past output samples  $D$  that form the feature vectors  $\mathbf{y}_i$  has also to be determined. Then, given a training set  $Z_L$ , the design parameter space turns out to be defined as  $\mathcal{P} = (\nu, C, \gamma, D)$ .

Our ultimate purpose is to find a profitable tradeoff between two contrastive design goals, namely, a good signal reconstruction accuracy and a low hardware resource usage. The former can be rephrased as the minimization of the measurement uncertainty, which we chose to quantify with the reconstruction MSE.

The latter design goal may consist of more than a single expression to be minimized, e.g., one for memory occupancy, one for power consumption, and so on, but for the sake of simplicity, here, we will consider only one of them, namely, the number of arithmetic operations required to output a single reconstructed value  $\hat{u}$ , which directly affects power consumption and system response time. Some assumptions have to be made on the hardware platform, as well as on the numerical algorithms employed in the calculation of (3), to derive a formula that relates any specific design point  $(\nu, C, \gamma, D)$  to a corresponding computational burden.

The following are assumptions based on existing implementations [17]. First, (3), where the Gaussian kernel (4) is substituted for the generic kernel  $k(\cdot)$ , is rearranged as follows:

$$\hat{u} = \sum_{i \in \text{SVs}} \exp\left(\gamma \sum_{j=1}^D \left(y_i^{(j)} - y^{(j)}\right)^2 + \ln(\alpha_i^* - \alpha_i)\right) + q \quad (5)$$

where  $y_i^{(j)}$  and  $y^{(j)}$  are the  $j$ th components of the  $i$ th SV and of the current sample, respectively. Noting that the terms  $\ln(\alpha_i^* - \alpha_i)$  can be precalculated, a single output  $\hat{u}$  requires

$(2D + 1)N_{\text{SV}}$  additions,  $(D + 1)N_{\text{SV}}$  multiplications, and  $N_{\text{SV}}$  exponentials. We assume to have a microcontroller with  $b$ -bit fixed-point arithmetic and no hardware multiplier. The  $\exp(\cdot)$  operation will be implemented using the CORDIC approximation algorithm [18]. Under these assumptions, a multiplication requires  $b$  additions (taking the worst case) and exponential  $3 \cdot I + 1$  additions, where we fix the number of CORDIC iterations to  $I = b$ . Summing up, the cost of (5), which is expressed in number of additions, is

$$N_+ = N_{\text{SV}} \cdot ((b + 2)D + 4b + 2). \quad (6)$$

In the following simulations, we will assume  $b = 16$ .

The design objective space is then defined as  $\mathcal{O} = (\widehat{\text{MSE}}, N_+)$ . In general, it is not possible to find a single point in the design space  $\mathcal{P}$  that jointly minimizes all the objectives. The classic MOO paradigm consists instead of the search for a set of *nondominated* points in  $\mathcal{O}$ , collectively forming the so-called Pareto front, which is the set of solutions for which no objective function can be lowered unless some other function is increased [19]. Since a full search in  $\mathcal{P}$  is computationally prohibitive, GAs were used as a search engine, as already shown in [6] and [20].

#### V. NUMERICAL SIMULATIONS

In this section, we present a selection of numerical simulations that were performed to validate the MOO approach described in the previous sections. The purpose is to show that the proposed MOO framework is a general and versatile design tool for dynamic sensor compensation, allowing fine and direct control on uncertainty–complexity tradeoffs. To this aim, we tested diverse sensor models, signals, and noise conditions, the most relevant ones being illustrated in the following discussion. At this stage of the work, numerical simulations provide an effective way to widen the range of validation conditions, whereas experiments on real-world data sets may be adopted in future analyses of specific classes of sensors/problems.

##### A. Simulation Setting

The MOO design framework described in the previous sections has been tested on several synthetic sensor models that are both linear and nonlinear and excited with different input signals  $u(n)$ . Linear models of different orders, having moderately low-pass frequency responses, were analyzed. In particular, the following two recursive sensor models were considered:

$$y(n) = g(u(n)) + 0.4y(n-1) - 0.16y(n-2) \quad (7)$$

where

$$g(u(n)) = u(n) \quad (8)$$

in the linear case, and

$$g(u(n)) = u(n) + 0.2u^2(n) + 0.2u^3(n) \quad (9)$$

in the nonlinear case. The former system has two complex conjugate poles in  $0.4 \exp(\pm j(\pi/3))$ , whereas the latter is a

Hammerstein model whose dynamic block coincides with the former system. In addition, the following two nonrecursive sensor models were taken into account:

$$y(n) = g(u(n)) + 0.4u(n-1) + 0.16u(n-2) \quad (10)$$

with zeros in  $-0.4 \exp(\pm j(\pi/3))$  in the linear version (8) and (10), whereas this time, the nonlinear version (9) and (10) is not in the class of the Wiener–Hammerstein separable nonlinear systems. Although that class of systems is very often used for modeling nonlinearities in sensors [21] and other real-world systems [22], [23], by choosing a model out of this class, we want to emphasize the generality of SVR as a nonparametric compensation tool.

A white Gaussian noise term  $w(n)$  was also added to the output, considering a signal-to-noise ratio of both 30 and 40 dB with respect to  $u(n)$ . The input signal  $u(n)$  was chosen to be a white noise uniformly distributed in  $[-1, 1]$ . Training sets  $Z_L = \{(y(n), \dots, y(n-D+1), u(n))\}_{n=1}^L$  with  $L = 50, 100, 200$ , and 400 consecutive samples were considered.

To perform the  $\nu$ -SVR algorithm, we employed the LIBSVM tool [24], whereas we implemented NSGA-II (Elitist Nondominated Sorting Genetic Algorithm) as a GA search engine [5]. Genetic coding of search variables in  $\mathcal{P}$  and NSGA-II parameters were chosen as described in [17], with minor changes to the purpose of speeding up search (a 5-bit coding length for  $\nu$ ,  $C$ , and  $\gamma$ ; population size = 30).  $N_+$  was calculated from (6), whereas MSE was estimated on a holdout test set of  $T = 10\,000$  samples using the formula

$$\widehat{\text{MSE}}_T^{h-o} = \frac{1}{T} \sum_{n=1}^T (\hat{u}(n) - u(n))^2. \quad (11)$$

To achieve baseline results, we considered memoryless end-point linearization (EPL). The steady-state solution of (7) and (10) at the extreme values  $u = \pm 1$  was calculated in both linear (8) and nonlinear (9) cases, and a straight line passing through those points provided the compensation equation. Sequences of 100 000 samples were used to estimate the corresponding baseline MSE, i.e.,  $\text{MSE}_{\text{EPL}}$ . This was compared with the SVR MSE, i.e.,  $\text{MSE}_{\text{SVR}}$ , with an accuracy improvement (AI) index defined as

$$\text{AI} = 10 \log_{10} \frac{\text{MSE}_{\text{EPL}}}{\text{MSE}_{\text{SVR}}} \quad [\text{in decibels}]. \quad (12)$$

## B. Simulation Results

Fig. 2 shows a Pareto front for the nonlinear version (9) of model (7) with 40 dB of added noise and a training set of  $L = 200$  samples. Starting from the rightmost point, we can inspect the graceful MSE degradation as less complex solutions are met. This confirms the goodness of SVR in this task and displays the richness of information provided by the MOO approach. For example, a designer can see that passing from about 7000 to 3000 additions, the MSE increases from  $5.9 \cdot 10^{-5}$  to  $1.2 \cdot 10^{-4}$ , which is around 3 dB (see Table I). This last choice

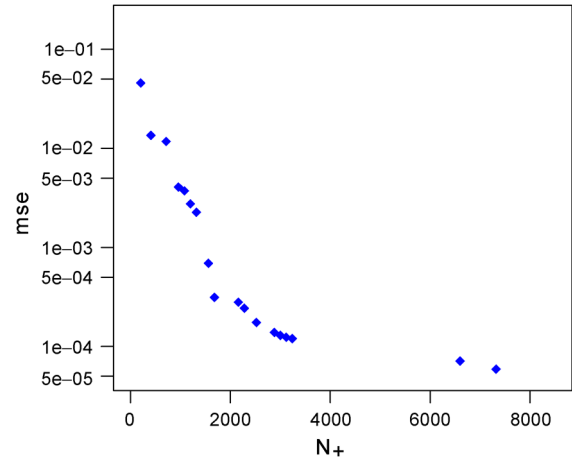


Fig. 2. Pareto front for the nonlinear model (7)–(9), 40 dB added noise,  $L = 200$ , MSE estimated with (11),  $T = 10\,000$ .

TABLE I  
SUMMARY FOR THE NONLINEAR MODEL (7)–(9): 40-dB NOISE. BASELINE  $\text{MSE}_{\text{EPL}} = 3.48 \cdot 10^{-1}$ ; LOWER BOUND  $\text{MSE}_{\text{LB}} = 3.33 \cdot 10^{-5}$

point	$L$	$D$	$\text{mse}_{\text{SVR}}$	AI [dB]	$N_{\text{SV}}$	$N_+$
min mse	50	3	$4.63 \cdot 10^{-4}$	28.8	18	2160
half $N_+$	50	3	$3.31 \cdot 10^{-3}$	20.2	9	1080
min mse	100	3	$1.06 \cdot 10^{-4}$	35.2	34	4080
half $N_+$	100	3	$2.93 \cdot 10^{-4}$	30.8	15	1800
min mse	200	3	$5.91 \cdot 10^{-5}$	37.7	61	7320
half $N_+$	200	3	$1.20 \cdot 10^{-4}$	34.6	27	3240
min mse	400	3	$4.20 \cdot 10^{-5}$	39.2	99	11880
half $N_+$	400	3	$7.34 \cdot 10^{-5}$	36.8	45	5400

TABLE II  
SUMMARY FOR THE NONLINEAR MODEL (9) AND (10): 40 dB NOISE. BASELINE  $\text{MSE}_{\text{EPL}} = 8.87 \cdot 10^{-2}$ ; LOWER BOUND  $\text{MSE}_{\text{LB}} = 3.33 \cdot 10^{-5}$

point	$L$	$D$	$\text{mse}_{\text{SVR}}$	AI [dB]	$N_{\text{SV}}$	$N_+$
min mse	50	4	$8.50 \cdot 10^{-4}$	20.2	25	3450
half $N_+$	50	2	$1.66 \cdot 10^{-3}$	17.3	13	1326
min mse	100	4	$4.11 \cdot 10^{-4}$	23.3	51	7038
half $N_+$	100	4	$6.94 \cdot 10^{-4}$	21.1	23	3174
min mse	200	5	$2.13 \cdot 10^{-4}$	26.2	71	11076
half $N_+$	200	4	$4.34 \cdot 10^{-4}$	23.1	36	4968
min mse	400	5	$1.48 \cdot 10^{-4}$	27.8	187	29172
half $N_+$	400	4	$3.52 \cdot 10^{-4}$	24.0	51	7038

can be considered an advantageous tradeoff, unless special design requirements are present.

The rest of this section is devoted to reporting the most relevant and general observations that we could extract from our simulations. Tables I and II report a selected set of simulations, where each pair of rows having the same value of  $L$  summarizes a single Pareto front (e.g., Fig. 2 is summarized by the fifth and sixth rows in Table I). Values in the “point” column have the following meaning: “min MSE” refers to the solution with minimum uncertainty, which is always the rightmost point on the front, whereas “half  $N_+$ ” refers to the point whose complexity  $N_+$  is closest but not greater than half the complexity of the “min MSE” point (e.g., in Fig. 2, it is the third from the right).

A first aspect to be analyzed is the capability of the SVR algorithm to accomplish the compensation task. In the case of a uniform white input  $u(n)$ , both linear and nonlinear sensor

models were compensated up to an uncertainty level that was often close to the lower bound given by the added noise  $w(n)$ . Such behavior was observed across repeated batches of simulations performed under the same conditions with different training sets, where the uncertainty lower bound was often reached only for a training set size of at least  $L = 400$ . More generally, uncertainty decreased with increasing  $L$ . Tables I and II refer to a 40-dB noise level, which corresponds to an MSE lower bound  $\text{MSE}_{\text{LB}} = 3.33 \cdot 10^{-5}$ , since  $u(n)$  has a power of 1/3. Non-recursive models (10) were usually harder to compensate than recursive models (7), and this has the following interpretation: To invert system (7), only three samples are enough, namely,  $y(n)$ ,  $y(n-1)$ , and  $y(n-2)$ . On the other hand, to invert (10), infinitely many samples in the past have to be considered. Thus, in this latter case, only an approximate inversion can be achieved. Tables I and II show this clearly in that  $D = 3$  everywhere in the former case, whereas variations are found in the latter case, with  $D$  increasing with the training set size  $L$ , i.e., when more information is available. To confirm the aforementioned interpretation, other nonrecursive models like (10) were tested, considering zeros at different positions in the linear case (8). The parameter  $D$  was observed to increase with decreasing distance from the unit circle.

Quite different results were found when feeding the model with a Gaussian white noise (not reported here). While Pareto optimal solutions showed the same behavior with respect to the number of past samples  $D$  used by the SVR, uncertainty levels were decidedly higher than the uniform case. A plausible explanation for this is found by noting that a Gaussian sample  $\{u(n)\}$  contains also a few values far from the mean; thus, the SVR algorithm is not able to generalize well in such situations. Having a representative training set is a general requirement for any learning-from-examples machine. In our problem, if only a Gaussian input is available (e.g., thermal noise), then data have to be preprocessed by removing values outside the range of interest.

A second aspect of great relevance in a resource-constrained design scenario is the way the complexity index  $N_+$  varies on the Pareto front, as well as with different values of  $L$ . Pareto fronts tend to be L-shaped, like in Fig. 2, which is a fact that is revealed by the usually moderate losses in accuracy, on the order of 3 dB, when passing from the “min MSE” to the “half  $N_+$ ” solution on the same front. When considering increasing training set sizes  $L$ , SVR often shows an increasing computational complexity trend, which is not always followed by a substantial decrease in uncertainty.

A final aspect concerns the capability of the NSGA-II algorithm to explore the design space  $\mathcal{P}$ . This was already shown in [6], [17], and [20] and was confirmed here. In particular, note that for the system (7)–(9), the values of  $D$  of the Pareto optimal solutions were automatically found by NSGA-II, and all of them coincided with the expected  $D = 3$ .

## VI. MSE UNCERTAINTY EVALUATION

While the complexity index  $N_+$  is, to some approximation, a deterministic number known after training,  $\widehat{\text{MSE}}$  is evaluated from the holdout estimator (11), whose statistical properties

depend on the holdout test set size  $T$ . In a real design setting, the availability of labeled data is limited due to cost reasons. Thus, we often must consider values of  $T$  much smaller than what we used in Section V. In this case, the holdout estimator (11) is known to exhibit poor performance in terms of high variability with respect to different realizations of the test set [17]. Thus, more sophisticated techniques have been devised, such as  $k$ -fold *cross validation* [25] and *bootstrap* estimators [26]. However, those techniques are computationally expensive because they require retraining SVR with the same parameters  $\mathcal{P}$  tens of times. Since this would proportionally increase the MOO search time, we decided to further investigate the possibility of using (11) as often as possible.

To this aim, we investigated the variability of estimates provided by (11) with respect to both test set realization and parameters  $\mathcal{P}$ . In fact, in a design space exploration, the test set will not vary, whereas variations will involve  $\mathcal{P}$ . A two-way analysis of variance was performed, where the variability in the parameter space  $\mathcal{P}$  and the variability of the test set were compared in the way they affect the MSE estimation error  $\epsilon$ , i.e.,

$$\epsilon = \widehat{\text{MSE}}_T^{h-o} - \text{MSE}_\infty \quad (13)$$

where the asymptotic value  $\text{MSE}_\infty$  was estimated using (11) with  $T = 10\,000$ , whereas  $T$  as small as 50 or 100 was considered in  $\widehat{\text{MSE}}_T^{h-o}$ . Results confirmed that variations with respect to the test set have a significant effect on  $\epsilon$  at the 95% confidence level, whereas this cannot be said for variations with respect to the design point  $\mathcal{P}$ . Then, we can expect that estimates obtained with (11) will not exhibit sensible variations in terms of  $\epsilon$  while using the same test set throughout the design space  $\mathcal{P}$ , but rather, they will tend to be uniformly shifted away from their respective asymptotic values  $\text{MSE}_\infty$ . This is an advantage in that a Pareto front, which is the result of global multidimensional ranking, is not going to be severely shuffled.

Exploiting such results, we decided to proceed in two steps: first, plug the holdout estimator (11) in the GA and perform a complete design space exploration, then recompute MSE only for the small set of solutions belonging to the Pareto front, this time using the bootstrap estimator. This way, we are using a more accurate but computationally expensive MSE estimator to correct a first “tentative” Pareto front obtained with a less reliable but computationally cheaper MSE estimator. Fig. 3 shows an example using the same setting as in Fig. 2 but now using  $T = 100$  for holdout estimates (diamond points). In the same figure, bootstrap estimates obtained with 100 bootstrap realizations (triangles) and asymptotic reference values (crosses) recomputed on the same points are also reported. Note that the Pareto front in Fig. 3, although not identical to the more accurate front of Fig. 2, has a quite similar behavior. In addition, bootstrap estimates are a bit better than the holdout estimates.

## VII. CONCLUSION AND FUTURE WORK

In this paper, a MOO framework for sensor compensation design has been introduced and analyzed. This framework ultimately enables explicit control of computational complexity at design time, which is an issue of paramount importance in

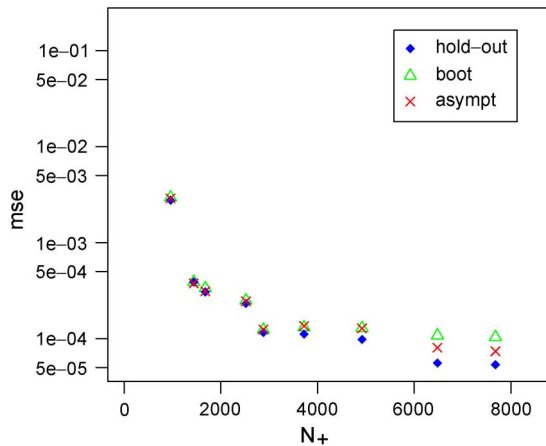


Fig. 3. Pareto front for the nonlinear model (7)–(9), 40 dB added noise,  $L = 200$ . MSE is estimated with (11),  $T = 100$  for holdout,  $T = 10\,000$  for asymptotic values, and bootstrap involved 100 iterations.

any reduced-complexity platform implementation. By using the SVR algorithm as a compensation engine, we showed how to tackle this design problem as a MOO problem, where design space exploration is driven toward the joint minimization of both measurement uncertainty and its own implementation complexity.

Results are encouraging, since not only SVRs could achieve good results in terms of signal reconstruction, but also, the MOO framework allowed us to quite effectively analyze uncertainty–complexity tradeoffs. Moreover, a statistical analysis provided the necessary assurance concerning the reliability of simple uncertainty estimators, even when only small data sets are employed, which corresponds to a realistic design setting.

Several future directions are open, and we will list just a couple of them here. The MOO framework could be extended to support more than two objectives. For example, real-time compensation could be dealt with by introducing three objectives, namely, uncertainty, execution time, and power consumption. SVMs showed good performance, but the MOO framework could be used as a valuable comparison tool for the state-of-the-art competing compensation techniques such as NNs. To our knowledge, none of the many empirical comparison of SVM versus NN has considered the point of view of uncertainty–complexity tradeoff, but rather, mainly uncertainty minimization is taken into account (e.g., see [27] and [28]).

## REFERENCES

- [1] A. Boni, F. Pianegiani, and D. Petri, “Low-power and low-cost implementation of SVMs for smart sensors,” *IEEE Trans. Instrum. Meas.*, vol. 56, no. 1, pp. 39–44, Feb. 2007.
- [2] F. Pianegiani, M. Hu, A. Boni, and D. Petri, “Energy-efficient signal classification in ad hoc wireless sensor networks,” *IEEE Trans. Instrum. Meas.*, vol. 57, no. 1, pp. 190–196, Jan. 2008.
- [3] V. Vapnik, *The Nature of Statistical Learning Theory*. Berlin, Germany: Springer-Verlag, 1995.
- [4] B. Schölkopf and A. Smola, *Learning With Kernels*. Cambridge, MA: MIT Press, 2002.
- [5] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.
- [6] A. Marconato, A. Boni, B. Caprile, and D. Petri, “Model selection for power efficient analysis of measurement data,” in *Proc. IEEE IMTC*, 2006, pp. 1524–1529.
- [7] M. Yamada and K. Watanabe, “A capacitive pressure sensor interface using oversampling  $\Delta$ – $\Sigma$  demodulation techniques,” *IEEE Trans. Instrum. Meas.*, vol. 46, no. 1, pp. 3–7, Feb. 1997.

- [8] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006.
- [9] D. Massicotte, S. Legendre, and A. Barwicz, “Neural-network-based method of calibration and measurand reconstruction for a high-pressure measuring system,” *IEEE Trans. Instrum. Meas.*, vol. 47, no. 2, pp. 362–370, Apr. 1998.
- [10] P. Arpaia, P. Daponte, D. Grimaldi, and L. Michaeli, “ANN-based error reduction for experimentally modeled sensors,” *IEEE Trans. Instrum. Meas.*, vol. 51, no. 1, pp. 23–30, Feb. 2002.
- [11] J. Patra, A. Kot, and G. Panda, “An intelligent pressure sensor using neural networks,” *IEEE Trans. Instrum. Meas.*, vol. 49, no. 4, pp. 829–834, Aug. 2000.
- [12] I.-S. Han, C. Han, and C.-B. Chung, “Melt index modeling with support vector machines, partial least squares, and artificial neural networks,” *J. Appl. Polym. Sci.*, vol. 95, no. 4, pp. 967–974, 2005.
- [13] C. Huang, L. S. Davis, and J. R. G. Townshend, “An assessment of support vector machines for land cover classification,” *Int. J. Remote Sens.*, vol. 23, no. 4, pp. 725–749, Feb. 2002.
- [14] W. Zhong, D. Pi, and Y. Sun, “SVM based soft sensor for antibiotic fermentation process,” in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2003, pp. 160–165.
- [15] A. Marconato, M. Hu, C. Marzadro, A. Boni, and D. Petri, “A resource-constrained sensor dynamic compensation using a learning-from-examples approach,” in *Proc. IEEE IMTC*, 2007, pp. 1–6.
- [16] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, “Choosing multiple parameters for support vector machines,” *Mach. Learn.*, vol. 46, no. 1–3, pp. 131–159, Jan. 2002.
- [17] M. Gubian, A. Marconato, A. Boni, and D. Petri, “Uncertainty–complexity trade-offs for sensor compensation design,” in *Proc. Int. Workshop AMUEM*, 2007, pp. 127–132.
- [18] R. Andraka, “A survey of CORDIC algorithms for FPGAs,” in *Proc. ACM/SIGDA 6th Int. Symp. Field Programm. Gate Arrays*, Monterey, CA, 1998.
- [19] V. Pareto, *Cours D’Economie Politique*. Lausanne, Switzerland: F. Rouge, 1896.
- [20] T. Suttorp and C. Igel, “Multi-objective optimization of support vector machines,” in *Studies in Computational Intelligence*, vol. 16. Berlin, Germany: Springer-Verlag, 2006, pp. 199–220.
- [21] M. Krasodomski, “Nonlinear measuring sensors influence on object identification quality,” in *Proc. IMTC*, Ottawa, ON, Canada, 2005, pp. 523–526.
- [22] J. R. S. J. Norquay and A. Palazoglu, “Application of Wiener model predictive control (WMPC) to a pH neutralization experiment,” *IEEE Trans. Control Syst. Technol.*, vol. 7, no. 4, pp. 437–445, Jul. 1999.
- [23] D. H. S. Prakriya, “Blind identification of LTI–ZMNL–LTI nonlinear channel models,” *IEEE Trans. Signal Process.*, vol. 43, no. 12, pp. 3007–3013, Dec. 1995.
- [24] *LIBSVM website*. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [25] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proc. IJCAI*, 1995, pp. 1137–1143.
- [26] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*. London, U.K.: Chapman & Hall, 1993.
- [27] D. Meyer, F. Leisch, and K. Hornik, “The support vector machine under test,” *Neurocomputing*, vol. 55, no. 1/2, pp. 169–186, Sep. 2003.
- [28] D. S. K. Karunasingha and S.-Y. Liong, “A comparison of support vector machines and artificial neural networks in hydrological/meteorological time series prediction,” *Adv. Geosci.*, vol. 6, pp. 91–96, 2006.



**Michele Gubian** received the M.Sc. degree in communication engineering from the Milan Polytechnic Institute, Milan, Italy, in 2004 and the Ph.D. degree in information and communication technology from the University of Trento, Trento, Italy, in 2008.

During 2003–2004, he was an Intern with the Advanced System Technology Laboratories, STMicroelectronics, Agrate Brianza, Italy, where he worked on a project in automatic speech recognition for embedded systems. He is currently a Postdoctoral

Fellow with Radboud University Nijmegen, Nijmegen, The Netherlands, where he is involved in the Sound2Sense Marie Curie Research Training Network, his task being the application of machine learning techniques to ASR systems. His main research interests include machine learning, statistical methods, multiobjective optimization, and language technology.



**Anna Marconato** (S'06) received the B.Sc. degree in mathematics and the M.Sc. degree in communication engineering in 2002 and 2005, respectively, from the University of Trento, Trento, Italy, where she is currently working toward the Ph.D. degree with the Department of Information and Communication Technology.

Her main research interests include support vector machines and multiobjective optimization in the context of smart embedded systems design.



**Andrea Boni** (M'05) was born in Genova, Italy, in 1969. He received the Ph.D. degree in electronic and computer science from the University of Genova in 2000.

He is currently an Assistant Professor with the Department of Information and Communication Technologies, University of Trento, Trento, Italy, where he teaches digital electronics with the School of Engineering. His main scientific research interests include digital signal theory and analysis, statistical signal processing, statistical learning theory, and

support vector machines and comprise the development of digital circuits for advanced information processing, particularly those implementable on programmable logic devices. The application fields of such interests focus on identification and control of nonlinear systems, pattern recognition, time-series forecasting, images and signals processing, etc.



**Dario Petri** (M'92–SM'06) received the Laurea (*summa cum laude*) and Ph.D. degrees in electronics engineering from the University of Padova, Padova, Italy, in 1986 and 1990, respectively.

From 1990 to 1992, he was an Assistant Professor with the Department of Electronics and Information Engineering, University of Padova. In 1992, he joined the University of Perugia, Perugia, Italy, as an Associate Professor, where he became the Chairperson of the undergraduate and graduate degree study programs in information engineering in 1999.

In the same year, he became a Full Professor of measurement and electronic instrumentation. In 2002, he joined the Department of Information Engineering and Computer Science, University of Trento, Trento, Italy, where he was the Chairperson of the International Ph.D. School of Information and Communication Technology from 2004 to 2007. He is currently the Chairperson of the undergraduate and graduate degree study programs in information engineering of the University of Trento. He is the author or coauthor of more than 150 papers published in international journals and conference proceedings. His research interests include measurement science and technology, particularly uncertainty evaluation methods, data-acquisition system design and testing, embedded system design and characterization, statistical inference methods, and the application of digital signal processing to measurement problems.

Prof. Petri has been the Chairman of the North Italy Chapter of the IEEE Instrumentation and Measurement Society since 2006. He is an Associate Editor of the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.