Design of PTP TC/Slave Over Seamless Redundancy Network for Power Utility Automation

Soonwoo Lee, Jimyung Kang, Sung Soo Choi, *Member, IEEE*, and Myo Taeg Lim, *Member, IEEE*

***Abstract*— IEC 61850, which drives to digitalize information in the entire system, is being increasingly adopted in the power utility industry, since the last decade. It is necessary for all devices communicating with distributed digital data to have a common understanding of time. As protection and control devices are directly related to the safety of the power system, they are strictly required to have time accuracies. This paper describes the design of a hybrid slave clock with the transparent clock (TC) functionality defined in the IEEE 1588 precision time protocol. The design objective includes not only the synchronization of the time accuracy but also the transient time, as specified in the IEC/IEEE 61850-9-3 precision time profile for power utility automation. To comply with these objectives, we adopt a hardware–software co-design approach. The timestamp of each port is implemented by hardware. Also, they are connected by hardware to calculate the bridge time on the fly, enabling the design to have a one-step TC functionality with low latency. On the other hand, the synchronizing controller is implemented by software for adapting to the varying external conditions. The controller uses the linear quadratic Gaussian (LQG) to achieve stable steady-state performance. Simultaneously, the least-squares estimator is utilized to estimate the initial conditions of the LQG controller for reducing the transition time. Experimental results show the three-sigma time inaccuracies are 15.6 ns for the TC and 149 ns for the slave with an average transition time of 13.5 s, satisfying the IEEE/IEC 61850-9-3 criteria.**

***Index Terms*— Finite impulse response (FIR) filter, high availability seamless redundancy (HSR), IEC 62439-3, IEEE 1588, linear quadratic Gaussian (LQG) controller, power utility automation, precision time protocol (PTP), redundancy network, time synchronization.**

I. Introduction

THE digital revolution has transformed the electric power utility industry. In the digital era, every device has its interface or protocol to be controlled and monitored. IEC 61850, a vendor independent protocol for substation and power systems, has been developed over the past decade for enabling standardized communication between various digital equipments. The structure of this standard can be divided into three levels: the station, bay, and process levels. Among these, our focus is on the process layer, which is related to the management of primary data such as voltage and current with switchyard equipment. It requires robust real-time communication because it transports time critical information such as sample values or generic object oriented substation event messages.

The parallel redundancy protocol (PRP) or the high availability seamless redundancy (HSR) defined in IEC 62439-3 [1] are suitable solutions for meeting these requirements as they provide zero recovery time against a single failure in the network. This feature enables these protocols to reference network topologies for the process bus [2].

Together with the network redundancy, time synchronization is a critical issue [3]–[5]. All the devices in the process bus have the same reference time. According to IEC 61850-5 [6], the accuracy should not exceed ± 1 μs for protection class P3 and metering classes, M2 and M3. The precision time protocol (PTP) defined in IEC 61588 (or IEEE 1588) [7] is used as the reference time protocol [2] and it ensures a time accuracy under ± 1 μs with dedicated hardwired timestamps. Fig. 1 shows the PTP node configuration in the process bus utilizing the PRP/HSR protocol. Each node, representing a process level device, is synchronized with a grand master clock (GMC).

The GMC periodically broadcasts a sync message containing its reference time on both the sides: sync A and sync B. The sync message transits through the network and suffers a network delay consisting of the link and residence delays for each node, before it reaches the destination. Therefore, each node in the ring should be a transparent clock (TC) because it forwards a sync message, while correcting the time by evaluating both the peer delay and residence time of the sync message. A node called hybrid clock or TC/slave is synchronized with the GMC, while transferring PTP messages to the others, as a bridge. Most nodes that follow the HSR protocol in the process bus assume the form of a hybrid TC/slave for time synchronization; hence, they are focused upon, in this paper.

The design objective of a TC/slave is to meet the functions and performance requirements of the IEC/IEEE 61850-9-3

Manuscript received August 09, 2017; revised October 09, 2017; accepted December 27, 2017. Date of publication February 21, 2018; date of current version June 7, 2018. This work was supported by the Korea Electrotechnology Research Institute Primary Research Program through the National Research Council of Science and Technology funded by the Ministry of Science, ICT and Future Planning under Grant N17-12-N0101-50. The Associate Editor coordinating the review process was Dr. Chao Tan. *(Corresponding author: Myo Taeg Lim.)*

S. Lee is with the School of Mechatronics, Korea University, Seoul 136-701, South Korea, and also with the Power Telecommunication Research Center, Korea Electrotechnology Research Institute, Ansan 426-910, South Korea.

J. Kang and S. S. Choi are with the Power Telecommunication Research Center, Korea Electrotechnology Research Institute, Ansan 426-910, South Korea (e-mail: jmkang@keri.re.kr; sschoi@keri.re.kr).

M. T. Lim is with the School of Electrical Engineering, Korea University, Seoul 136-701, South Korea (e-mail: mlim@korea.ac.kr).

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.2018.2800858

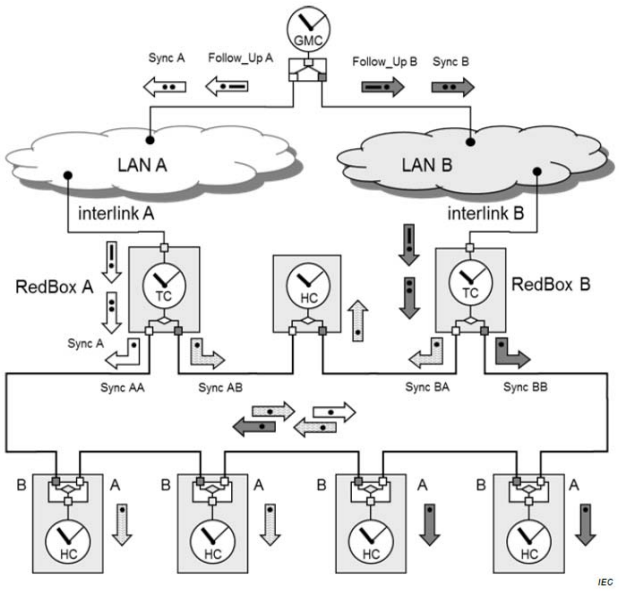

Fig. 1. Example configuration of PTP devices over a PRP/HSR network [1].

international standard of the precision time protocol profile for power utility automation [8]. There are two important points to be noted: 1) the TC should introduce a time accuracy of less than 50 ns and 2) the steady state is defined as 30 s, after the start of a single master and 16 s, after a master changes.

II. Related Works

Related works have been published, as the PTP is applied in various areas. Han and Jeong [9] implemented a TC and presented the overall system architecture and detailed operation. Recently, Moreira *et al.* [10] presented a PTP IP working on a Xilinx field-programmable gate array (FPGA) device and compared four different hardware–software architectures for implementing the PTP functionalities. They demonstrated an excellent time offset under 20 ns in terms of the standard deviation; however, they did not consider the transient time sufficiently. In contrast, Kirmann *et al.* [11] suggested a full-hardware PTP implementation to enhance area and memory utilization of the FPGA. They implemented the best master clock algorithm (BMCA) and management logic using hardware with no processor support, which reduces the FPGA resources and increases portability. This approach, however, is limited to the designing of the filter for the slave's clock, in order to achieve high performance against varying noise conditions. Furthermore, it reduces the benefit in the case of FPGA based on system-on-chip platform integrating dedicated processors.

The filter design for the clock servo in a slave is also crucial. Giorgi and Narduzzi [12] introduced a state-variable clock model and developed a Kalman filter-based clock controller employing the model. Wolfrum *et al.* [13] formulated clock synchronization as a stochastic estimation control problem and a linear quadratic Gaussian (LQG) controller was derived. The simulation result shows that the LQG controller significantly reduces the error compared to the standard PTP algorithm; however, they did not consider the transient time response as thoroughly as Giorgi and Narduzzi [12].

Some papers have considered not only the steady-state performance but also the transient time, for the design. Chao *et al.* [14] combined fuzzy control and proportional integral control schemes to meet the two requirements. Although they reduced the transient time compared to the traditional PI control scheme, it was still insufficient to meet the stringent rules of the IEC 61850-9-3. Viejo *et al.* [15] achieved a fast convergence time of 15 s for an adjustment interval of 1 s with a complete hardware implementation. Despite the fast speed of the design, it is vulnerable to noise during steady state because of the fully hardwired simple offset correction scheme. Moreover, the transient time response chiefly depends upon the initial condition of the slave clock.

In this paper, an efficient combination of both hardware and software components is presented to meet the design constraints. The TC part, mainly consisting of a data path requiring real-time operation and high-precision time-stamping, is implemented by hardware, whereas the slave part for synchronizing the local clock with the master's by parsing the received PTP packet is described in software. In the slave part, in particular, the design pursues fast convergence time, irrespective of the initial clock condition as well as stable steady-state performance compatible with the IEC 61850-9-3. To enhance immunity from noise in the steady state, an infinite impulse response (IIR) filter-based LQG controller is adopted for clock synchronization. Simultaneously, to reduce the transition time, regardless of the initial condition of the slave clock, a receding horizon recursive estimator [16], which is a type of finite impulse response (FIR) filter, is applied to predict the initial state of the clock. The hybrid method attempts to combine the fast convergence of FIR filters and the high steady-state performance of IIR filters with a lower complexity [17]–[19]. Pak *et al.* [17] attempted to improve the reliability of localization through hybrid particle/FIR filtering. In their scheme, particle filter failure is recovered by an auxiliary FIR filter. Ferro *et al.* [18] introduced an FIR/IIR combination for providing good convergence, while maintaining a low computational cost in system identification. Lin and Smith [19] used hybrid filters to maximize the compression performance for images.

In the next section, the system models are derived. In Section IV, focusing on the TC part, the hardware structures are overviewed and the implementation results are discussed. In Section V, the clock filter/controller, which is the core of the PTP slave, is designed and the test results are evaluated. Finally, in Section VI, the paper is summarized and concluded.

III. Clock Model

It is assumed that several peer-to-peer (P2P) TC/slaves are connected to a master clock (MC) in a cascaded form as shown in Fig. 2. All the TC/slaves are numbered sequentially from 1 to i in accordance with the distance from the MC. The state of clock, $\mathbf{x}_i(t)$, of the ith TC/slave can be modeled as $\mathbf{x}_i(t) = [\rho_i(t)\ \tau_i(t)]^T$ with a frequency, $\rho_i(t)$ and time, $\tau_i(t)$ [12], [20]. If we define the difference between the MC,

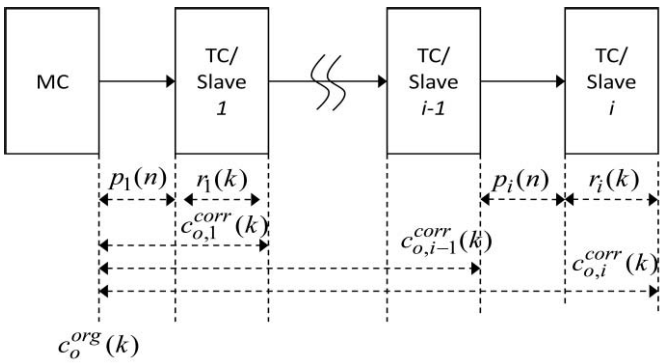

Fig. 2. Network configuration of the MC and the TC/slaves.

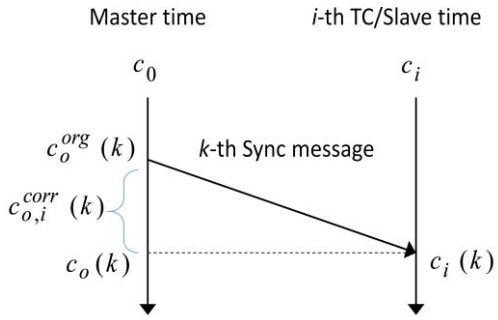

Fig. 3. Measurement time for the kth sync message passing through the ith TC/slave.

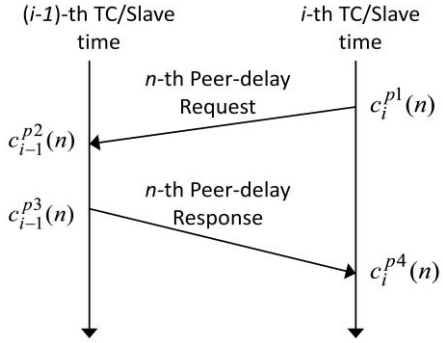

Fig. 4. Peer-delay mechanism [7].

$\mathbf{x}_0(t)$ and the ith TC/slave clock, $\mathbf{x}_i(t)$, as $\mathbf{e}_i(t) = \mathbf{x}_i(t) - \mathbf{x}_0(t)$, the discrete time state vector $\mathbf{e}_i(k)$ the sampled version of $\mathbf{e}_i(t)$ with a period ΔT can be expressed as (1). The sampling period ΔT is the same as the interval of the sync message generated by the MC; generally, it is 1 s as per the default setting of the IEC/IEEE 61850-9-3 [8]. Note that $\mathbf{e}_i(k)$ can be interpreted as the state error of the ith TC/slave with respect to the MC, when the kth update is performed by the kth sync message in the TC/slave

$$\begin{aligned} \mathbf{e}_i(k+1) &= F\mathbf{e}_i(k) + \mathbf{u}_i(k) + \mathbf{w}_i(k) \\ y_i(k) &= H\mathbf{e}_i(k) + v_i(k). \end{aligned} \tag{1}$$

1) The state vector, $\mathbf{e}_i(k) = \mathbf{x}_i(k) - \mathbf{x}_0(k)$, where $\mathbf{x}_i(k) = [\rho_i(k)\tau_i(k)]$ and $\mathbf{x}_0(k) = [1\ \tau_0(k)]$ represent the clock states of the ith TC/slave and the MC, respectively. $\rho_i(k)$ is the normalized frequency with respect to that of the master. $\tau_i(k)$ and $\tau_0(k)$ are the time terms of the ith TC/slave clock and the MC, respectively.
2) The transition matrix, $F = e^{\Delta T A}$, $A = \left[\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right]$.
3) The control input, $\mathbf{u}_i(k) = [u_{\rho_i}\ 0]$, is composed of the frequency and time. For this design, only the frequency term, u_{ρ_i}, is used as the control parameter because a direct change of time can cause problems in the host applications.
4) The process noise, $\mathbf{w}_i(k) = \int_0^{\Delta T} e^{tA}\{\mathbf{n}_i(t) - \mathbf{n}_0(t)\}dt$,
5) Where $\mathbf{n}_i(t) = [n_{\rho_i}\ n_{\tau_i}]$ and $\mathbf{n}_0(t) = [n_{\rho 0}\ n_{\tau 0}]$ are the noise vectors of the ith TC/slave and the MC, respectively, comprising the frequency noise, n_{ρ_i}, n_{ρ_0} and the time noise n_{τ_i}, n_{τ_0}. They are assumed to have normal distributions: $n_{\rho_i} \sim N(0, \sigma_{n_\rho}^2)$, $n_{\rho_0} \sim N(0, \sigma_{n_{\rho 0}}^2)$, $n_{\tau_i} \sim N(0, \sigma_{n_\tau}^2)$, $n_{\tau_0} \sim N(0, \sigma_{n_{\tau 0}}^2)$.
6) $H = [0\ 1]$
7) The measurements are $y_i(k)$ and the measurement noise is v_i.

Note that $y_i(k)$ can be interpreted as $y_i(k) = c_i(k) - c_0(k)$, which is the difference in the measured time or timestamps of the ith TC/slave, $c_i(k)$, and the MC, $c_0(k)$, for the kth sync message passing through the TC/slave, as depicted in Fig. 3. However, $c_0(k)$ cannot be known directly in the TC/slave; it can be estimated from the *originTimestamp* and *correctionField* enclosed in each sync message, which arrives in the TC/slave. The value of *originTimestamp* $c_0^{\text{org}}(k)$ for the kth sync message is generated from the TX timestamp in the MC. As illustrated in Fig. 3, the value of *correctionField* $c_{0,i}^{\text{corr}}(k)$ is the cumulative line and bridging delays from the MC up to the ith TC/slave for the kth sync message. The sum of these values is equal to $c_0(k)$, as represented in the following equation:

$$\begin{aligned} y_i(k) &= c_i(k) - c_0(k) \\ c_o(k) &= c_o^{\text{org}}(k) + c_{o,i}^{\text{corr}}(k). \end{aligned} \tag{2}$$

As shown in Fig. 2, $c_{0,i}^{\text{corr}}(k)$ can be divided into three terms: $c_{0,i-1}^{\text{corr}}(k)$, the line delay, $p_i(n)$, and the bridging delay, $r_i(k)$, of the ith TC/slave, as in the following equation:

$$\begin{aligned} c_{o,i}^{\text{corr}}(k) &= p_i(n) + r_i(k) + c_{o,i-1}^{\text{corr}}(k) \\ c_{o,0}^{\text{corr}}(k) &= 0. \end{aligned} \tag{3}$$

The line delay, $p_i(n)$, between two adjacent ports is calculated from the peer-delay mechanism [7]. In general, the order of the peer-delay measurement differs from that of the sync message ($k \neq n$). According to the peer-delay mechanism, $p_i(n)$ is estimated from $c_i^{p1}(n)$ and $c_{i-1}^{p2}(n)$ and $c_{i-1}^{p3}(n)$ and $c_i^{p4}(n)$, which represent the TX and RX timestamps for the nth peer-delay request message and the nth peer-delay response message, respectively, as illustrated in Fig. 4. $\hat{p}_i(n)$ and the estimation of $p_i(n)$ are expressed by the following equation:

$$\widehat{p}_i(n) = \frac{1}{2} \times \left\{ \frac{c_i^{p4}(n) - c_i^{p1}(n)}{\hat{\rho}_i(n)} + \frac{c_{i-1}^{p3}(n) - c_{i-1}^{p2}(n)}{\hat{\rho}_{i-1}(n)} \right\}. \tag{4}$$

In (4), $\hat{\rho}_i(n)$ and $\hat{\rho}_{i-1}(n)$ are the estimated normalized frequencies of the ith TC/slave and the $(i-1)$th TC/slave

clocks, respectively. $p_i(n)$ can be expressed by (5), where the timestamp noise is assumed to have the same normal distribution $N(0, \sigma_\varepsilon^2)$, regardless of the ports and directions. For simplicity, each normalized frequency, $\hat{\rho}_i(n)$, $\hat{\rho}_{i-1}(n)$, is assumed to be unity because the errors are negligible (under 10^{-6}) for deriving the noise, $\varepsilon_p(n)$, as in the following equation:

$$p_i(n) = \widehat{p}_i(n) + \varepsilon_p(n), \quad \varepsilon_p(n) \sim N(0, \sigma_\varepsilon^2). \tag{5}$$

Similarly, the bridging delay, $r_i(k)$, in (3) can be expressed by the following equation:

$$\begin{aligned} r_i(k) &= \widehat{r}_i(k) + \varepsilon_r(k), \varepsilon_r(k) \sim N(0, 2\sigma_\varepsilon^2) \\ \widehat{r}_i(k) &= \frac{c_i^{s2}(k) - c_i^{s1}(k)}{\hat{\rho}_i(k)}. \end{aligned} \tag{6}$$

The bridging delay noise, $\varepsilon_r(k)$, is modeled as a normal distribution with zero mean and $2\sigma_\varepsilon^2$ standard deviation, assuming $\hat{\rho}_i(k)$ to be unity. $c_i^{s1}(k)$ and $c_i^{s2}(k)$ are the RX and TX timestamps, respectively, for the kth sync message passing through the ith TC/slave.

If the signals are described by dividing the estimation and noise terms as in (7), the measurement, $y_i(k)$, can be expressed as (8), from (2)–(7). In (7) and (8), I indicates the number of hops from the MC

$$\begin{aligned} y_i(k) &= \widehat{y}_i(k) + \varepsilon_{y_i}(k) \\ c_i(k) &= \widehat{c}_i(k) + \varepsilon_i(k) \\ c_o^{\text{org}}(k) &= \widehat{c}_o^{\text{org}}(k) + \varepsilon_o(k) \\ c_{o,i}^{\text{corr}}(k) &= \widehat{c}_{o,i}^{\text{corr}}(k) + I \times \{\varepsilon_p(n) + \varepsilon_r(k)\} \\ &\qquad \text{where, } I : \#\text{ hop from master.} \end{aligned} \tag{7}$$

Note that in (7), $\varepsilon_i(k)$ and $\varepsilon_0(k)$ can be modeled with $\varepsilon_i(k) \sim N(0, \sigma_\varepsilon^2)$, $\varepsilon_o(k) \sim N(0, \sigma_\varepsilon^2)$ because each noise is related to a single timestamp

$$\begin{aligned} \widehat{y}_i(k) &= \widehat{c}_i(k) - \left\{\widehat{c}_o^{\text{org}}(k) + \widehat{c}_{o,i}^{\text{corr}}(k)\right\} \\ \varepsilon_{y_i}(k) &= \varepsilon_i(k) + [\varepsilon_o(k) + I \times \{\varepsilon_p(n) + \varepsilon_r(k)\}] \\ &\qquad \text{where, } I : \#\text{ hop from master.} \end{aligned} \tag{8}$$

From the noise models of $\varepsilon_i(k)$, $\varepsilon_0(k)$, $\varepsilon_p(n)$, $\varepsilon_r(k)$, and (8), it can be observed that the measurement noise power, $\varepsilon_{y_i}(k)$, increases by $3\sigma_\varepsilon^2$ on increasing the number of hops from the MC.

IV. Hardware Design

A. Design Structures

Each node in the HSR network shown in Fig. 1 has a general form of communication interface. There are two redundant ports (A and B), an *interlink port* (IL), and a *link layer interface* (LLI) as depicted in Fig. 5. To achieve seamless redundancy, a node sends the same frame in both the directions (A and B) and each node relays the frame, until it reaches the source again or is otherwise discarded. Therefore, a destination node receives two frames on both the ports; it uses the first and discards the duplicate. The IL and LLI are optional ports, depending upon the role of the node. The IL links non-HSR nodes to HSR rings. Upper layer protocols such as those of the network layer and transport layer or applications can be connected by LLI [1].

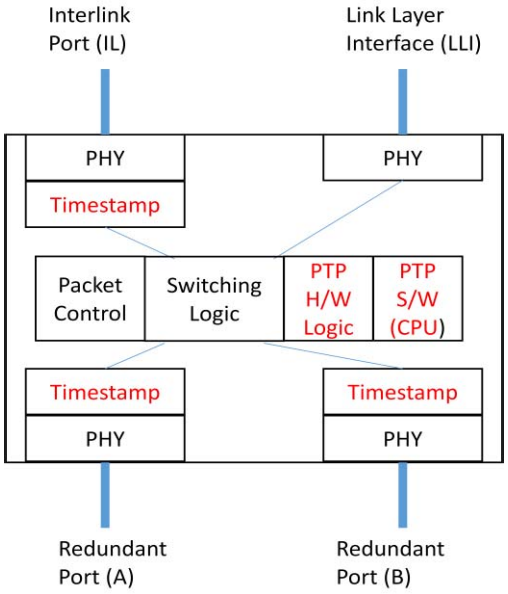

Fig. 5. IEEE 1588 TC/slave structure supporting seamless redundancy, compliant with IEC 62439-3.

For the PTP functionality of a node, all the external ports, including A, B, and IL have timestamps to mark the time at which the incoming or outgoing messages pass a reference plane, based on the value of the local clock. The timestamp values are used for calculating the time taken by a PTP message to traverse a node, i.e., the bridging or residence time. In addition, the link delays or time offsets from the master node need to be measured [7].

The detailed design is depicted in Fig. 6. All the ports have the same structure, except LLI because this port does not need a timestamp. The *timestamp units* (TSUs) of the RX and TX are located outside the switching logic to maintain fixed delays on the timestamp reference plane, regardless of the switching time. To support a one-step TC for every port, each *time correction unit* in the TX can directly access every TSU of the RX. Synchronizing with the MC, the *PTP slave clock* generates the reference time for the TSUs. The *clock control* module periodically calculates the time and/or frequency adjustment to be synchronized with the MC, from the PTP messages and related timestamps. The *timestamp insertion unit* merges a PTP message with the associated TX timestamp for forwarding to the *clock control* module.

Fig. 7 illustrates the structure of the *PTP slave clock* and the associated variables. The *current_time* consists of the 48-bit second part, the 32-bit nanosecond part, and the 32-bit sub-nanosecond part, which is compatible with IEC 61588 [7]. After being initialized with the *time_offset* value, the *current_time* is mainly adjusted by the *frequency* value with a precision of 2^{-32} ns. It is important to generate suitable *frequency* values because they determine the synchronization accuracy, which is the role of the synchronization controller described in Section V.

B. Implementation Results

The proposed design was implemented on a Xilinx KCU 105 evaluation board, as shown in Fig. 8. The two

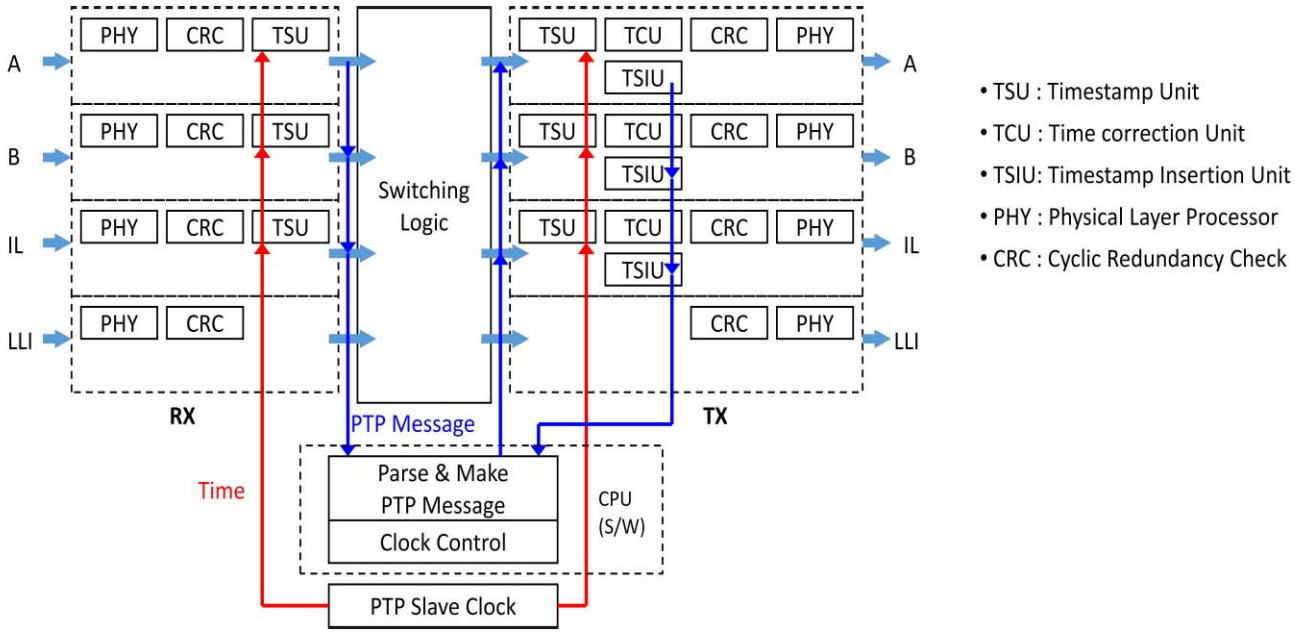

Fig. 6. Design structure of the TC/slave.

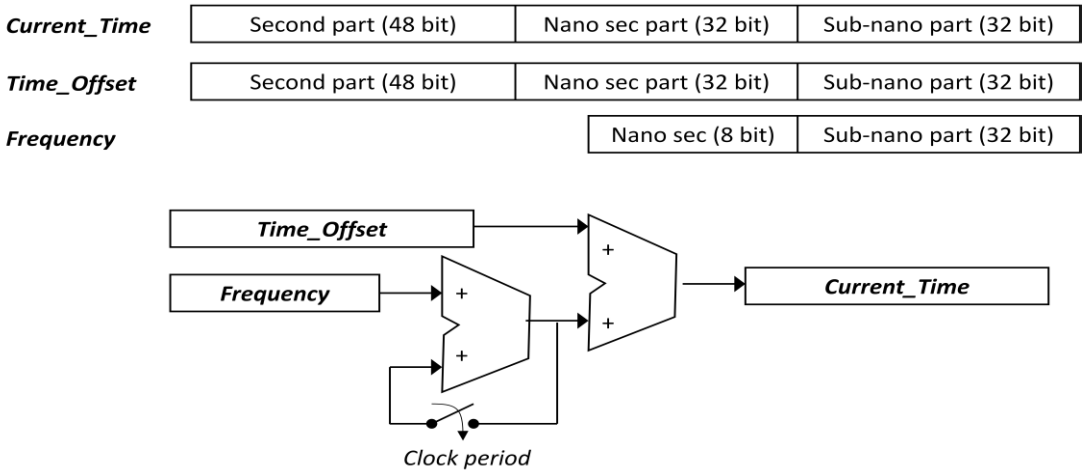

Fig. 7. Structure of the PTP slave clock.

redundant ports (A, B) were connected to small form-factor pluggable (SFP) modules and the IL port was linked to an RJ45 socket, which supports 1-Gbps Ethernet. The performance of the hardware design, focusing on the TC functionality, is summarized by the time uncertainty. The TC time uncertainty includes the inaccuracy in the computation of the path delay and residence time, as a network bridge; using (8), it is calculated as $3\sigma_{\varepsilon}^2$, where σ_{ε} is the standard deviation of the timestamp error. Fig. 9 shows the test setup for measuring the timestamp error. The design under test (DUT) receives sync messages from the IL port and forwards them to the redundancy ports (A, B). Let us assume T_{OSC} to be the time delay for a sync message passing from IL to B, measured by an oscilloscope and T_{COR} as the value of the *correctionField* in the sync message, after being updated in the DUT. Then, the difference in T_{COR} relative to T_{OSC}, defined as T_{Diff}, reflects the DUT timestamp error. As two timestamps (RX, TX) are involved in bridging each sync message, the variance of the time error caused by the DUT, σ_{TC}^2, is derived from T_{Diff} as in the following equation:

$$\sigma_{\mathrm{TC}}^2 = 1.5 \times \sigma_{T_{\mathrm{Diff}}}^2. \tag{9}$$

Fig. 10 shows the waveform of a sync message measured by an oscilloscope, Tektronix TDS 5104 with 5-GS/s sampling rate, probing on the low-voltage differential signal (LVDS) in the IL and B SFP modules. To detect the LVDS signals, rising edge of RX_DV of the gigabit media-independent interface (GMII) was used as a triggering point for IL. The GMII is used for connecting the FPGA with Rx/Tx PHYs of A, B, and IL, respectively. In Fig. 10, "RX_PHY," "Logic," and "TX_PHY" show the signal delays from IL to FPGA(in), from FPGA(in) to FPGA(out), and from FPGA(out)

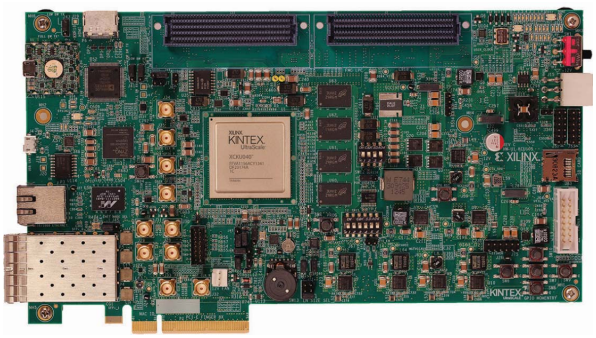

Fig. 8. Xilinx KCU 105 evaluation board.

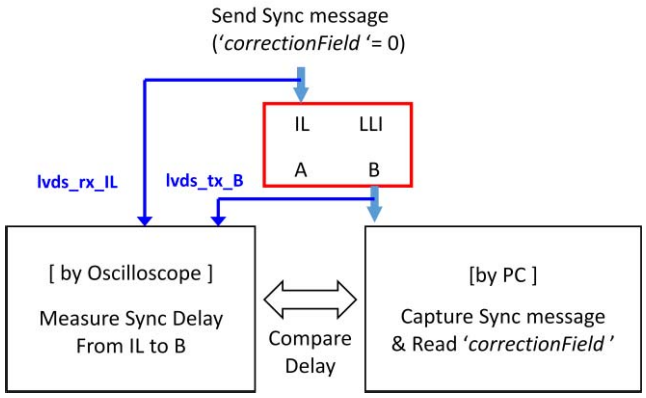

Fig. 9. Test setup for measuring the timestamp uncertainty of a TC/Slave.

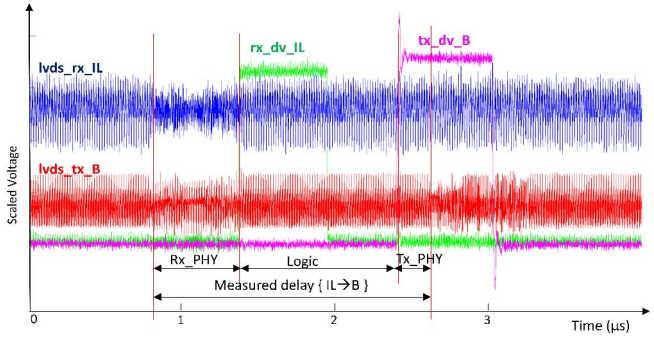

Fig. 10. Waveform of a sync message measured by an oscilloscope at the IL port (lvds_rx_IL) and port-B (lvds_tx_B). rx_dv_IL and tx_dv_B represent RX_DV/IL and TX_DV/B of GMII signal generated by FPGA, respectively.

TABLE I

TEST RESULTS OF THE MEASURED TC TIMESTAMP ERROR

MEASURED BY OSCILLOSCOPE		READ FROM *'CORRECTIONFIELD'* IN SYNC MESSAGE		DIFFERENCE	
MEAN	STD	MEAN	STD	MEAN	STD
1829	4.24	1829	0	0	4.24

Time unit: ns

to B, respectively. Table I summarizes the test results for a hundred sync messages. Finally, the TC time uncertainty, defined as $3 \times \sigma_{TC}$ in IEC/IEEE 61850-9-3 [8], is observed to be 15.6 ns, which is within 50 ns.

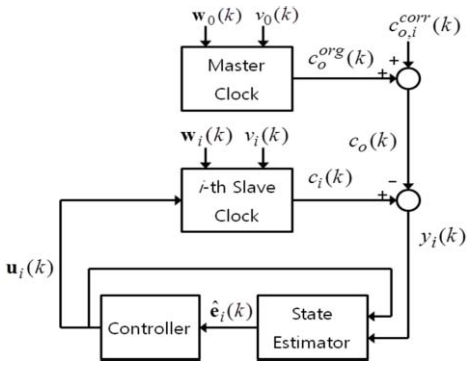

Fig. 11. Structure of the clock controller for a PTP slave.

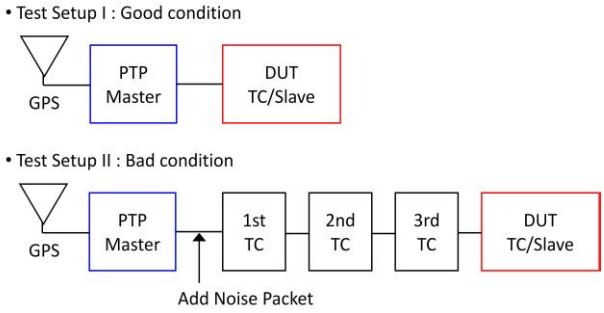

Fig. 12. Test setup to evaluate the performance of the TC/slave.

V. CLOCK CONTROLLER DESIGN

A. LQG Controller Design

For synchronization with the MC, the TC/slave controller generates signals to adjust the time and/or a frequency of the local clock, based on the state of the MC. In most cases, it is necessary to estimate the state of the MC because it is infected by noise. Fig. 11 shows the structure of a TC/slave controller. The controller output, $\mathbf{u}_i(k)$, is generated from the estimated state information, $\mathbf{e}_i(k)$. The state estimator or state observer separates the signals from the noisy measurements, $y_i(k)$. The controller design is based on the LQG, which is a combination of the Kalman filter as a state observer, along with an optimal linear quadratic regulator. The optimal state feedback gain, K_f and the state estimation gain, K_k are easily derived from the state equations of (1) and are well established in [21]–[23].

The software code for the controller was developed with *MicroBlaze* core, which is a 32-bit RISC processor implemented on a Xilinx FPGA, as shown in Fig. 8.

B. TC/Slave Testing With the LQG Controller

To evaluate the designed controller in a TC/Slave, two different setups were prepared, as depicted in Fig. 12. The first setup, with a TC/slave as the DUT and an MC directly connected to it, was prepared for a good condition test; the second setup comprising a master, three TCs, and a DUT was for a bad condition. In the second setup, for emulating the process bus in a substation, IEC 61850 *sampled value* packets, using 80% of the bandwidth, were added to the Ethernet line as noise between the MC and the first TC, as illustrated in Fig. 12. The MC used in the test is a Meinberg LANTIME M1000

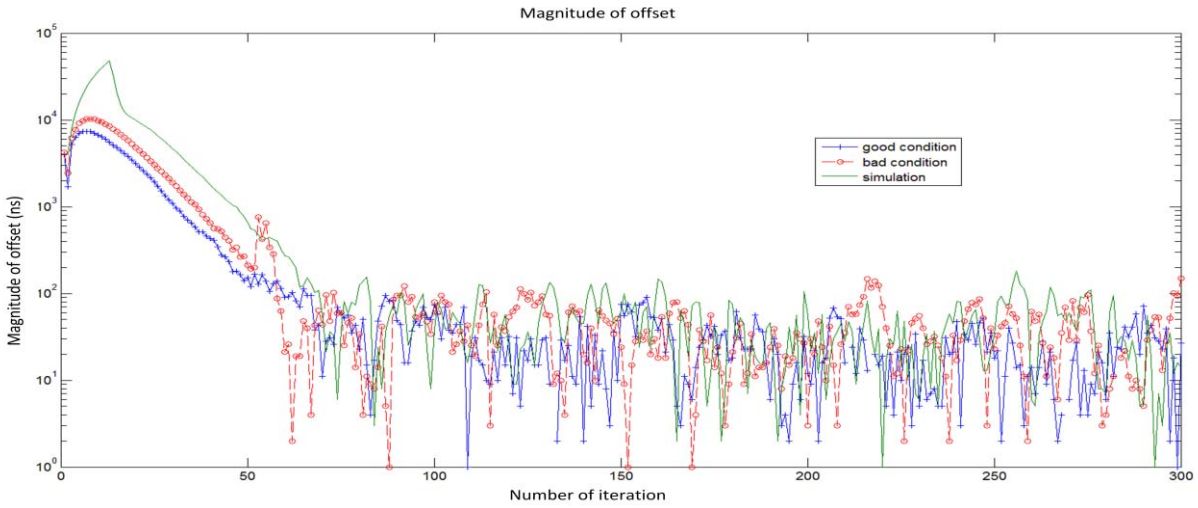

Fig. 13. Test plot: absolute value of time difference between master and DUT with LQG controller (blue line with "+" mark: good condition, red line with "o" mark: bad condition, green line: simulation).

TABLE II
SIMULATION SPECIFICATION

Parameter		Symbol	Value
Process noise Eq. (1)	Standard deviation of noise n_{ρ_i}	σ_{n_ρ}	5x10^{-9} s
	Standard deviation of noise n_{ρ_0}	$\sigma_{n_{\rho 0}}$	0
	Standard deviation of noise n_{τ_i}	σ_{n_τ}	25x10^{-9} s
	Standard deviation of noise n_{τ_0}	$\sigma_{n_{\tau 0}}$	0
	Sampling Period	ΔT	1s
Measurement noise Eq. (7)	Standard deviation of noise ε_i	σ_ε	4x10^{-9} s
	Standard deviation of noise ε_0	σ_ε	4x10^{-9} s
	Standard deviation of noise ε_p	σ_ε	4x10^{-9} s
	Standard deviation of noise ε_r	$\sqrt{2}\sigma_\varepsilon$	5.7x10^{-9} s
	# hop from master	I	3

TABLE III
TEST RESULT OF THE STEADY-STATE OFFSET

TEST SETUP	AVERAGE	STANDARD DEVIATION
GOOD CONDITION	-0.19	45.6
BAD CONDITION	-0.49	53.7
SIMULATION	-0.53	48.3

Time unit: ns

(intelligent, modular synchronization solution) and the TC is a Ruggedcom RSG2488 (network switch). All the PTP devices were configured as per the default settings of the power utility automation profile [8]. The test measurements were compared with the simulation results of an LQG controller modeled using (1). The simulation parameters are listed in Table II.

Three plots for the time differences, in absolute values, between the MC and DUT are displayed in Fig. 13; the blue line is for the good condition test, the red for the bad condition test, and the green for the simulation. The x-axis in the graph depicts the measurement sequence corresponding to the order of a sync message generated from the MC. The standard deviation of the steady-state error, σ_{SS}, is measured in 45.6 ns for a good condition, 53.7 ns for a bad condition, and in 48.3 ns by simulation, as listed in Table III. The transition time must be closely related to the initial condition of the clocks. For the experiment, a transition time exceeding $3 \times \sigma_{\mathrm{SS}}$ was measured in 55 s on an average, when the initial difference in frequency was approximately 4 us/s between the MC and TC/slave.

C. Design of the Modified LQG Controller

If a TC/Slave knows the initial states of the clocks, the transition time can be reduced, regardless of the initial conditions. The idea can be extended to a real system, where the ith TC/slave estimates the initial states, $\mathbf{e}_{i,\mathrm{INIT}}$, from a few measurements, N, before starting the LQG controller, as illustrated in Fig. 14. The least-squares approach attempts to minimize the squared difference between the assumed signal and the given measured data with no probabilistic assumptions, rendering it suitable for selection as an estimator. Moreover, it is necessary for the estimator to calculate the initial state, $\mathbf{e}_{i,\mathrm{INIT}}$, based on a limited number of past measurements because the algorithms are implemented on an embedded processor. The best candidate is an FIR filter called the receding horizon recursive state estimator [16]. From(1), we only consider the deterministic part as represented in (10). The aim is to obtain an estimate, $\mathbf{e}_{i,\mathrm{INIT}}$, using a finite number of $k+1$ measurements, $\{y_i(0), y_i(1) \ldots, y_i(k)\}$

$$\begin{aligned} \mathbf{e}_i(k+1) &= F\mathbf{e}_i(k) \\ y_i(k) &= H\mathbf{e}_i(k). \end{aligned} \tag{10}$$

Then, the result is computed in the following equation:

$$\mathbf{e}_{i,\mathrm{INIT}} = F^N \left(M_N^T M_N\right)^{-1} M_N^T Y_{k,N} \tag{11}$$

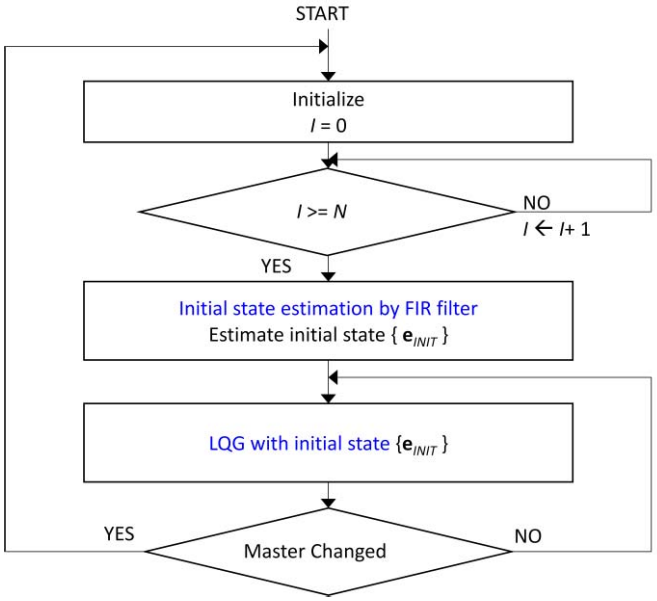

Fig. 14. Flowchart for the modified LQG controller.

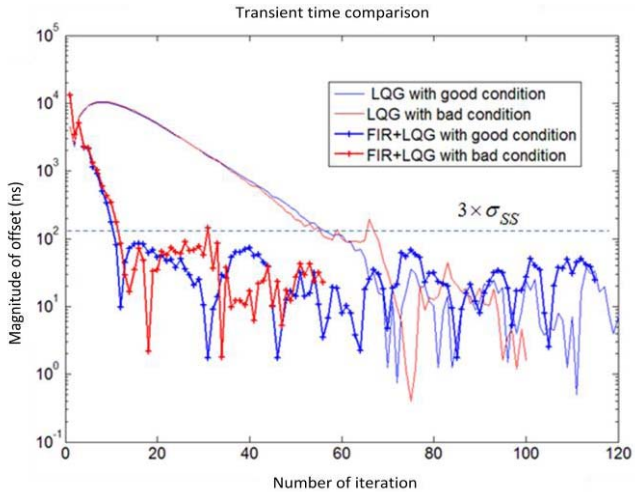

Fig. 15. Absolute value of the time difference between the master and DUT with the modified LQG controller after power-ON (blue line: LQG with a good condition, red line: LQG with a bad condition, blue line with "+" marks: FIR + LQG with a good condition, red line with "+" marks: FIR + LQG with a bad condition).

where

$$M_N = [(HF^N)^T, (HF^{N-1})^T, \ldots, H^T]^T$$
$$Y_{k,N} = [y_i(k), y_i(k-1), \ldots, y_i(k-N)]^T.$$

In (11), N is called the moving window or state estimation horizon, which can be selected as per to the designer's purpose [16]. Although (11) appears complicated, in general, it can be much simpler for fewer measurements or estimation horizon. Moreover, the term, $F^N(M_N^T M_N)^{-1} M_N^T$, can be predetermined, regardless of the measurements, for given values of k, N, H, and F.

As depicted in Fig. 14, initial state estimation is performed on the initial measurements, when a TC/slave is started or an MC is changed. The BMCA notifies the controller, when an MC is changed [7].

The time-offset plot shown in Fig. 15 demonstrates that the transition time with the modified LQG is considerably better than that of the original control. Here, the transition time is defined as a time interval exceeding $3 \times \sigma_{SS}$ (σ_{SS}: standard deviation of the steady state), from power-ON or on changing an MC. For the modified LQG controller, it also includes the time for estimating the initial state. The average transition time obtained from 100 trials for each setup is listed in Table IV, where $N = 2$ and $k = 3$ in (11). In the experiment, the initial frequency difference between the MC and TC/slave is observed to be approximately 4 μs/s. To analyze the relationship between the initial frequency offset and the transition time, simulation is performed with three different frequencies, 1, 4, and 10 μs/s. The results in Table V demonstrate that the convergence time of the FIR + LQG is faster and less sensitive to the initial conditions.

TABLE IV
TEST RESULTS OF THE AVERAGE TRANSITION TIME

MEASUREMENT SETUP		LQG	FIR+LQG
POWER ON	GOOD CONDITION	55	13
	BAD CONDITION	65	14
MASTER CHANGED	GOOD CONDITION	18	10
	BAD CONDITION	19	11

Time unit: s

TABLE V
SIMULATION RESULTS OF THE TRANSITION TIME

INITIAL FREQUENCY DIFFERENCE BETWEEN THE MASTER AND SLAVE	LQG		FIR+LQG	
	MEAN	STD	MEAN	STD
1 μs/s	54.6	7.1	12.4	1.2
4 μs/s	64.5	3.8	13	0.7
10 μs/s	73.5	3.1	14.8	1.8

Time unit: s

Compared to the commercial products, it shows better performance in transition time. Vendor A and Vendor B, as evaluated by Ingram *et al.* [24], require 35 s and 5 min to be stabilized within ± 1 μs, respectively. The standard deviations of steady-state offset are shown similar results which are roughly observed as 65 and 50 ns from Fig. 6 [24], respectively. In [10], the result show that approximately 40 s is required to reach the stable state under ± 1 μs, even though they showed excellence steady-state offset with standard deviations of few nanoseconds.

VI. CONCLUSION

For power utility automation, precise time reference is necessary for measuring the current/voltage levels, calculating the time differences between events, and for describing fast changing emergency situations.

A PTP TC/slave that supports seamless redundancy is described in this paper, focusing on the design architecture. The design objectives include functional integrity and performances based on the IEC/IEEE 61850-9-3, specifying the highest synchronization classes for power utility automation [8]. Implemented on the Xilinx KCU105 evaluation board,

the experimental results demonstrate that the three-sigma time uncertainty of the TC is 15.6 ns and that of the slave is 149 ns with an average transient time of 13.5 s, satisfying the standard.

REFERENCES

[1] *Industrial Communication Networks—High Availability Automation Networks—Part 3: Parallel Redundancy Protocol (PRP) and High-availability Seamless Redundancy (HSR)*, 3rd ed., IEC/IEEE Standard 62439-3, 2016.

[2] *Communication Networks and Systems for Power Utility Automation—Part 90-4: Network Engineering Guidelines*, 1st ed., document IEC/TR 61850-90-4, 2013.

[3] D. M. E. Ingram, P. Schaub, D. A. Campbell, and R. R. Taylor, "Performance analysis of PTP components for IEC 61850 process bus applications," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 4, pp. 710–719, Apr. 2013.

[4] D. C. Mazur, R. D. Quint, and V. A. Centeno, "Time synchronization of automation controllers for power applications," *IEEE Trans. Ind. Appl.*, vol. 50, no. 1, pp. 25–32, Jan. 2014.

[5] H. Guo and P. Crossley, "Design of a time synchronization system based on GPS and IEEE 1588 for transmission substations," *IEEE Trans. Power Del.*, vol. 32, no. 4, pp. 2091–2100, Aug. 2017.

[6] *Communication Networks and Systems for Power Utility Automation—Part 5: Communication Requirements for Functions and Device Models*, 2nd ed., IEC Standard 61850-5, 2013.

[7] *Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, 2nd ed., IEEE Standard IEC 61588/IEEE 1588, 2009.

[8] *Communication Networks and Systems for Power Utility Automation—Part 9-3: Precision Time Protocol Profile for Power Utility Automation*, 1nd ed., document IEC/IEEE 61850-9-3, 2016.

[9] J. Han and D.-K. Jeong, "A practical implementation of IEEE 1588-2008 transparent clock for distributed measurement and control systems," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 2, pp. 433–439, Feb. 2010.

[10] N. Moreira, J. Lázaro, U. Bidarte, J. Jimenez, and A. Astarloa, "On the utilization of system-on-chip platforms to achieve nanosecond synchronization accuracies in substation automation systems," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1932–1942, Jul. 2017.

[11] H. Kirrmann, C. Honegger, D. Ilie, and I. Sotiropoulos, "Performance of a full-hardware PTP implementation for an IEC 62439-3 redundant IEC 61850 substation automation network," in *Proc. ISPCS Int. IEEE Symp.*, Sep. 2012, pp. 1–6.

[12] G. Giorgi and C. Narduzzi, "Performance analysis of Kalman-filter-based clock synchronization in IEEE 1588 networks," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 8, pp. 2902–2909, Aug. 2011.

[13] P. Wolfrum, R. L. Scheiterer, and D. Obradovic, "An optimal control approach to clock synchronization," in *Proc. ISPCS Int. IEEE Symp.*, Sep./Oct. 2010, pp. 122–128.

[14] I.-C. Chao, K.-Y. Tu, S.-Y. Lin, and F.-R. Chang, "Design and implementation of a switching controller for transient improvement in a time synchronization system," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 7, pp. 2184–2190, Jul. 2011.

[15] J. Viejo, J. Juan, M. J. Bellido, A. Millan, and P. Ruiz-de-Clavijo, "Fast-convergence microsecond-accurate clock discipline algorithm for hardware implementation," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 12, pp. 3961–3963, Dec. 2011.

[16] K. V. Ling and K. W. Lim, "Receding horizon recursive state estimation," *IEEE Trans. Autom. Control*, vol. 44, no. 9, pp. 1750–1753, Sep. 1999.

[17] J. M. Pak, C. K. Ahn, Y. S. Shmaliy, and M. T. Lim, "Improving reliability of particle filter-based localization in wireless sensor networks via hybrid particle/FIR filtering," *IEEE Trans. Ind. Informat.*, vol. 11, no. 5, pp. 1089–1098, Oct. 2015.

[18] H. F. Ferro, L. F. O. Chamon, and C. G. Lopes, "FIR–IIR adaptive filters hybrid combination," *Electron. Lett.*, vol. 50, no. 7, pp. 501–503, Mar. 2014.

[19] J. Lin and M. J. T. Smith, "Two-band hybrid FIR–IIR filters for image compression," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3063–3072, Nov. 2011.

[20] D. Fontanelli and D. Macii, "Accurate time synchronization in PTP-based industrial networks with long linear path," in *Proc. IEEE Int. Symp. Precision Clock Synchronization Meas., Control Commun.*, Sep./Oct. 2010, pp. 97–102.

[21] M. Athans, "The role and use of the stochastic linear-quadratic-Gaussian problem in control system design," *IEEE Trans. Autom. Control*, vol. AC-16, no. 6, pp. 529–552, Dec. 1971.

[22] G. M. Siouris, *An Engineering Approach to Optimal Control and Estimation Theory*. Hoboken, NJ, USA: Wiley, 1996.

[23] F. L. Lewis and S. Ge, *Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2008.

[24] D. M. E. Ingram, P. Schaub, and D. A. Campbell, "Use of precision time protocol to synchronize sampled-value process buses," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 5, pp. 1173–1180, May 2012.

Soonwoo Lee received the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2002. He is currently pursuing the Ph.D. degree with the Korea University, Seoul, South Korea.

He has been with the Korea Electrotechnology Research Institute since 2005 and is currently a Senior Researcher with the Power Telecommunication Research Center. His current research interests include signal processing, digital control, and digital circuit design for power utility and smart grid applications.

Jimyung Kang received the B.S. and M.S. degrees in computer engineering from Seoul National University, Seoul, South Korea, in 2004 and 2006, respectively.

He has been with the Korea Electrotechnology Research Institute since 2006 and is currently a Senior Researcher with the Power Telecommunication Research Center. His current research interests include machine learning and communication in smart grid.

Sung Soo Choi (S'99–M'03) received the Ph.D. degree in electrical engineering from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2003.

He is currently a Principal Researcher with the Power Telecommunication Research Center, Korea Electrotechnology Research Institute, Ansan, South Korea. His current research interests include digital communication and digital circuits for power utility and smart grid applications.

Myo Taeg Lim (M'94) received the B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, South Korea in 1985 and 1987, respectively, the M.S. and Ph.D. degrees in electrical engineering from Rutgers University, NJ, USA, in 1990 and 1994, respectively.

He was a Senior Research Engineer, Samsung Advanced Institute of Technology, Suwon, South Korea, and an Assistant Professor with the Department of Control and Instrumentation, National Changwon University, Changwon, South Korea. Since 1996, he has been a Professor with the School of Electrical Engineering, Korea University, Seoul, South Korea. His current research interests include optimal and robust control, vision based motion control, and autonomous mobile robots. He has authored or co-authored over 80 journal papers and two books: *Optimal Control of Singularly Perturbed Linear Systems and Application*: *High-Accuracy Techniques, Control Engineering Series*, Marcel Dekker, New York, 2001; and *Optimal Control*: *Weakly Coupled Systems and Applications, Automation and Control Engineering Series*, CRC Press, New York, 2009.

Dr. Lim is a Fellow of the Institute of Control, Robotics and Systems, and a member of the IEEE and Korea Institute of Electrical Engineers. He serves as an Editor for *International Journal of Control, Automation, and Systems*.