

# Approximating Optimal Estimation of Time Offset Synchronization with Temperature Variations

Maurizio Mongelli, *Member, IEEE*, Stefano Scanzio

**Abstract**—The paper addresses the problem of time offset synchronization in the presence of temperature variations, which lead to a non-Gaussian environment. In this context, regular Kalman filtering reveals to be suboptimal. A functional optimization approach is developed in order to approximate optimal estimation of the clock offset between master and slave. A numerical approximation is provided to this aim, based on regular neural network training. Other heuristics are provided as well, based on spline regression. An extensive performance evaluation highlights the benefits of the proposed techniques, which can be easily generalized to several clock synchronization protocols and operating environments.

**Index Terms**—Synchronization, Clocks, Control systems, Neural networks, Real time systems

## I. INTRODUCTION

**C**lock Synchronization Protocols (CSPs) are becoming a fundamental component of industrial networks as they are widely used in all the contexts in which a common time reference is required [1], [2]. The notion of time usually follows a clock register, which is cyclically updated with a frequency derived from a Crystal Oscillator (XO). Clock registers may differ in virtue of changes in XOs oscillation frequencies due to time-varying environmental conditions involving temperature, power supply, vibrations, humidity or pressure [3]. The purpose of a CSP is exchanging synchronization messages to keep the clock registers of different nodes aligned, by using timestamps on such messages. The synchronization error thus depends also on the rate at which the synchronization messages are exchanged and on their noisy content (timestamps *jitter*) due to unreliability of both communication and messages processing.

### A. Motivation

Synchronization algorithms are used to reduce the synchronization error by exploiting the information contained in the timestamps. At this purpose, several CSPs are based on Kalman filtering, in particular with respect to the IEEE 1588 protocol [4]–[7], which is the “de facto” standard in this context. Similar Kalman-based approaches have been also investigated for other CSPs, such as the Network Time Protocol (NTP) [8]. The fundamental hypotheses of Kalman estimation are: linear dynamics of the system, Gaussian noise components for both state and measurements variables and a-priori knowledge of the noise covariances.

This work was partially supported by the Ministry of Education, University, and Research of Italy (MIUR) in the framework of the Project Wi-Fact “Wireless FACTory and beyond” (DM53543).

### B. Contribution

XOs are usually modeled by linear equations [4], [5], but non-Gaussian noises lead to sub-optimal performance as stated in [4]. In this perspective, this paper formulates an optimal estimation approach beyond regular LQG hypotheses [6] (i.e., linear dynamics, quadratic cost function, Gaussian noises). The key idea is deriving the functional optimization formulation of optimal estimation and computing the inherent solution via neural approximation [9]. The main focus relies on the frequency variation component of the clock, whose noise is affected by temperature changes and other effects outside the control of the system designer, which are far from being Gaussian. Despite the approach presented in this paper does not guarantee optimal estimation, it outperforms Kalman filtering and infers the right synchronization correction over a large set of experimental conditions, without any ad-hoc adjustment of the algorithm’s parameters. The approach can be generalized to other non-LQG conditions, as the one of [10], in which non-linear clock models are claimed to be more accurate for Wireless Sensor Networks (WSNs). It must be noted, however, that the WSNs context deserves special attention also with respect to the energy efficiency of the algorithm, which this paper does not address explicitly.

The following section deals with the used CSP model, which is updated by an additional component concerning the XO exposed to temperature variations. In Section III, the concept of optimal filter is formulated. Kalman filtering is addressed by Section IV. The proposed approaches are: high-order regression splines and neural approximation of the optimal filter. They are addressed by Sections V and VI, respectively. The experimental setting is discussed in Section VII. Simulation results are provided in Section VIII. Section IX concludes the paper and summarizes possible topics of future research.

## II. CLOCK MODEL AND CORRECTION

The clock model considered here slightly differs from the one of [5]. At a generic time  $k$ , the clock registers of the master and the slave nodes (denoted by  $C_M(k)$  and  $C_S(k)$ , respectively) differ of the offset quantity  $\theta(k)$ .

$$C_S(k) = C_M(k) + \theta(k) \quad (1)$$

A typical two states clock model can be represented by the following equations in the discrete domain:

$$\theta(k) = \theta(k-1) + \gamma(k-1) \cdot \tau + \omega_\theta(k-1) \quad (2)$$

$$\gamma(k) = \gamma(k-1) + \omega_\gamma(k-1) + \omega_{\gamma T}(k-1, \cdot) \quad (3)$$

where  $\tau$  is the size of the time step and  $\gamma(k)$  represents the *skew* variations, as commonly represented in the scientific literature. The *skew* defines the normalized difference between the XO oscillation frequency and its nominal frequency. The  $\omega_\theta$  and  $\omega_\gamma$  quantities represent the noises affecting  $\theta$  and  $\gamma$ , respectively, whose distributions are of the Gaussian type; the inherent standard deviations are denoted by  $\sigma_{\omega_\theta}$  and  $\sigma_{\omega_\gamma}$ , respectively. The  $\omega_{\gamma T}$  quantity represents the noise effect due to temperature variations as detailed later.

CSPs without the compensation of the propagation delay, such as the *sender-receiver* FTSP [11] or the *receiver-receiver* RBS [12] and RBIS [13], estimate the offset  $\theta(k)$  and the skew  $\gamma(k)$  using two timestamps. Without any synchronization assistance, the simplest way to update the two quantities at the slave is:

$$\hat{\theta}(k) = \hat{T}_S(k) - \hat{T}_M(k) \quad (4)$$

$$\hat{\gamma}(k) = \frac{\hat{\theta}(k) - \hat{\theta}(k-1)}{\hat{T}_M(k) - \hat{T}_M(k-1)} \quad (5)$$

having the timestamps  $\hat{T}_M(k)$  and  $\hat{T}_S(k)$  acquired on a common time event  $k$  at master and slave. The timestamps are affected by jitters  $\omega_M$  and  $\omega_S$ , whose standard deviations are denoted by  $\sigma_M$  and  $\sigma_S$ , respectively:

$$\hat{T}_M(k) = C_M(k) + \omega_M(k) \quad (6)$$

$$\hat{T}_S(k) = C_S(k) + \omega_S(k) \quad (7)$$

The analysis made in [14] highlights the separate effects of both the frequency skew and the jitter. CSPs with the compensation of the propagation delay (such as the IEEE 1588 [15] and RTSP [10]) make use of 4 timestamps to estimate the propagation delay. In such a case, equations (4) and (5) are slightly different, see, e.g., [5]. The algorithms formulated in this work take (4) and (5) as a reference and are applicable to other CSPs (i.e., sender-receiver, receiver-receiver, with and without the estimation of the propagation delay), by updating (4) and (5) in agreement with the specific characteristics of the CSP<sup>1</sup>.

The distribution of jitters is supposed here Gaussian to focus on non-Gaussian temperature variations, but the hypothesis may be questionable, especially with software timestamping.

#### A. Temperature Variations

The model of the temperature variations takes as a reference a scenario in which an object changes different environments, characterized by different temperatures. A typical example deals with a Programmable Logic Controller (PLC) in a mobile carriage that moves inside and outside a warehouse. Other examples are embedded systems placed near a heat source, such as a car engine, or exposed to a transition between a state of sunshine and a state of shadow exposition. In these and other typical situations, the temperature variations can be fast, thus causing wide variations of the XO frequency.

Let two consecutive temperature changes have a round trip

<sup>1</sup>The three state equation model of [4] can be incorporated in the following formulations as well, without significant changes to the derivation of the proposed algorithms.

period  $p$ . When the object is placed in an environment with a different temperature, the temperature  $T$  of the XO varies with the following law<sup>2</sup>:

$$T = T_E + (T_{XO} - T_E) \cdot e^{-\frac{1}{tc}\Delta t} \quad (8)$$

where  $T_{XO}$  is the initial temperature,  $tc$  is the *thermal time constant* (i.e., the time needed to reach the 63.2% of  $T_E$ , in relation with the initial temperature  $T_{XO}$ ) and  $T_E$  is the temperature of the environment in which the XO is moved on. The left plot of Fig. 1 represents the temperature variation of a XO contained in a shell with  $tc = 60$  s; it is periodically moved every  $p = 600$  s between two environments with temperatures  $T_E^{High} = 40^\circ\text{C}$  and  $T_E^{Low} = -10^\circ\text{C}$ , respectively. Different types of XOs with different characteristics are available. Almost the totality of off-the-shelf PCs and embedded systems make use of AT-cut XOs to beat time, since they are very cheap. Unfortunately, such a kind of XOs highly suffer of oscillation frequency variations with respect to temperature. Temperature compensated XOs exist (VCXO, OCXO, TCXO, TVVCXO, OCVCXO), but usually they are not installed in common systems. For this reason, temperature variation is one of the most important causes of frequency instability on today devices [3]. The temperature-frequency relation for AT-cut XOs [16] can be accurately expressed with a 3rd polynomial form:

$$\frac{\Delta f}{f} = a \cdot (T - T_0) + b \cdot (T - T_0)^2 + c \cdot (T - T_0)^3 \quad (9)$$

where  $T_0$  is the reference temperature of the XO and  $a$ ,  $b$  and  $c$  three coefficients that represent the XO characteristic. The values used in this work are  $T_0 = 25^\circ\text{C}$ ,  $a = 0.0$ ,  $b = 0.4 \cdot 10^{-9}$  and  $c = 109.5 \cdot 10^{-12}$ , as reported in [16], and the temperature-frequency characteristic is reported in the middle plot of Fig. 1. From the time-temperature plot (left in Fig. 1) and the temperature-frequency function (9), the noise component due to temperature  $\omega_{\gamma T}(t, \cdot) = \omega_{\gamma T}(t, tc, p, T_E^{High}, T_E^{Low})$  can be eventually derived and quantized. The inherent probability distribution is not Gaussian; it is actually a multi-modal distribution, with significant asymmetry among the peaks.

### III. OPTIMAL FILTER

Let  $\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \boldsymbol{\xi}(k-1))$  be the state equation in compact form from (2) and (3) with  $\mathbf{x}(k) = [\theta(k), \gamma(k)]$ ,  $\boldsymbol{\xi}(k) = [\omega_\theta(k), \omega_\gamma(k) + \omega_{\gamma T}(k, \cdot)]$  and  $\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k), \boldsymbol{\eta}(k))$  be the measurement equation from (4) and (5) with  $\mathbf{y}(k) = [\hat{\theta}(k), \hat{\gamma}(k)]$ ,  $\boldsymbol{\eta}$  being the vector of measurement noises in (6)-(7) on  $\theta$  and  $\gamma$ , respectively. Both  $\mathbf{f}(\cdot)$  and  $\mathbf{g}(\cdot)$  are linear functions and can be expressed in terms of basic matrix algebra<sup>3</sup>. The optimal estimator (or optimal filter) is defined by the law  $\boldsymbol{\nu}_k^o(\cdot) = \boldsymbol{\nu}_k^o(\mathbf{I}_k)$  minimizing the following functional cost:

$$\boldsymbol{\nu}_k^o(\mathbf{I}_k) = \arg \min_{\boldsymbol{\nu}_k(\mathbf{I}_k)} \mathbb{E}_{\mathbf{x}(k)} \{ \mathbf{h}(\mathbf{x}(k) - \boldsymbol{\nu}_k(\mathbf{I}_k)) | \mathbf{I}_k \}, \forall \mathbf{I}_k \quad (10)$$

<sup>2</sup>The temperature variation of an object moved to an environment with a different temperature can be modelled by the so-called *Newton's law of cooling*. If the object has a negligible internal thermal resistance, the law is a precise approximation of its temperature evolution over time. The hypothesis of negligible thermal resistance may be reasonable for a XO.

<sup>3</sup> $\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \boldsymbol{\xi}(k-1)$ ,  $\mathbf{A} = [1, \tau; 0, 1]$ ;  $\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) + \boldsymbol{\eta}(k)$ ,  $\mathbf{H}$  being the identity matrix.

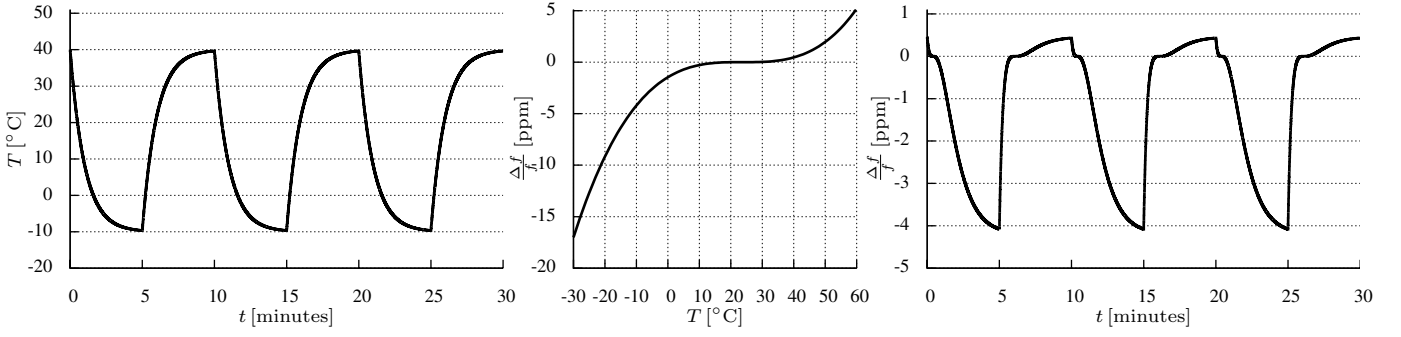


Fig. 1. Time-temperature variation, temperature-frequency characteristic, and time-frequency variation plots with  $p = 600$  s and  $t_c = 60$  s.

$\mathbf{I}_k$  being the information vector collecting all the measurements from the beginning  $\mathbf{I}_k = [\mathbf{y}(0), \dots, \mathbf{y}(k)]$  and  $\mathbf{h}(\cdot)$  being a Bayesian risk function<sup>4</sup>. The optimal filter thus implies the solution of a functional optimization problem. Formally, a new instance of problem (10) is stated at each  $k$ , thus leading to increasing sizes of the information vector. In practice, such a size is limited by looking at a fixed observation horizon of the past, namely:  $\mathbf{I}_k = [\mathbf{y}(k - K), \dots, \mathbf{y}(k)]$ . The size of the information vector,  $K$ , should be accurately set in order to find a good trade-off between limiting memory usage (which may imply some computational burden for the involved estimation algorithm) and acquiring sufficient knowledge for a reliable decision making.

#### A. Virtual clock

The explicit presence of control variables is disregarded when stating the optimal filter (10). The rationale of this choice relies on the fact that the master dynamics is independent of the decision taken at the slaves and slave regulation simply consists of correcting the offset component according to the estimation of  $\theta$  and the frequency skew component by using the estimation of  $\gamma$  [5]. In this perspective, the concept of *virtual clock* consists of the following: no correction (control variable) is applied to the slave clock register, but a mapping function (i.e., the virtual clock) drives the slave to infer the master notion of time, denoted in the following by  $C_M^v$ . In other words, a desired actuation may be scheduled at the slave with respect to the last update of  $C_M^v$ , and a timestamp on an external event may be acquired by a slave and converted in the master timescale using the most recent estimation of  $C_M^v$ . The virtual clock is updated each time new timestamps can be paired by a slave node. The impact of introducing the control variable in the functional cost, and derive the Riccati equations accordingly, may lead to a smoothing effect on the slave dynamics [6]. Such an effect is not investigated in this work.

### IV. KALMAN

The Kalman filter consists of deriving  $\nu^o(\cdot)$  in closed-form under the mentioned LQG hypotheses. The key issue

<sup>4</sup> $\mathbf{h}(\mathbf{z})$  is a Bayesian risk function if the following are met:  $\mathbf{h}(\mathbf{z})$  is not negative, it is symmetric, i.e.,  $\mathbf{h}(\mathbf{z}) = \mathbf{h}(-\mathbf{z})$  and it is not decreasing with increasing positive  $\mathbf{z}$ ; in the scalar case, examples are:  $h(z) = z^2$  and  $h(z) = |z|$ .

is to observe that the probability  $Pr(\mathbf{x}(k)|\mathbf{I}_k)$  consists of a multidimensional Gaussian function with mean  $\boldsymbol{\mu}_k$  and covariance  $\boldsymbol{\Sigma}_k$ ,  $\boldsymbol{\mu}_k$  being the estimation of  $\mathbf{x}(k)$  and  $\boldsymbol{\Sigma}_k$  being the variance of the estimator. Both  $\boldsymbol{\mu}_k$  and  $\boldsymbol{\Sigma}_k$  are computed by regular Kalman equations, whose variables depend on the matrices of  $\mathbf{f}(\cdot)$  and  $\mathbf{g}(\cdot)$  and on covariances of  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$ . In this paper, the Kalman equations of [5] are used without reporting them in detail for the sake of synthesis. Very similar equations are used in [6] as well.

#### A. Remarks

The Kalman filter guarantees the optimal solution of problem (10) (under the LQG hypotheses): at each  $k$ , the best estimation of  $\mathbf{x}(k)$  is  $\boldsymbol{\mu}_k$  and the components of  $\boldsymbol{\Sigma}_k$  decrease to zero with  $k \rightarrow \infty$ ; in other words, the uncertainty on  $\mathbf{x}(k)$ , expressed by the difference between  $\mathbf{x}(k)$  and  $\boldsymbol{\mu}_k$  under the  $\mathbf{h}(\cdot)$  function, tends to zero with  $k \rightarrow \infty$ . This holds in stationary conditions, namely, with fixed  $\mathbf{f}(\cdot)$ ,  $\mathbf{g}(\cdot)$ , and stationary noises  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$ . In non-stationary conditions, e.g., if some variance of  $\boldsymbol{\xi}$  or  $\boldsymbol{\eta}$  changes, the Kalman filter automatically adapts its calculations and achieves a new optimal solution after a transient period. At each discrete time instant, the Kalman estimation of  $\theta$  (denoted by  $\hat{\theta}$ ) leads to the virtual clock correction, as defined in III-A, directly through the application of  $\hat{\theta}$  in (1).

In passing, it is interesting to note that the Kalman optimality holds for every form of the function  $\mathbf{h}(\cdot)$  in (10), since the symmetries of both  $\mathbf{h}(\cdot)$  and  $Pr(\mathbf{x}(k)|\mathbf{I}_k)$  (when the latter is Gaussian) let the integration deriving from the  $\mathbf{E}(\cdot)$  operator in (10) achieve its global minimum exactly when  $\boldsymbol{\mu} \rightarrow \mathbf{x}$ . Outside LQG hypotheses, on the other hand, the  $Pr(\mathbf{x}(k)|\mathbf{I}_k)$  probability is not Gaussian anymore and its analytical computation, together with its application to the minimization of the above integral, become quite impractical, thus leading to the need of investigating numerical approximation mechanisms.

#### B. Perfect knowledge of state noises

The optimality of the Kalman filter is guaranteed with a perfect knowledge of  $\boldsymbol{\xi}$  and  $\boldsymbol{\eta}$  covariances. This may be questionable for the  $\boldsymbol{\xi}$  component due to the difficulty of acquiring a perfect knowledge of the XO characteristic curve. While the covariances of the timestamps can be estimated through experiments on the available device, a measurement process

of the state noises is a hard task because it involves the direct sampling of the XO physical properties. This is of particular importance with respect to the temperature variations. As the probability distribution of the noise component  $\omega_{\gamma T}(k, \cdot)$  is changing in dependence of the environment, one would want to derive an ‘autonomic’ estimation algorithm, capable to adapt to the current conditions (thermal time constant, temperature-frequency XO function (9), temperature levels) without manual intervention on the algorithm’s parameters in dependence of the environment. This is exactly the final goal of the neural approach.

### C. Kalman extensions

The well-known generalized versions of Kalman filtering, such as the ‘Extended’ and ‘Unscented’ homonymous filters try to follow non-linear dynamics under the Gaussian hypothesis [17]. For this reason, they are disregarded here. Other extensions of the Kalman filters, which have been disregarded for the involved computational burden, but which may be of interest for further performance comparison, are: the on line estimation of the covariances in parallel to the Kalman filter (e.g., [18]) and the Kalman Smoother [19], which may help reduce the estimation error through its post-processing of the Kalman filter data.

## V. REGRESSION SPLINES

Before addressing directly the numerical approximation of (10), simpler heuristics can be formulated. They consist of defining the two-dimensional vector  $\mathbf{q}_k = [\hat{T}_S(k), \hat{T}_M(k)]$  and a new information vector:

$$\mathbf{I}_k^s = [\mathbf{q}_{k-K}, \dots, \mathbf{q}_k]. \quad (11)$$

The  $i$ -th order spline (denoted by  $S_i$ ) is derived by interpolating, with the  $i$ -th order, the set of points in  $\mathbf{I}_k^s$  by means of the Ordinary Least Squares method [20]. Points  $\mathbf{q}_\kappa$ ,  $\kappa = k - K, \dots, k$ , lie around the 1 - th order spline ( $S_1$ ) representing the line of ideal correspondence between master and slave clocks. On the other hand, the distances of  $\mathbf{q}_\kappa$  from  $S_1$  denotes the quantity of the asynchronism between master and slave. The underlying idea of the splines is therefore to interpolate the trend of those points and to infer the position of the successive points  $[\hat{T}_S(\kappa), \hat{T}_M(\kappa)]$ ,  $\kappa > k$  as being generated by the spline itself. Being  $[\hat{T}_S(\kappa), \hat{T}_M(\kappa)]$  representative of  $[C_S(\kappa), C_M(\kappa)]$  according to (6) and (7), the  $i$ -th order spline becomes a possible virtual clock function ( $v_k^{S_i}(\cdot)$ ), as defined in III-A, by simply stating:  $C_M^v(t) = v_k^{S_i}(C_S(t))$ ,  $t > k$ . The process of spline interpolation is repeated every  $k$ :  $\mathbf{I}_k^s$  moves ahead of one time unit in order to update the calculation of the spline according to the last measurements ( $\hat{T}_S(k+1), \hat{T}_M(k+1)$ ) and by disregarding old measurements before  $k+1-K$ . The continuous updating helps follow a non-stationary behavior of the noises.

In the absence of measurement noise,  $\hat{T}_M(k) = C_M(k)$ ,  $\hat{T}_S(k) = C_S(k)$  and a perfect estimation is obtained from (4) and (5). In the presence of measurement noise, increasing the number of samples  $K$  mitigates the effect of oscillations around the spline due to timestamps jitters ( $\omega_M$  and  $\omega_S$  in

(6) and (7)), but, at the same time, the effect of frequency variations due to state noises makes the approximation less accurate. This happens in particular with large  $K$  and small orders  $i$ . The parameter  $K$  is thus a compromise between robustness to timestamps jitters and frequency variations.

With sudden frequency variations, spline orders higher than 1 may perform better because the elements of  $\mathbf{I}_k^s$  are not linear dependent. The first three order splines are therefore used (linear, quadratic and cubic functions); quadratic and cubic splines try to capture the non-linear behavior of the underlying trend in  $\mathbf{I}_k^s$ . On the other hand, higher order splines have been disregarded to avoid overfitting, to which the splines are more sensitive with more noise and large  $K$ .

Splines are intrinsically heuristics because they approximate the optimal filter implicitly, without addressing directly the solution of (10).

First order regression splines have been successfully used in [11]–[13]; at the best of the authors’ knowledge, this is the first time regression splines with order larger than 1 are used in this context.

An approach similar to linear regression has been applied by [8] to the NTP protocol. Splines surprisingly reveal to be more efficient than expected when the Gaussian hypothesis is not met [8].

## VI. NEURAL APPROXIMATION

A direct way to approximate the optimal filter is now addressed. The proposed method slightly differs from the ones of [21]–[24]. The following non-linear programming problem is defined:

$$\mathbf{w}^o = \arg \min_{\mathbf{w}} \mathbb{E}_{\mathbf{x}(k)} \{ \|\mathbf{x}(k) - \hat{\nu}_k(\mathbf{I}_k, \mathbf{w})\|^2 | \mathbf{I}_k \} \quad (12)$$

$\hat{\nu}_k(\mathbf{I}_k, \mathbf{w})$  being a neural network with input  $\mathbf{I}_k$ , with output an estimation of  $\mathbf{x}$  and with vector weights  $\mathbf{w}$ ; its input-output mapping depends also on the number and form of the internal basis function of each layer (e.g., sigmoidal or radial functions). A first approximation is derived by assuming the optimal filter  $\nu^o(\mathbf{I})$  being obtainable by searching in the space of the functions defined by the chosen structure of  $\hat{\nu}(\cdot, \mathbf{w})$ . Another approximation step comes from the observation that the  $\mathbb{E}(\cdot)$  operator above and its gradient cannot be obtained in closed-forms. As a result, regular non-linear programming procedures cannot be directly applied to (12). A countermeasure to this consists of resorting to a standard neural network training as follows. The new information vector is defined:

$$\mathbf{I}_k^{NN} = [d_{k-K}, \dots, d_k] \quad (13)$$

$d_k$  being the distance between each  $\mathbf{q}_k$  in  $\mathbf{I}_k^s$  as defined in (11) and the first order spline ( $S_1$ ) interpolating the set of points in  $\mathbf{I}_k^s$ . Then, a training set is stated with  $h = 1, \dots, H$  samples of  $\mathbf{I}_k^{NN}$  as in (13) (being used as input of the neural network in place of  $\mathbf{I}_k$ ), and of  $e_k = C_M(k) - v_k^{S_1}(k)$ , to be used as target of the neural network. Differently from (12), a scalar output

is stated for the approximated filter. The problem consists of finding the weights assignment  $\mathbf{w}^o$  so that:

$$\sum_{h=1}^H [{}^h e_k - \hat{\nu}_k({}^h \mathbf{I}_k^{NN}, \mathbf{w}^o)]^2 \leq \rho. \quad (14)$$

Problem (14) consists of tuning the output of the neural network (NN) in order to approximate the collected values of  $e_k$  as a function of the information vector  $\mathbf{I}_k^{NN}$ . The front end of the neural network thus becomes the synchronization errors introduced by the first order spline. Typical values for the bound  $\rho$  are in the range [0.001, 0.5]. After training, the neural network applies the virtual clock as:  $C_M^v(t) = v_k^{S1}(C_S(t)) + \hat{\nu}_k(\mathbf{I}_k^{NN}, \mathbf{w}^o)$ , where  $k$  is the most recent sample near time  $t$ . The underlying idea of the method is to follow the concept of approximating the trends in  $\mathbf{I}_k^s$ , as done by the splines, but with a more powerful inference capability than the splines. To this aim, the optimization problem (14) captures the mapping between the actual measurements and the synchronization error. During training, the NN should be capable to explore this mapping through the functional dependence on  $\mathbf{I}_k^{NN}$ , which should be more precise than simple interpolation of  $\mathbf{I}_k^s$ .

*Remark 1:* For the sake of simplicity, the target of the neural estimation focuses on the offset component ( $\theta$ ), without explicitly addressing the skew component ( $\gamma$ ). As evidenced by the following experiments, a significant performance improvement is achieved, despite this simplification. The results also show how the estimation of  $\gamma$  may be more crucial to limit the synchronization error between two consecutive synchronization instants with large  $\tau$ .

*Remark 2:* The rationale of the adoption of  $\mathbf{I}^{NN}$  (instead of the one of the optimal filter  $\mathbf{I}$ ) relies on the need of deriving neural inputs that do not depend on the absolute value of time. The values collected in  $\mathbf{I}$  formally depend on the initial instant of time from which the dynamical system is considered; in other words, they depend on the absolute values of the steps  $\{0 \cdot \tau, 1 \cdot \tau, 2 \cdot \tau, \dots\}$ . The values of  $\mathbf{I}^{NN}$ , on the other hand, are relative to the slave time shift through the measured distances from the 1-th order spline.

*Remark 3:* Both in [21], [22], [24] and in the present paper, a neural approach is applied to approximate optimal control or state estimation (optimal control in [21], [24], state estimation in [22] and in this paper). The main difference between those works and the idea presented here relies on the numerical approach used to derive the approximation. In [21], [22], [24], the functional cost provides indication in the direction of the optimal solution without the explicit knowledge of it. The minimization process is therefore driven by sampling the cost and its gradient and performing a descent step (also known as stochastic gradient [25]). Here, the offline sampling of the optimal solution (i.e., the exact master time) is available. This allows building a training database in which the information vector is mapped onto the difference between master and slave times. A similar approach has been used in [23].

*Remark 4:* The sequence of approximation steps from (10) to (14) leads to the conclusion that evaluating the error computed in approximating the unknown optimal estimation

law,  $\nu^o(\cdot)$ , with the actual one,  $\hat{\nu}^o(\cdot)$ , or, in other words, the convergence of the performance of  $\hat{\nu}^o(\cdot)$  to the one of  $\nu^o(\cdot)$ , is a hard task, even though some theoretical results assure the sub-optimal properties of these kinds of approximation schemes. The interested reader is referred to section V of [24] for an overview or to [9] for details on this subject.

Despite all the envisaged approximations, the obtained results guarantee a good level of suboptimality and give rise to several insights into the structure of the problem.

*Remark 5:* Another difficult task arises from finding good generalization capabilities of  $\hat{\nu}(\cdot)$  with respect to non-stationarity conditions, as previously mentioned for the Kalman filter. In this respect, the training is developed with respect to samples coming from non-stationary probability distributions. This consequently leads to the adaptation of  $\hat{\nu}(\cdot)$  to variable system conditions. An example of this is available in [21], in which a similar neural law approximates suboptimal controls in the presence of time-varying conditions. As outlined in subsection IV-B, Kalman does not guarantee this property.

## VII. EXPERIMENTS SETTING

### A. State parameters

The stability of a XO highly influences the performance of a CSP. The Allan variance [26] is the typical parameter used to characterize XOs stability. It depends on the sampling period  $\tau$  and can be calculated as  $\sigma_y^2(\tau) = \frac{1}{2\tau^2} \langle x_{n-2} - 2x_{n+1} + x_n \rangle$ , where  $n$  identifies a specific sample  $x_n$  of a temporally ordered sequence of timestamps spaced by an interval  $\tau$ . The variances ( $\sigma_\theta^2$  and  $\sigma_\gamma^2$ ) of the Gaussian noise components ( $\omega_\theta$  and  $\omega_\gamma$ ) of the clock model of equations (2) and (3) have been set in agreement to the typical values of Allan variance of AT-cut XOs. The first clock model considered models a XO that is not subjected to temperature changes. Other three clock models have been introduced in order to take into account temperature variations.

- *Gaussian* ('G'):  $\sigma_\theta^2 = 10^{-17} \text{ s}^2$  and  $\sigma_\gamma^2 = 10^{-19}$ . The thermal skew component is not preset ( $\omega_{\gamma T}(\cdot) = 0$ ). The G condition models the instabilities of a XO in a temperature controlled environment. This model takes into account all the typical XOs frequency instabilities (power supply, vibrations, humidity and pressure variations), with the exception of temperature effects. Since  $\omega_{\gamma T}(\cdot) = 0$ , equations (2) and (3) are equal to those reported in [5]. The values of  $\sigma_\theta^2$  and  $\sigma_\gamma^2$  are typical of an AT-cut XO of medium stability and they have been calibrated in order to obtain a slightly bigger Allan variance than the one reported in [27], [28].
- *Temperature* ('T'): the frequency skew due to temperature ( $\omega_{\gamma T}(\cdot) \neq 0$ ) is the only effect on quartz stability ( $\sigma_\theta^2 = 0$  and  $\sigma_\gamma^2 = 0$ ). Although unrealistic, this model is useful to capture the impact of temperature variations without any other source of state noise.
- *Gaussian+Temperature* ('G+T'): it is a realistic condition, in which random noise components of equations (2) and (3) ( $\sigma_\theta^2 = 10^{-17} \text{ s}^2$  and  $\sigma_\gamma^2 = 10^{-19}$ ) are taken into account together with the thermal skew component

TABLE I  
SIMULATION PARAMETERS AND ALLAN VARIANCES FOR THE ANALYZED  
CLOCK MODELS

Scenario	$\sigma_\theta^2$ [s <sup>2</sup> ]	$\sigma_\gamma^2$	$\sigma_{\gamma T}^2(\cdot)$	$\sigma_y^2(0.1\text{ s})$	$\sigma_y^2(1.0\text{ s})$
$G$	$10^{-17}$	$10^{-19}$	$= 0$	$9.97 \cdot 10^{-17}$	$1.00 \cdot 10^{-17}$
$T$	0	0	$\neq 0$	$6.24 \cdot 10^{-18}$	$6.37 \cdot 10^{-16}$
$G+T$	$10^{-17}$	$10^{-19}$	$\neq 0$	$1.06 \cdot 10^{-16}$	$6.47 \cdot 10^{-16}$

( $\omega_{\gamma T}(\cdot) \neq 0$ ). The ‘ $G+T$ ’ condition models a XO that is affected by the typical instabilities of a quartz, and it is cyclically exposed to temperature excursions.

### B. Other parameters

1) *Step size*: The value of the step size  $\tau$  represents the sending period of synchronization messages (i.e., common events used by the CSP to evaluate  $\hat{T}_M(k)$  and/or  $\hat{T}_S(k)$ ), as well as the time step of the clock model of equations (2) and (3). In practice, synchronization messages are sent periodically every 0.1 – 1 s and timestamps are frequently acquired in software since devices supporting hardware timestamping are commonly used only for time critical applications, such as the automation of high-voltage substations [29]. The use of software timestamp also in this context is becoming an important research challenge [30]. Simulation results are provided for  $\tau = 0.1\text{ s}$  and  $\tau = 1\text{ s}$ . Such values are consistent with CSP typical values: 1 or 2 s for *Sync* messages in IEEE 1588 and for the majority of CSPs and 100 ms for RBIS. The values of Allan variances for the two analyzed  $\tau$  and for the three experimental conditions are reported in Table I.

2) *Measurement noises*: The measurement jitters ( $\omega_M$  and  $\omega_S$  in (6) and (7)) are both fixed to a standard deviation of  $1\text{ }\mu\text{s}$ , in coherence with the typical accuracy of software timestamps obtained in the *interrupt handler* of the *device driver* of today’s real PCs [31]. Experiments performed by the authors on real wireless adapters confirm these results. The maximum uncertainty introduced by the timestamp mechanism could be even in the order of  $10\text{ }\mu\text{s}$  when timestamps are acquired at the communication *socket* level and, in case of high computational load, it can raise up to  $224\text{ }\mu\text{s}$  [32].

3) *Neural network*: The neural network used in the experiments (denoted in the pictures with the ‘NN’ acronym) has  $K$  inputs, 10 hyperbolic tangent hidden units and one linear output. The input samples  $\mathbf{I}_k^{NN}$  and the target  $e_k$  were normalized by dividing them with the maximum of the absolute value computed over all the input samples and targets of a single train database. The normalization constant computed for each train database is used for the normalization of test samples. It is trained using the standard back-propagation algorithm in 8 epochs with a *learning rate* linearly decreased during the train from 0.001 to 0.00001 and with a *momentum* of 0.01.

4) *Kalman*: Differently from the neural network, which is asked to provide estimations over a large set of environments without ad-hoc re-training, the Kalman filter is constantly updated with the right covariances of the state and measurement noises. This is needed each time a new configuration of the temperature noise parameters takes place.

When using Kalman, the first 5000 samples of each experiment are discarded. This ensures the performance evaluation after the convergence of the filter’s parameters. The two approaches are therefore compared unfairly. This helps highlight the robustness of the neural network as any other Kalman filter, which performs on line estimation of the covariances, may hardly guarantee better performance than the one shown here.

5) *Performance target*: All train and test databases consist of 50000 samples. The 99.9 percentile of the synchronization errors is the performance metric. It is denoted by  $p99.9$  and represents the 99.9 percentile of the difference between the real master time  $C_M(k)$  and the master time estimated at the slave by using the virtual clock  $C_M^v(k)$ , in correspondence with the arrival of the synchronization messages. The rationale of this choice relies on the need of capturing the capability of the algorithms to limit master-slave asynchronism below a given threshold, in the 99.9% of the cases. The qualitative behavior arising from the following results holds for other metrics as well, such as the variance of the synchronization error. The percentile is represented as a function of the size of the information vector  $K$ . The ranges of  $K$  considered in all of the following figures capture the best performance of each algorithm. The experiment related to the ‘ $G+T$ ’ scenario was repeated twice. For the second repetition, denoted as ‘ $G+T+\tau$ ’, the 99.9 percentile is evaluated after  $\tau$  seconds the synchronization time instants. The  $\tau$  delay represents the maximum time between two synchronization messages (under the hypothesis of an ideal communication network without loss of messages) and the point with higher likelihood of the maximum error. In this respect, ‘ $G+T+\tau$ ’ is the worst case scenario. The estimation of the offset at time  $k + \tau$  is obtained assuming a linear behavior of the XO after the last synchronization instant, i.e., by using the last estimations of  $\hat{\theta}(k)$  and  $\hat{\gamma}(k)$  to compute  $\hat{\theta}(k + \tau) = \hat{\theta}(k) + \hat{\gamma}(k) \cdot \tau$ . For the Kalman filter  $\hat{\theta}(k)$  and  $\hat{\gamma}(k)$  are the outputs of the filter, for the splines  $\hat{\theta}(k + \tau) = v_k^{Si}(k + \tau)$  and for the neural approximation  $\hat{\gamma}(k) = m - 1$ , where  $m$  is the slope of the 1 – th order spline<sup>5</sup>.

## VIII. PERFORMANCE EVALUATION

The results shown here are obtained with the following initial conditions:  $\theta(0) \in [-5, 5]\text{ s}$  and  $\gamma(0) = 0$ ; similar results are obtained with  $\gamma(0) = 10$  and 100 ppm. The left column of Fig. 2 represents the performance under the first Gaussian (‘ $G$ ’) scenario. Configuration parameters of the  $\omega_{\gamma T}(\cdot)$  function are  $tc = 60\text{ s}$ ,  $p = 600\text{ s}$ ,  $T_E^{Low} = -10^\circ\text{C}$  and  $T_E^{High} = 40^\circ\text{C}$ . As expected, Kalman achieves the best performance independently to the setting of  $K$  and  $\tau$ . Such a performance is also reached for  $\tau = 0.1\text{ s}$  by NN and  $S_1$  for  $K > 600$  and by  $S_2$  and  $S_3$  for larger  $K$  (1500 and 2000 with  $S_2$  and  $S_3$ , respectively). Similar comments are applicable to the  $\tau = 1.0\text{ s}$  case, a part from  $S_1$ , which guarantees the lowest percentile only at  $K = 260$ . As mentioned in Section

<sup>5</sup>Although the proposed techniques are focused on the estimation of  $\theta$ , we need for the ‘ $G+T+\tau$ ’ scenario the  $\gamma$  estimation which is derived from  $S_1$  when using the NN. The further extension of the NN for  $\gamma$  estimation is an ongoing topic of research.

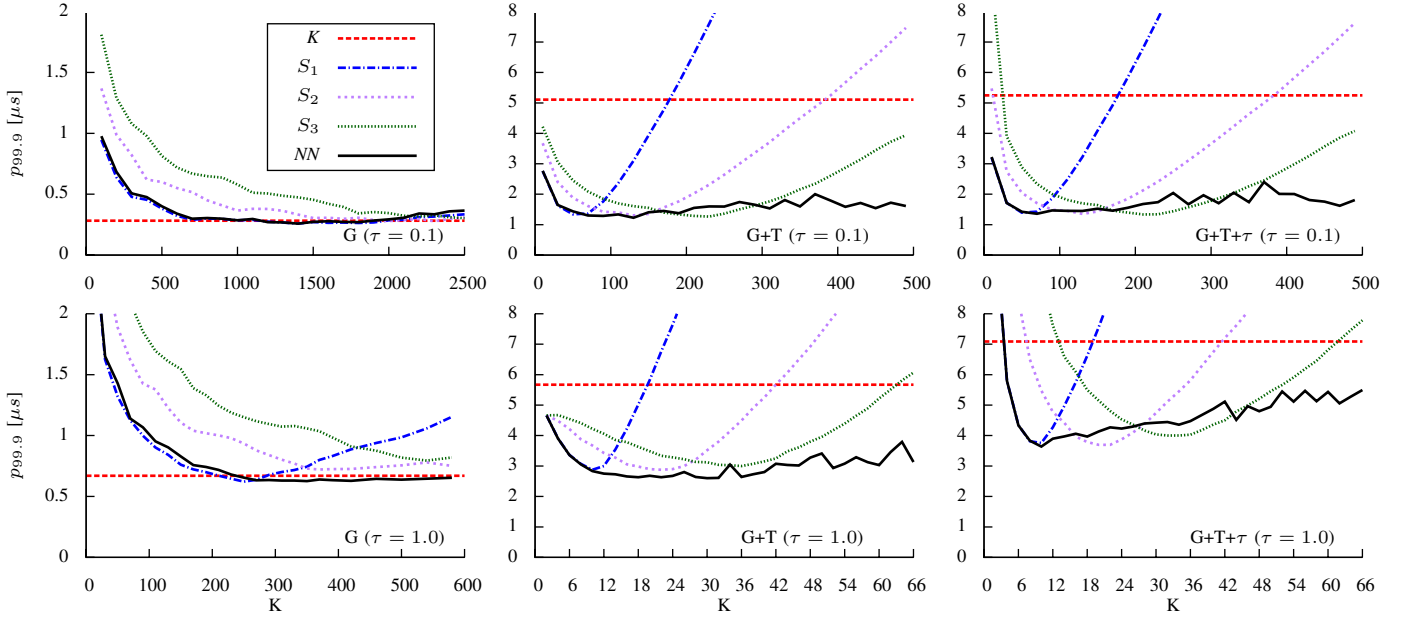


Fig. 2. 99.9-percentile of the synchronization error, for  $K$ ,  $S_1$ ,  $S_2$ ,  $S_3$  and  $NN$  control algorithms, for the  $G$ ,  $G+T$  and  $G+T+\tau$  clock models, and with  $\tau = 0.1$  s and  $\tau = 1.0$  s

V, splines achieve good performance when  $K$  is large enough to obtain a reliable result from the regression applied to the available noisy points and small enough to avoid capturing wrong trends from the data. The  $S_1$  with  $\tau = 1.0$  s is topical in this perspective because it reveals an increasing sensitivity to noise with  $K > 260$ . The same concept holds true for the middle column of Fig. 2 ( $G+T$  scenario), in which all the splines experience a very short range of the optimal  $K$ . In that case, the  $NN$  is much more robust to the setting of  $K$  and guarantees the best performance because the noise is not Gaussian anymore. The performance gap of Kalman is significant as well. It is also remarkable that the splines' performance is not distant from the best one, in particular with  $\tau = 0.1$  s. This is an attractive property because splines require much less implementation and computational effort than the neural approach. The synchronization accuracy significantly degrades for all the methods when results are obtained after the synchronization instants, and its worsening is directly proportional with the size of the gap ( $G+T+\tau$  scenario). It is more evident with  $\tau = 1.0$  s, because the estimation of  $\hat{\gamma}$  becomes older. For  $NN$ , less robustness to  $K$  is evidenced for  $\tau = 1.0$  s (not for  $\tau = 0.1$  s). This is due to less accuracy of the  $\hat{\gamma}$  estimation, based on  $S_1$ , which degrades with increasing  $K$ . This suggests the extension of the  $NN$  including  $\hat{\gamma}$ , which can be easily incorporated since the scalar output of the  $NN$  should be replaced by a vector addressing both  $\hat{\theta}$  and  $\hat{\gamma}$ . The performance of the ' $T$ ' model, mentioned in the previous subsection, is not depicted in Fig. 2 because it is very similar to the one of the ' $G+T$ ' case. Overall, the neural approach reveals to be essential to match possible non-Gaussian behaviors of the noises. The splines could be also helpful in this perspective, but they need an accurate tuning of the size of the information vector.

The  $NN$  estimation algorithm has been also implemented on

TABLE II  
THE CONFIGURATION PARAMETERS OF THE FUNCTION  $\omega_{\gamma T}(k, \cdot)$  FOR THE DATABASES USED IN THE SIMULATION.

Environment		$T_E^{Low}$ [°C]	$T_E^{High}$ [°C]	$tc$ [s]	$p$ [s]
Train	A	-10	40	20	200
	B	-10	40	60	600
	C	-10	40	180	1800
Test	D	-5	35	100	1200
	E	-30	60	10	100
Train	F	-30	60	10	100
	G	-30	60	60	600
	H	-30	60	180	1800
Test	I	-10	40	20	200
	L	-35	65	20	200

$\sigma_\theta^2 = 10^{-17} (s^2)$  and  $\sigma_\gamma^2 = 10^{-19} (s^2)$

an Atmel ATmega328P microcontroller running at 16 MHz. The aim was to evaluate the inherent computational effort on a microcontroller with the typical complexity of those used in WSN nodes. The results previously obtained with  $\tau = 1$  s and  $K = 15$  have been taken as a reference because they are compatible with the execution of the algorithm in the WSNs context (i.e., with high  $\tau$  and small  $K$ ). When setting these parameters on the microcontroller, the registered mean execution time of each iteration involving both features extraction (from  $S_1$ ) and the computation of the output of the trained  $NN$  was 4.932 ms, which is much smaller than the used  $\tau$ . The algorithm scales linearly with respect to  $K$ .

#### A. Generalization capabilities of the $NN$

In this set of experiments, the proposed estimation methods are now tested in the ' $G+T$ ' scenario under variable conditions involving temperature thresholds ( $T_E^{High}$ ,  $T_E^{Low}$ ), thermal time constant ( $tc$ ) and object' movements ( $p$ ). Table II summarizes all the tested conditions. The temperature thresholds were



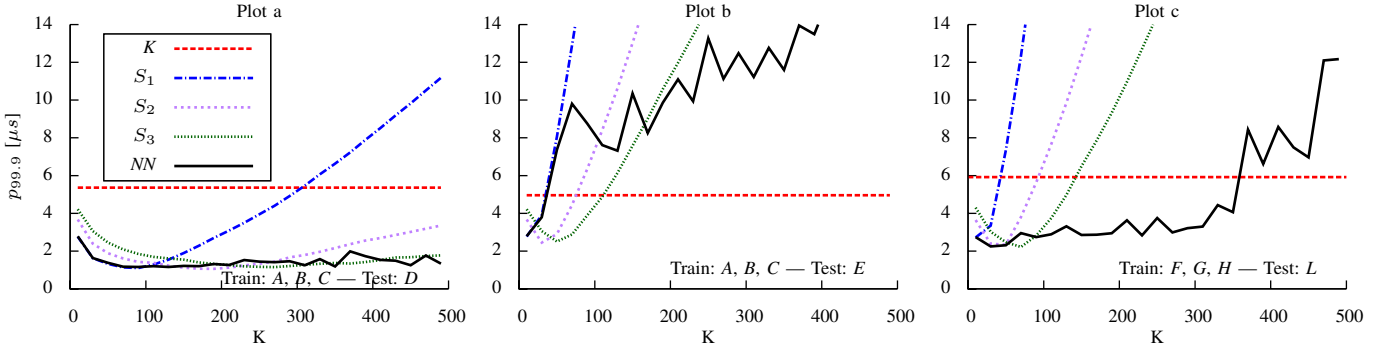


Fig. 3. 99.9-percentile of the synchronization error for three train and test conditions and for  $K$ ,  $S_1$ ,  $S_2$ ,  $S_3$  and  $NN$  control algorithms ( $\tau = 0.1$  s).

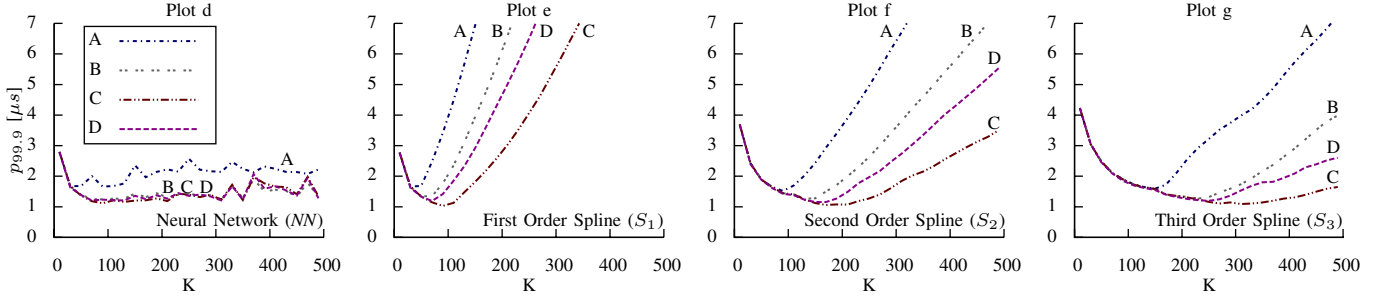


Fig. 4. 99.9-percentile of the synchronization error for  $S_1$ ,  $S_2$ ,  $S_3$  and  $NN$  control algorithms for conditions A, B, C, D ( $\tau = 0.1$  s).

chosen to represent environments with medium ( $T_E^{Low} = -10^\circ\text{C}$  and  $T_E^{High} = 40^\circ\text{C}$ ) and high temperature excursions ( $T_E^{Low} = -30^\circ\text{C}$  and  $T_E^{High} = 60^\circ\text{C}$ ). The time constant, that represents the thermal inertia of the XO, was chosen between two extremes:  $tc = 10$  s represents a XO poorly insulated, or placed in a fan cooled case, while  $tc = 180$  s is referred to a well thermal insulation. The  $p$  value is about ten times  $tc$  to allow the XO to reach the new temperature after a temperature change. In such a way, results include all of the temperature excursions between  $T_E^{High}$  and  $T_E^{Low}$ .

In the first plot of Fig. 3 (plot a), the  $NN$  has been trained with the environmental conditions A, B and C. They represent a range of conditions that should cover all the typical temperature variations to which a XO may be exposed. The test condition D represents a situation in which the parameters of the  $\omega_{\gamma T}(\cdot)$  noise assume values within the range of the parameters used in the training phase. Despite the parameters of condition D are inside the training range, the inherent values denote significantly different environmental conditions involving  $T_E^{High}$ ,  $T_E^{Low}$ ,  $p$  and  $tc$ . Plot a helps highlight the generalization capability of the  $NN$  model. It achieves the lowest  $p_{99.9}$ , together with the splines, but over a larger range of  $K$ . It is however worth remembering that only the  $NN$  method needs a training phase, the splines and Kalman are applied directly during the test phase. When the test parameters are outside the training range, the  $NN$  does not generalize well (plot b). All the algorithms are however very sensitive to the  $K$  parameter in plot b. Kalman has a higher  $p_{99.9}$  over all the presented settings (plots a, b and c), even if it is always updated with a perfect knowledge of the noises'

statistics, which are numerically evaluated in advance.

A way to correct the poor  $NN$  results of plot b is to enlarge the training set even more, by including  $T_E^{High}$  and  $T_E^{Low}$  of test set E and by keeping variable  $p$  and  $tc$ . This corresponds to train conditions F, G and H. Two test sets are used: I and L (see again Table II). Test set I is equivalent to D (i.e., temperature range included in the training set), but with a larger temperature range than D. Test set L is equivalent to E (i.e., temperature range not included in the training set), but with smaller oscillations outside training. The results of test I are satisfactory because they lead to a situation qualitatively very similar to plot a; the inherent plot is not reported for the sake of synthesis. As a result of this, it appears relevant how the  $NN$  is capable to achieve the best performance over a large range of the  $K$  coefficient, together with good generalization capabilities if it operates in environments whose  $T_E^{High}$  and  $T_E^{Low}$  have been used during training. The results of test L are shown in plot c of Fig. 3. The values of  $T_E^{High}$  and  $T_E^{Low}$ , which slightly lie outside training, cause performance oscillations of the  $NN$  with respect to  $K$ . Those oscillations do not however lead to a performance decrease as appears in plot b. As far as the splines are concerned with respect to tests I and L, their minimum reveals again to be very sensitive to  $K$  (the I case is not reported but it is very similar to the L case of plot c).

In this perspective, Fig. 4 highlights the dependence between  $p_{99.9}$  and the  $K$  coefficient over conditions A, B and C. Similar results are experienced with F, G and H. In the  $NN$  case, presented in plot d, there is a wide range of  $K$  values for which the algorithm achieves the minimum of  $p_{99.9}$  in all the



considered scenarios ( $A$ ,  $B$  and  $C$ ). The same does not hold true for the splines as evidenced by plots  $e$ ,  $f$  and  $g$ . Choosing a unique  $K$  for all conditions  $A$ ,  $B$  and  $C$  is almost impossible with the splines. For this reason, the  $NN$  reveals to be the most suitable algorithm for the synchronization problem in the presence of temperature variations. The inherent generalization capabilities guarantee good performance over a large set of environments, this avoiding any on-line adjustment of the algorithm.

## IX. CONCLUSIONS AND FUTURE WORK

The paper has presented an innovative estimation approach to clock offset synchronization. It is of interest when the Gaussian hypothesis for clock state equations is not applicable, e.g., in the presence of temperature variations. The approach reveals to be more precise than regular Kalman filtering and more robust to parameters setting with respect to other regression schemes. It reveals to be also applicable to other situations in which other traditional hypotheses, involving linear dynamics or quadratic cost functions, are not met.

As the optimal filter is analytically unknown, future work deals with the adoption of other approximating schemes, together with the adoption of other elements of non-linearity, e.g., in other wireless contexts, e.g., [13] and by also including some energy metric.

## REFERENCES

- [1] G. Cena, I. Cibrario Bertolotti, S. Scanzio, A. Valenzano, and C. Zunino, "Synchronize your watches: Part I: General-purpose solutions for distributed real-time control," *IEEE Industrial Electronics Magazine*, vol. 7, no. 1, pp. 18–29, 2013.
- [2] —, "Synchronize Your Watches: Part II: Special-Purpose Solutions for Distributed Real-Time Control," *IEEE Industrial Electronics Magazine*, vol. 7, no. 2, pp. 27–39, 2013.
- [3] J. R. Vig, "Quartz Crystal Resonators and Oscillators; For Frequency Control and Timing Applications - A Tutorial," Tutorial, US Army Communications-Electronics Research, Development & Engineering Center Fort Monmouth, NJ, USA, Rev. 8.5.2.2, 2004.
- [4] D. Fontanelli, D. Macii, P. Wolfrum, D. Obradovic, and G. Steindl, "A clock state estimator for PTP time synchronization in harsh environmental conditions," in *International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*, 2011, pp. 99–104.
- [5] G. Giorgi and C. Narduzzi, "Performance Analysis of Kalman-Filter-Based Clock Synchronization in IEEE 1588 Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 8, pp. 2902–2909, 2011.
- [6] P. Wolfrum, R. Scheiterer, and D. Obradovic, "An optimal control approach to clock synchronization," in *International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*, 2010, pp. 122–128.
- [7] D. Fontanelli and D. Macii, "Accurate time synchronization in PTP-based industrial networks with long linear paths," in *International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication (ISPCS)*, 2010, pp. 97–102.
- [8] A. Bletsas, "Evaluation of Kalman filtering for network time keeping," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 52, no. 9, pp. 1452–1460, 2005.
- [9] R. Zoppoli, M. Sanguineti, and T. Parisini, "Approximating networks and extended Ritz method for the solution of functional optimization problems," *J. Optim. Theory Appl.*, vol. 112, no. 2, pp. 403–440, Feb. 2002.
- [10] M. Akhlaq and T. Sheltami, "RTSP: An Accurate and Energy-Efficient Protocol for Clock Synchronization in WSNs," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 3, pp. 578–589, 2013.
- [11] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys)*. New York, NY, USA: ACM, 2004, pp. 39–49.
- [12] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *Oper. Syst. Rev. (SIGOPS)*, vol. 36, pp. 147–163, Dec. 2002.
- [13] G. Cena, S. Scanzio, A. Valenzano, and C. Zunino, "The reference-broadcast infrastructure synchronization protocol," in *IEEE 17th Conference on Emerging Technologies Factory Automation (ETFA)*, 2012, pp. 1–4.
- [14] R. Scheiterer, C. Na, D. Obradovic, and G. Steindl, "Synchronization Performance of the Precision Time Protocol in Industrial Automation Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 1849–1857, 2009.
- [15] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. 1–269, 2008.
- [16] R. Bechmann, A. Ballato, and T. Lukaszek, "Higher-Order Temperature Coefficients of the Elastic Stiffnesses and Compliances of Alpha-Quartz," *Proceedings of the IRE*, vol. 50, no. 8, pp. 1812–1822, 1962.
- [17] T. Fiorenzani, C. Manes, G. Oriolo, and P. Peliti, "Comparative study of unscented kalman filter and extended kalman filter for position/attitude estimation in unmanned aerial vehicles," *Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti"*, CNR, 2008. [Online]. Available: <http://www.iasi.cnr.it/reports/R8008/R8008.html>
- [18] B. Akesson, J. Jorgensen, N. Poulsen, and S. Jorgensen, "A tool for kalman filter tuning," *Computer Aided Process Engineering*, vol. 24, pp. 859–864, 2007.
- [19] A. Aravkin, "Robust methods for kalman filtering/smoothing and bundle adjustment," *PhD Dissertation, Univ. of Washington, Seattle*, 2010.
- [20] W. H. Greene, *Econometric analysis*. Prentice Hall, 2002.
- [21] M. Baglietto, F. Davoli, M. Marchese, and M. Mongelli, "Neural approximation of open-loop feedback rate control in satellite networks," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1195–1211, 2005.
- [22] F. Davoli, M. Marchese, and M. Mongelli, "Non-linear coding and decoding strategies exploiting spatial correlation in wireless sensor networks," *IET Communications*, vol. 6, no. 14, pp. 2198–2207, 2012.
- [23] —, "Neural decision making for decentralized pricing-based call admission control," in *IEEE International Conference on Communications (ICC)*, vol. 3, 2005, pp. 1556–1560 Vol. 3.
- [24] 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.
- [25] H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.
- [26] D. B. Sullivan, D. W. Allan, D. A. Howe, and F. L. Walls, "Characterization of clocks and Oscillators," *NIST, Technical Note 1337*, pp. 1–342, 1990.
- [27] G. Gaderer, A. Nagy, and P. Loschmidt, "Achieving a Realistic Notion of Time in Discrete Event Simulation," *International Journal of Distributed Sensor Networks*, vol. 2011, pp. 1–11, 2011.
- [28] P. Loschmidt, R. Exel, A. Nagy, and G. Gaderer, "Limits of synchronization accuracy using hardware support in IEEE 1588," in *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS)*, 2008, pp. 12–16.
- [29] D. Ingram, P. Schaub, and D. Campbell, "Use of Precision Time Protocol to Synchronize Sampled-Value Process Buses," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 5, pp. 1173–1180, 2012.
- [30] M. Lixia, A. Benigni, A. Flammini, C. Muscas, F. Ponci, and A. Monti, "A Software-Only PTP Synchronization for Power System State Estimation With PMUs," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 5, pp. 1476–1485, 2012.
- [31] P. Regnier, G. Lima, and L. Barreto, "Evaluation of interrupt handling timeliness in real-time Linux operating systems," *Oper. Syst. Rev. (SIGOPS)*, vol. 42, no. 6, pp. 52–63, Oct. 2008.
- [32] P. Ferrari, A. Flammini, S. Rinaldi, A. Bondavalli, and F. Brancati, "Experimental Characterization of Uncertainty Sources in a Software-Only Synchronization System," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 5, pp. 1512–1521, 2012.