Robust Time Synchronisation for Industrial Internet of Things by H_∞ Output Feedback Control

Yan Zong, *Member, IEEE,* Xuewu Dai, *Member, IEEE,* Zhuangkun Wei, Mengbang Zou, Weisi Guo, *Senior Member, IEEE,* and Zhiwei Gao, *Senior Member, IEEE*

***Abstract*—Precise timing over timestamped packet exchange communication is an enabling technology in the mission-critical industrial Internet of Things, particularly when satellite-based timing is unavailable. The main challenge is to ensure timing accuracy when the clock synchronisation system is subject to disturbances caused by the drifting frequency, time-varying delay, jitter, and timestamping uncertainty. In this work, a Robust Packet-Coupled Oscillators (R-PkCOs) protocol is proposed to reduce the effects of perturbations manifested in the drifting clock, timestamping uncertainty and delays. First, in the spanning tree clock topology, time synchronisation between an arbitrary pair of clocks is modelled as a state-space model, where clock states are coupled with each other by one-way timestamped packet exchange (referred to as packet coupling), and the impacts of both drifting frequency and delays are modelled as disturbances. A static output controller is adopted to adjust the drifting clock. The H_∞ robust control design solution is proposed to guarantee that the ratio between the modulus of synchronisation precision and the magnitude of the disturbances is always less than a given value. Therefore, the proposed time synchronisation protocol is robust against the disturbances, which means that the impacts of drifting frequency and delays on the synchronisation accuracy are limited. The one-hour experimental results demonstrate that the proposed R-PkCOs protocol can realise time synchronisation with the precision of six microseconds in a 21-node IEEE 802.15.4 network. This work has widespread impacts in the process automation of automotive, mining, oil and gas industries.**

***Index Terms*—Time synchronisation, packet-coupled oscillator, H_∞ control, pulse-coupled oscillators, wireless sensor networks.**

I. Introduction

OVER the last decade, the rapid proliferation of Internet of Things (IoT) has been instrumental in the digital manufacturing revolution (fourth industrial revolution), and a new era of the Industrial Internet of Things (IIoT) has emerged with different requirements to traditional IoT systems. Precise timing is one of the most sought after IIoT attributes in mission-critical industrial applications, especially those that have control loops commonly found in chemical engineering and precision manufacturing. This means that the time-sensitive wireless IIoT networks have stringent requirements on the reliability and the real-time of data transmission and control operation command [1], [2]. Hence, the enabling technology time synchronisation is required to provide a common sense of timing among wireless nodes.

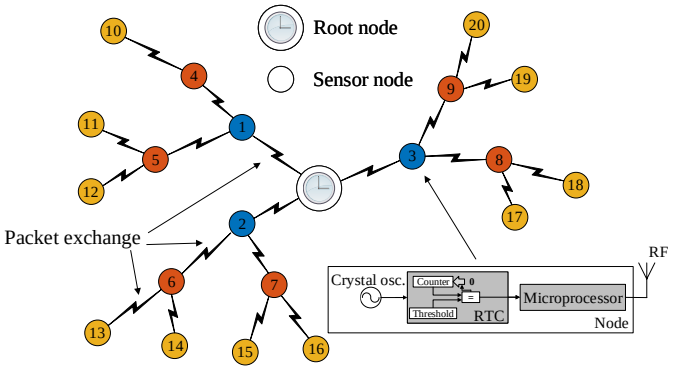


Fig. 1. Spanning tree clock synchronisation network.

Due to the inherent low energy consumption [3] and reliability [4] characteristics of spanning tree topology, it has been widely used for time synchronisation [5]. Also, inspired by the synchrony of fireflies' flashing [6], a typical model, Pulse-Coupled Oscillators (PCO), is proposed in natural and physical science communities [7]. Thanks to its simplicity and scalability, this model is particularly suitable for resource-constrained wireless sensor networks [8]. However, the assumptions of PCO [e.g. failure of producing the physical *Pulse* signal, and no delays exist during the firing information (i.e. *Pulse*) exchange among oscillators] limit its application in Off-the-Shelf wireless networks. Thus, it needs to be improved for employment in industrial applications.

In IEEE 802.15.4 (also known as Zigbee) [9] networks, the PCO's *Pulse* waveform cannot be generated from the Medium Access Control (MAC) layer. Nevertheless, the wireless packet can be treated as a substitute solution for the *Pulse* signal. Moreover, the periodic resetting feature of the clock model is similar to the firing-resetting procedure in PCO [10]. Therefore, our earlier work [11] proposed the Packet-Coupled Oscillators (PkCOs) model, where the *Sync* packet (from a transmitter) is utilised for reporting the firing information to other nodes. In this work, we utilise the H_∞ method for selecting the PkCOs [11] parameters. Thus, a Robust PkCOs (R-PkCOs) protocol is proposed for the spanning tree clock

Manuscript received October 31, 2021; accepted January 05, 2022. *(Corresponding author: Xuewu Dai.)*

Y. Zong, Z. Wei, M. Zou, and W. Guo are with the School of Aerospace, Transport and Manufacturing, Cranfield University, U.K. (e-mail: {y.zong, zhuangkun.wei, m.zou, weisi.guo}@cranfield.ac.uk).

X. Dai, and Z. Gao are with the Department of Mathematics, Physics and Electrical Engineering, Northumbria University, Newcastle upon Tyne, U.K. (e-mail: {xuewu.dai, zhiwei.gao}@northumbria.ac.uk).

Digital Object Identifier 10.1109/JIOT.2022.3144199

synchronisation network (e.g. Fig. 1[1]).

A. Related Work

As a result of the widespread importance of synchronisation, it has been studied in various communities, and many synchronisation protocols have been proposed for wireless networks. In the communication engineering community, from the perspective of packet exchange strategy, these algorithms can be categorised into two types, which are the receiver-to-receiver (e.g. RBS [12]) and sender-to-receiver[2] (e.g. TPSN [13], RMTS [14], PISync [15]) synchronisation protocols. The principle of these algorithms is to measure the clock offset (which is referred to as the time difference between two connected clocks) through packet exchange during each synchronisation cycle T; and the employment of the offset estimate to the local clock lets a network achieve time synchronisation.

For the RBS and TPSN algorithms, during a cycle T, several timestamped packets are sent and received between a pair of nodes. However, once these two protocols are employed to the large-scale wireless network, the offset estimate suffers from delay jitter, owing to factors, such as packet collisions and retransmission. Utilising inaccurate offset estimates reduces the synchronisation performance of RBS and TPSN in the multi-hop network. In addition, since the Radio Frequency (RF) transceiver is the most power consumption unit in a wireless node [16], frequent RF communication poses a challenge on the energy-constrained node.

In many mission-critical industrial applications, the slot-based contention-free packet transmission mechanism is used to guarantee that the packet exchange delay is almost deterministic, and thus to insure a high Quality of Service (QoS) [11], [17]. Thanks to this feature, instead of transmitting multiple wireless packets, the one-way sender-to-receiver protocol (also referred to as the flooding algorithm) only needs one packet to obtain a more accurate clock offset, thereby leading to better synchronisation precision. Moreover, two timestamps are required in the flooding algorithm; one is generated when a transmitter sends a packet, the other one is on the reception of the packet on a receiver.

The IEEE 802.15.4 standard provides the beacon-enabled operation on the MAC layer, and the corresponding superframe [consisting of Contention Access Period (CAP), Contention-Free Period (CFP) and Inactive Period] offers hybrid transmission mechanisms. Specifically, during CAP, all nodes need to contend for the access of a frequency channel. Instead, the contention-free period guarantees a specific slot to each node. The control packet (i.e. *Sync*) of R-PkCOs is sent in CFP to guarantee low-latency transmission and tiny jitter. Furthermore, the R-PkCOs protocol only demands one timestamp (which is generated upon the reception of a *Sync* packet), and the packet itself represents the clock firing information. This feature can further reduce the effects of timestamping uncertainty, improve offset estimate accuracy and enhance synchronisation performance, compared to the flooding protocol.

In addition to the packet exchange strategy, using advanced processing techniques (e.g. maximum likelihood estimation [18], and linear least squares regression [19]) is an alternative way to improve time synchronisation precision. Typically, the clock frequency difference may let the achieved synchronisation lose gradually [20]. The clock skew[3] correction method allows the longer synchronised state, and the less frequent re-synchronisation among coupled clocks.

In [14] and [18], the maximum likelihood estimation method is used to estimate clock skew, and also to obtain a more accurate clock offset; however, the resource-limited node (with a 32-bit microprocessor) has difficulty in handling such a complex processing technology. Thus, in [19], the estimation procedure of clock offset and skew via the linear least squares regression method is on the cluster head, which is equipped with a powerful processor, rather than on the local node. Even though [21] adopts the lower computational complexity solution (i.e. exponential moving average) to calculate the clock skew, the proposed synchronisation algorithm is still evaluated on FPGA-based wireless nodes. Moreover, [15] states that a proportional-integral controller is utilised in PISync, from (3) and (6) of this cited work, the used controlling strategy actually is a proportional controller. In this paper, a static output feedback controller is adopted for clock correction, and it demands fewer computational overhead, compared to the above processing methods (e.g. [18], [21]).

It is notable that, although the works cited above take the non-identical [14], [21] and drifting [15] clock frequency, and packet exchange delay [11], [14], [15], [21] into consideration, only the theoretical analysis of a synchronisation protocol is presented. In addition, the logical (or virtual) clock, which is an affine function of the physical clock in Fig. 2a, is used in the protocol analysis and hardware experiments [14], [15], [21]. Thus, the processing delay, which occurs during the data processing, and the employment of offset and skew estimates, is missing. Overall, there still exists a lack of theoretical design of synchronisation protocol parameters, with the effects of clock noises and external disturbances (from packet exchange and processing delays). This motivates us to use the H_∞ control solution for parameter selection of the R-PkCOs protocol, and extending our previous researches [8], [11].

B. Contributions and Paper Organisation

In this paper, we propose a robust PkCOs protocol to correct both clock skew and offset for improving synchronisation precision, subject to the impacts of drifting frequency, and external perturbations from delays. In addition to using the slot-based one-way *Sync* packet transmission mechanism, the H_∞ control method is also adopted to guide the R-PkCOs parameter selection, for letting clock and delay noises possess a small effect on the synchronisation accuracy. Specifically, through designing the static controller, the ratio between the

[1]We refer the reader to Section 2 for more details of Fig. 1.

[2]The sender-to-receiver synchronisation algorithm can be further classified into two kinds, namely, the one-way (e.g. RMTS, PISync) and two-way (e.g. TPSN) exchange protocols.

[3]The skew is defined as the normalised difference between two clock frequencies, see (3) of Section 2.

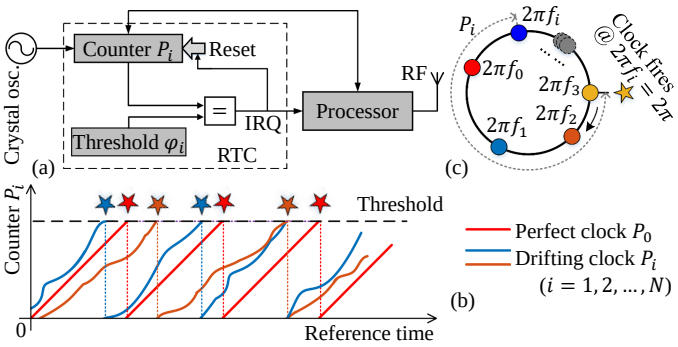


Fig. 2. (a) Structure of a real-time clock module. (b) Dynamics of the counter register in the RTC module. (c) Packet-Coupled Oscillators.

modulus of the achieved synchronisation precision and the magnitude of the noises is always less than a given value. Thus, the robustness of R-PkCOs is guaranteed, in the presence of internal clock and external delay noises. The one-hour experimental results show that the proposed R-PkCOs protocol is capable of achieving time synchronisation with the precision of 6 microseconds in a spanning tree clock network.

The rest of this paper is organised as follows: Section 2 describes the R-PkCOs model, which consists of the mathematical modelling of a non-identical and drifting embedded clock, and the packet-coupled synchronisation scheme. Then, Section 3 presents the H_∞ output feedback control for the R-PkCOs time synchronisation method. The simulation and experimental results are, respectively, shown in Sections 4 and 5. Eventually, Section 6 concludes this work.

II. Robust Packet-Coupled Oscillators

The spanning tree clock synchronisation network, for example, in Fig. 1, can be described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$[4], where $\mathcal{V} = \{0, 1, ..., N\}$ denotes a set of nodes, and a set of edges $\mathcal{E}$ induced by the adjacency matrix $\mathcal{A}$. The network is composed of a root node (i.e. $i = 0$) and a set of sensor nodes represented by $\mathcal{N} = \{i : i \in \mathcal{V},\ i \geq 1\}$. The root node is unique, and is equipped with a Global Positioning System (GPS) clock to provide the reference time to all the sensor nodes in a network. For the i-th sensor node, it corrects the local clock, upon the reception of a *Sync* packet from the parent node.

A. Modelling of Drifting Embedded Clocks

In the embedded system, the Real-Time Clock (RTC) module of each node is implemented by a counter register, which is driven by a crystal oscillator (see Fig. 2a). As shown in Fig. 2b, once counter reaches the pre-defined value in the threshold register, it is reset and counts from zero again; in the meantime, an interrupt (i.e. IRQ) signal is sent to the processor for triggering an event (e.g. sending a wireless packet for synchronisation purposes). This periodic resetting feature can be modelled as an oscillator running on the unit circle (see Fig. 2c). Once the oscillator's time variable P_i reaches the threshold, the oscillator fires (i.e. a *Sync* packet is generated and transmitted in CFP) and P_i is reset to zero, after which it increases again. Note that since *Sync* itself contains the clock firing information, there is no need to generate a timestamp when sending the packet. This helps R-PkCOs reduce the effects of timestamping uncertainty.

Referring first to the case of an ideal embedded clock on the root node, the time variable $P_0[n]$ can be utilised to model the clock's periodic resetting behaviour, and $P_0[n]$ at the n-th event satisfies the following form

$$P_0[n] = n\tau_0 - \sum_{h=0}^{k} \varphi_0[h], \tag{1}$$

where τ_0 is the nominal (clock update) period, and the nominal frequency f_0 of the perfect clock is equal to $f_0 = 1/\tau_0$. $\varphi_0[k]$ is the clock's threshold; in practice, $\varphi_0[k]$ may be a constant value, which equals the time synchronisation cycle T. k is the number of clock resetting from $n = 0$ to the n-th event, and it also represents that the clock is at the k-th synchronisation cycle. In addition, we assume that the perfect clock updates m_0 times in a cycle T (i.e. $\varphi_0 = m_0\tau_0$) [11]. Thus, k can be obtained from the floor function $k = \lfloor n/m_0 \rfloor$.

However, due to the manufacturing tolerance and environmental temperature, the i-th clock's time variable $P_i[n]$ cannot be the same as $P_0[n]$ of the perfect clock. Through modelling the random noise from the phase variation $\phi_i[n]/2\pi f_0$ and the clock frequency deviation $\chi_i[n] = f_i[n] - f_0$ [10], $P_i[n]$ is

$$P_i[n] = n\tau_0 + \frac{\sum_{h=0}^{n-1} \chi_i[h]\tau_0}{f_0} + \frac{\phi_i[n]}{2\pi f_0} - \sum_{h=0}^{k} \varphi_i[h], \tag{2}$$

where $\varphi_i[k]$ is the i-th clock threshold. $\varphi_i[k]$ is assumed to equal $\varphi_0[k]$ during the clock modelling.

Let the clock offset $\theta_i[n]$ represent the difference between $P_i[n]$ and $P_0[n]$. The clock skew $\gamma_i[n]$ is referred to as the normalised difference between $f_i[n]$ and f_0. Hence, $\theta_i[n]$ and $\gamma_i[n]$ are, respectively, given by

$$\theta_i[n] = P_i[n] - P_0[n], \quad \gamma_i[n] = \tfrac{\chi_i[n]}{f_0}. \tag{3}$$

By substituting (1) and (2) into (3), calculating the offset difference between two consecutive clock events, and expanding the clock offset and skew from n-dimension into k-dimension [22], the drifting embedded clock (2) is re-modelled as

$$\begin{cases} \theta_i[k+1] = \theta_i[k] + \gamma_i[k]T + \omega_{\theta_i}[k] \\ \gamma_i[k+1] = \gamma_i[k] + \omega_{\gamma_i}[k] \end{cases}, \tag{4}$$

where $\omega_{\theta_i}[k]$ and $\omega_{\gamma_i}[k]$ are the Gaussian random noise processes, and the corresponding variances are $\sigma^2_{\theta_i}$ and $\sigma^2_{\gamma_i}$ [23]. The matrix form of (4) is also obtained:

$$x_i[k+1] = Ax_i[k] + \omega_i[k], \tag{5}$$

where $x_i[k] = [\theta_i[k], \gamma_i[k]]^T$ represents the i-th clock state vector. $\omega_i[k] = [\omega_{\theta_i}[k], \omega_{\gamma_i}[k]]^T$ is the i-th clock noise process vector. The matrix A is equal to $A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$.

[4]To guarantee high performance, the node with a more accurate timing (e.g. the root node in Fig. 1) drives the cascade synchronisation, and no reverse driving exists.

B. Packet-Coupled Synchronisation Scheme

In order to reduce the effects of packet exchange delay jitter on the synchronisation precision, this work allocates the *Sync* packet transmission event to a specific time slot (in the contention-free period) for synchronising drifting embedded clocks. To be specific, at the k-th time synchronisation cycle, upon the reception of a *Sync* packet (which is from node j, and is transmitted at the time slot t_{k_j}) after the packet exchange delay $\kappa_{ij}[k]$, node i generates a timestamp $\hat{P}_i[k]$ via reading the counter register:

$$\hat{P}_i[k] = P_i(t_{k_j} + \kappa_{ij}[k]), \tag{6}$$

where $\kappa_{ij}[k]$ is the Gaussian random process with the mean of $\bar{\kappa}_{ij}$ and the variance of $\sigma^2_{\kappa_{ij}}$ [11], [14].

Next, the utilisation of the local timestamp $\hat{P}_i[k]$ can calculate the offset estimate $\hat{\theta}_i[k]$, following

$$\hat{\theta}_i[k] = \theta_i(t_{k_j} + \kappa_{ij}[k]). \tag{7}$$

In the meantime, the skew estimate is obtained from $\hat{\gamma}_i[k] = (\hat{\theta}_i[k] - \hat{\theta}_i[k-1]^+)/T$, where $\hat{\theta}_i[k-1]^+$ is the offset after the i-th embedded clock is adjusted at the $(k-1)$-th cycle. Even though $\hat{\theta}_i[k-1]^+$ is unknown, the clock offset approaches to zero at synchronised state. Thus, we assume that the clock offset is perfectly corrected, and $\hat{\theta}_i[k-1]^+$ is zero. The skew estimate $\hat{\gamma}_i[k]$ is calculated from

$$\hat{\gamma}_i[k] = \frac{\hat{\theta}_i[k]}{T}. \tag{8}$$

Instead of employing the complete offset and skew estimates to a drifting clock, this paper utilises a static controller to improve synchronisation performance, yielding

$$\begin{cases} u_{\theta_i}[k] = \alpha(r_{\theta_i} - \hat{\theta}_i[k]) \\ u_{\gamma_i}[k] = \beta(r_{\gamma_i} - \hat{\gamma}_i[k]) \end{cases}, \tag{9}$$

where $u_{\theta_i}[k]$ and $u_{\gamma_i}[k]$ are the offset and skew correction inputs respectively. r_{θ_i} and r_{γ_i} are, respectively, the offset and skew reference inputs. α and β are the controller's parameters.

Practically, due to limitations of the processor architecture, the processing delay $\eta_i[k]$ occurs, when the offset correction input $u_{\theta_i}[k]$ is applied to the counter register [8]. The clock's time variable actually is adjusted at the time $t_{k_j}+\kappa_{ij}[k]+\eta_i[k]$:

$$P_i[k]^+ = P_i(t_{k_j} + \kappa_{ij}[k] + \eta_i[k]) + (u_{\theta_i}[k] + \bar{\eta}_i), \tag{10}$$

where the processing delay $\eta_i[k]$ is the Gaussian random process with the mean of $\bar{\eta}_i$ and the variance of $\sigma^2_{\eta_i}$. The extra value of processing delay is unintentionally employed to correct the local clock, and the effects of timestamping uncertainty are modelled in $\eta_i[k]$ [11]. This work compensates for the impacts of this processing delay via adding its mean value to $u_{\theta_i}[k]$, as shown in [8].

From (3), the employment of offset correction input $u_{\theta_i}[k]$ is equivalent to the clock correction action on $\theta_i[k]$. That is

$$\theta_i[k]^+ = \theta_i(t_{k_j} + \kappa_{ij}[k] + \eta_i[k]) + (u_{\theta_i}[k] + \bar{\eta}_i). \tag{11}$$

To correct the clock skew, the following expression is used:

$$\gamma_i[k]^+ = \gamma_i(t_{k_j}) + u_{\gamma_i}[k]. \tag{12}$$

Remark 1. The packet exchange delay $\kappa_{ij}[k]$ is almost deterministic, owing to the slot-based *Sync* packet transmission in the contention-free period. The employment of $\bar{\kappa}_{ij}$ eliminates the effects of $\kappa_{ij}[k]$ [11]. Thus, in the experiments, $\hat{\theta}_i[k]$ of (9) is calculated from

$$\hat{\theta}_i[k] = \begin{cases} \hat{P}_i[k] - \bar{\kappa}_{ij} + \Delta t_{d_{ij}} & \text{if } \hat{P}_i[k] - \bar{\kappa}_{ij} + \Delta t_{d_{ij}} < \frac{\varphi_i[k]}{2} \\ \hat{P}_i[k] - \bar{\kappa}_{ij} + \Delta t_{d_{ij}} - \varphi_i[k] & \text{if } \hat{P}_i[k] - \bar{\kappa}_{ij} + \Delta t_{d_{ij}} \geq \frac{\varphi_i[k]}{2} \end{cases}, \tag{13}$$

where $\Delta t_{d_{ij}} = t_{d_i} - t_{d_j}$ is the difference of anti-phase synchronisation duration between node i and j, and the anti-phase synchronisation duration t_{d_i} is defined by $t_{d_i} = \begin{cases} 0 & \text{if } i = 0 \\ t_{dp} + (i-1)t_{sd} & \text{if } i \geq 1 \end{cases}$, t_{dp} means the contention access period, and the application data stream can be sent during this CAP. t_{sd} represents the slot duration.

For the purpose of realising collision-free packet transmission in CFP, r_{θ_i} and r_{γ_i} are, respectively, set to $-\Delta t_{d_{ij}}$ and 0 to allocate the slot t_{d_i} for node i. Once a network system is at steady synchronised state, the i-th clock offset $\theta_i[k]$ approaches t_{d_i} to realise the scheduling of wireless *Sync* packets, and $\gamma_i[k]$ converges to zero to achieve synchronisation of drifting clocks. This *Sync* scheduling solution can help decrease packet exchange delay jitter, thereby improving synchronisation precision.

Remark 2. Due to the difficulty of adjusting embedded clock frequency, in the experiments, the clock threshold correction is utilised as a substitute approach for the frequency adjustment, yielding

$$\varphi_i[k+1] = \varphi_i[k] + u_{\varphi_i}[k], \tag{14}$$

where the threshold correction value $u_{\varphi_i}[k]$ is equal to $u_{\varphi_i}[k] = -\beta(r_{\gamma_i} - \hat{\theta}_i[k])$.

III. Robust Output Feedback Controller

Apart from reducing packet exchange delay jitter via the slot-based transmission mechanism, we also adopt the H_∞ control solution to let clock and delay noises possess a small impact on the accuracy, which further improves synchronisation performance. This section starts by presenting the state space representation of a static output feedback controller. Then, the H_∞ control is utilised to design the R-PkCOs parameters, thus guaranteeing the robustness of the proposed method in a spanning tree clock synchronisation network.

A. Output Feedback Controller in State Space

In contrast to the conventional continuous control system with delays, delays play a different role in the discrete time synchronisation system. This means that the impacts of packet exchange and processing delays of the synchronisation system can be decoupled from the temporal dimension, and are represented as biases or disturbances in the variable P_i dimension [11]. The effects of packet exchange delay in the

temporal dimension can be removed by subtracting $\bar{\kappa}_{ij}$ from the timestamp $\hat{P}_i[k]$ [8]. Thus, (7) and (8) are re-written as

$$\begin{cases} \hat{\theta}_i[k] = \theta_i[k] + \nu_{\theta_i}[k] \\ \hat{\gamma}_i[k] = \gamma_i[k] + \nu_{\gamma_i}[k] \end{cases}, \tag{15}$$

where $\nu_{\theta_i}[k] = \kappa_{ij}[k] + \delta_{\kappa_{ij}}[k] - \bar{\kappa}_{ij}$ is the offset measurement noise with the mean of $\bar{\nu}_{\theta_i}$ and the standard deviation of $\sigma_{\nu_{\theta_i}}$. $\nu_{\gamma_i}[k] = (\kappa_{ij}[k] + \delta_{\kappa_{ij}}[k] - \bar{\kappa}_{ij})/T$ is the clock skew measurement noise with the mean of $\bar{\nu}_{\gamma_i}$ and the standard deviation of $\sigma_{\nu_{\gamma_i}}$. $\delta_{\kappa_{ij}}[k]$ and $\delta_{\eta_i}[k]$ are the extra offset values, which are the joint impacts of clock skew and the length of corresponding (packet exchange and processing) delays. Let $y_i[k] = [\hat{\theta}_i[k], \hat{\gamma}_i[k]]^T$, $\nu_i[k] = [\nu_{\theta_i}[k], \nu_{\gamma_i}[k]]^T$, according to (15), the matrix-vector measurement equation is obtained:

$$y_i[k] = C_2 x_i[k] + \nu_i[k], \tag{16}$$

where $y_i[k]$ is the clock output vector. $\nu_i[k]$ is the measurement noise vector. C_2 is a 2×2 identity matrix.

Likewise, (9) is also modified to the following form via defining the control vector $u_i[k] = [u_{\theta_i}[k], u_{\gamma_i}[k]]^T$:

$$u_i[k] = K(r_i - y_i[k]), \tag{17}$$

where $r_i = [r_{\theta_i}, r_{\gamma_i}]^T$ is a reference input matrix. The gain matrix is equal to $K = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}$.

For the purposes of theoretical study, (11) and (12) are re-written as

$$x_i[k]^+ = x_i[k]^- + u_i[k] - \digamma_i[k], \tag{18}$$

where $\digamma_i[k] = [(\eta_i[k] + \delta_{\eta_i}[k]) - \bar{\eta}_i, 0]^T$ is the processing delay noise vector [11].

Through applying $u_i[k]$ to the i-th embedded clock model (5), it is modified to

$$x_i[k+1] = Ax_i[k] + Bu_i[k] + Ed_i[k], \tag{19}$$

where the disturbance vector $d_i[k] = [\omega_i^T[k], \nu_{ij}^T[k], (\eta_i[k] + \delta_{\eta_i}[k]) - \bar{\eta}_i]^T$ consists of internal clock noises and external perturbations (from packet exchange and processing delays). B is a 2×2 identity matrix. The matrix E is equal to $E = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$.

Eventually, the pairwise output feedback control synchronisation system is given by

$$\begin{cases} x_i[k+1] = Ax_i[k] + Bu_i[k] + Ed_i[k] \\ o_i[k] = C_1 x_i[k] + Fd_i[k] \\ y_i[k] = C_2 x_i[k] + Hd_i[k] \\ u_i[k] = K(r_i - y_i[k]) \end{cases}, \tag{20}$$

where $o_i[k]$ is the performance output vector. The matrices C_1, F and H are, respectively, equal to

$$C_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}, \; F = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \\ H = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

As shown in (20), time synchronisation is described as a state-space model, whose output is synchronisation precision $o_i[k]$. In addition, this model is also disturbed by clock and delay noises $d_i[k]$. The purpose of the H_∞ control is to let $d_i[k]$ possess a tiny impact on the output accuracy $o_i[k]$. In other words, by designing the static controller, the ratio between the modulus of the achieved synchronisation precision and the magnitude of the noises is always less than a given value.

B. Controller Optimisation

Here, we propose a design condition to guarantee that the networked system is robust in the presence of disturbances (i.e. $d_i[k]$), caused by the drifting clock, and packet-exchange and processing delays. Mathematically speaking, $G[z] = C_1(zI - A)^{-1}E + F$ is the transfer function of (20) relating $d_i[k]$ to $o_i[k]$. The performance H_∞ of (20) is guaranteed (i.e. $\sup_{\|d_i\|_2 \le 1} \|o_i\|_2 < \rho$), if the infinity norm $\|G[z]\|_\infty$, equalling the two-norm ratio between $o_i[k]$ and disturbances $d_i[k]$, is less than ρ. That is

$$\begin{aligned} \|G[z]\|_\infty &= \sup_{\|d_i\|_2 \le 1} \|o_i\|_2 \\ &= \frac{\|o_i\|_2}{\|d_i\|_2} < \rho. \end{aligned} \tag{21}$$

Let r_i be a 2×1 zero matrix, and the closed-loop system (20) is modified to the following form

$$\begin{bmatrix} x_i[k+1] \\ o_i[k] \end{bmatrix} = (\mathbf{A} + \mathbf{BKC}) \begin{bmatrix} x_i[k] \\ d_i[k] \end{bmatrix}, \tag{22}$$

where the matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{K}$ and $\mathbf{C}$ are, respectively, equal to

$$\mathbf{A} = \begin{bmatrix} A & E \\ C_1 & F \end{bmatrix}, \mathbf{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \mathbf{K} = -K, \mathbf{C} = \begin{bmatrix} C_2 & H \end{bmatrix} \tag{23}$$

Before carrying out the main work, we introduce the following preliminary lemma.

Lemma 1. [24] For the square matrices X and S, and the matrices $\mathrm{T} = \mathrm{T}^T$, A, P, L with appropriate dimensions, the following two inequalities are equivalent:

$$\begin{bmatrix} \mathrm{T} + (\mathrm{LA}) + (\mathrm{LA})^T & * \\ \mathrm{XP}^T - \mathrm{XL}^T + \mathrm{SA} & -\mathrm{SX}^T - \mathrm{XS}^T \end{bmatrix} < 0 \tag{24}$$

$$\mathrm{T} + (\mathrm{PA}) + (\mathrm{PA})^T < 0. \tag{25}$$

Proof. (24) $\Rightarrow$ (25): From (24), the inequality $-\mathrm{SX}^T - \mathrm{XS}^T < 0$ is obtained. This means that X is a nonsingular matrix. Through pre- and post-multiplying (24) with $[I, \mathrm{A}^T\mathrm{X}^{-1}]$ and its transpose, we have

$$\begin{aligned} &\begin{bmatrix} I & \mathrm{A}^T\mathrm{X}^{-1} \end{bmatrix} \begin{bmatrix} \mathrm{T} + (\mathrm{LA}) + (\mathrm{LA})^T & * \\ \mathrm{XP}^T - \mathrm{XL}^T + \mathrm{SA} & -\mathrm{SX}^T - \mathrm{XS}^T \end{bmatrix} \begin{bmatrix} I \\ \mathrm{X}^{-T}\mathrm{A} \end{bmatrix} \\ &= \begin{bmatrix} \left(\mathrm{T} + (\mathrm{LA}) + (\mathrm{LA})^T + \mathrm{A}^T\mathrm{X}^{-1}(\mathrm{XP}^T - \mathrm{XL}^T + \mathrm{SA})\right)^T \\ \left((\mathrm{XP}^T - \mathrm{XL}^T + \mathrm{SA})^T + \mathrm{A}^T\mathrm{X}^{-1}(-\mathrm{SX}^T - \mathrm{XS}^T)\right)^T \end{bmatrix}^T \begin{bmatrix} I \\ \mathrm{X}^{-T}\mathrm{A} \end{bmatrix} \\ &= \begin{bmatrix} \left(\mathrm{T} + (\mathrm{LA}) + (\mathrm{LA})^T + \mathrm{A}^T\mathrm{P}^T - \mathrm{A}^T\mathrm{L}^T + \mathrm{A}^T\mathrm{X}^{-1}\mathrm{SA}\right)^T \\ \left((\mathrm{XP}^T - \mathrm{XL}^T + \mathrm{SA})^T - \mathrm{A}^T\mathrm{X}^{-1}\mathrm{SX}^T - \mathrm{A}^T\mathrm{S}^T\right)^T \end{bmatrix}^T \begin{bmatrix} I \\ \mathrm{X}^{-T}\mathrm{A} \end{bmatrix} \\ &= \mathrm{T} + (\mathrm{LA}) + (\mathrm{LA})^T + \mathrm{A}^T\mathrm{P}^T - \mathrm{A}^T\mathrm{L}^T + \mathrm{A}^T\mathrm{X}^{-1}\mathrm{SA} \\ &\quad + \left((\mathrm{XP}^T - \mathrm{XL}^T + \mathrm{SA})^T - \mathrm{A}^T\mathrm{X}^{-1}\mathrm{SX}^T - \mathrm{A}^T\mathrm{S}^T\right)^T \left(\mathrm{X}^{-T}\mathrm{A}\right) \\ &= \mathrm{T} + (\mathrm{PA}) + (\mathrm{PA})^T < 0. \end{aligned} \tag{26}$$

Hence, (25) is obtained.

(25) $\Rightarrow$ (24): Let L = P, S = I, and X = $\varkappa I$ where the scalar $\varkappa > 0$, the matrix inequality (24) is modified to

$$\begin{bmatrix} \mathrm{T} + (\mathrm{PA}) + (\mathrm{PA})^T & A^T \\ \mathrm{A} & -2\varkappa I \end{bmatrix} < 0. \tag{27}$$

Based on the Schur complement, (27) is re-written as

$$\mathrm{T} + (\mathrm{PA}) + (\mathrm{PA})^T + \frac{1}{2\varkappa}\mathrm{A}^T\mathrm{A} < 0. \tag{28}$$

Since $\mathrm{T}+(\mathrm{PA})+(\mathrm{PA})^T < 0$ holds, the sufficient large number $\varkappa > 0$ guarantees that the above inequality (28) is true. □

Theorem 1. *Given a spanning tree clock synchronisation network denoted by $\mathcal{G}$, consisting of a perfect root node's clock and N sensor node clocks with non-identical and drifting frequencies $f_i[k] \in \{f_i[k] : f_i[k] \neq f_0$ and $i \in \mathcal{N}\}$, and a scalar $\rho > 0$. For the known parameters ζ and $\xi \neq 0$, if there exist the matrices $Q > 0 \in \mathbb{R}^{2\times 2}$ and $G \in \mathbb{R}^{3\times 3}$, and the diagonal matrices $V \in \mathbb{R}^{2\times 2}$ and $U \in \mathbb{R}^{2\times 2}$ such that*

$$\begin{bmatrix} \Phi_1 & * & * \\ \Psi_1 & \Psi_2 & * \\ \Phi_2 & (\xi G\mathbf{B} - \mathbf{B}U)^T & -(\mathbf{B}^T\mathbf{B}U) - (\mathbf{B}^T\mathbf{B}U)^T \end{bmatrix} < 0 \tag{29}$$

where $\Phi_1 = -diag(Q, \rho^2 I) + (\zeta H\mathbf{B}V\mathbf{C}) + (\zeta H\mathbf{B}V\mathbf{C})^T$, $\Phi_2 = (\mathbf{B}^T\mathbf{B}V\mathbf{C}) - (\zeta H\mathbf{B}U)^T$, $\Psi_1 = G\mathbf{A} + \mathbf{B}V\mathbf{C}$, $\Psi_2 = -G - G^T + diag(Q, I)$, $H = [I \in \mathbb{R}^{3\times 3}, 0 \in \mathbb{R}^{3\times 3}]^T$, and the control gain matrix is $\mathbf{K} = \xi U^{-1}V$, then the prescribed H_∞ performance (21) is guaranteed.

Proof. The directed spanning tree system $\mathcal{G}$ can be decomposed into N two-dimensional systems (22). For an arbitrary closed-loop pairwise system, suppose that (29) holds, $-(\mathbf{B}^T\mathbf{B}U) - (\mathbf{B}^T\mathbf{B}U)^T < 0$ implies that U is a nonsingular matrix. By defining $U = \xi\mathbf{U}$, and letting the matrices T, L, A, P, S and X in Lemma 1 equal to

$$\begin{aligned} \mathrm{T} &= \begin{bmatrix} -diag(Q, \rho^2 I) & \Psi_1^T \\ \Psi_1 & \Psi_2 \end{bmatrix}, \\ \mathrm{L} &= \begin{bmatrix} (\zeta H\mathbf{B}\mathbf{U})^T & 0 \end{bmatrix}^T, \mathrm{A} = \mathbf{U}^{-1}V \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix}, \\ \mathrm{P} &= \begin{bmatrix} 0 & (G\mathbf{B} - \mathbf{B}\mathbf{U})^T \end{bmatrix}^T, \mathrm{S} = \mathbf{B}^T\mathbf{B}\mathbf{U}, \mathrm{X} = \xi I. \end{aligned}$$

From (25) of Lemma 1, the following matrix inequality is obtained:

$$\begin{aligned} &\begin{bmatrix} -diag(Q, \rho^2 I) & \Psi_1^T \\ \Psi_1 & \Psi_2 \end{bmatrix} + \begin{bmatrix} 0 \\ G\mathbf{B} - \mathbf{B}\mathbf{U} \end{bmatrix} \mathbf{U}^{-1}V \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \\ &+ \left(\begin{bmatrix} 0 \\ G\mathbf{B} - \mathbf{B}\mathbf{U} \end{bmatrix} \mathbf{U}^{-1}V \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \right)^T \\ &= \begin{bmatrix} -diag(Q, \rho^2 I) & \Psi_1^T \\ \Psi_1 & \Psi_2 \end{bmatrix} \\ &+ \left(\begin{bmatrix} 0 & I \end{bmatrix}^T (G\mathbf{B} - \mathbf{B}\mathbf{U}) \mathbf{U}^{-1}V \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \right) \\ &+ \left(\begin{bmatrix} 0 & I \end{bmatrix}^T (G\mathbf{B} - \mathbf{B}\mathbf{U}) \mathbf{U}^{-1}V \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \right)^T < 0. \end{aligned} \tag{30}$$

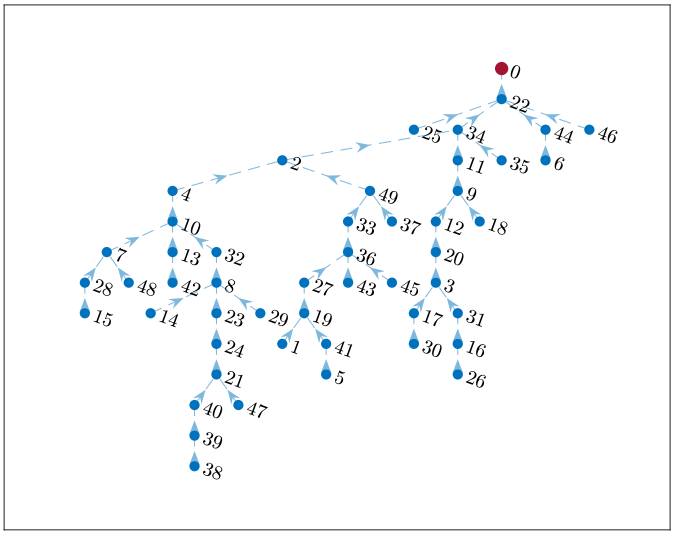


Fig. 3. 50-node spanning tree clock synchronisation network.

Algorithm 1 R-PkCOs Synchronisation Protocol

Initialisation
  configure parameters; $k = 0$;
  initialise RF;
  initialise RTC;
IRQ: Clock Firing
  reset counter (counter = 0);
  $send\_pkt(Sync)$;
AMI: Reception of A *Sync* Packet
  read counter ($\hat{P}_i[k]$ = counter);
  estimate clock offset, according to (13);
  adjust clock threshold, based on (14):
    threshold = threshold + $u_{\varphi_i}[k]$;
  correct counter, following:
    if $(\hat{P}_i[k] + u_{\theta_i}[k] + \bar{\eta}_i) <$ threshold **then**
      counter = $\hat{P}_i[k] + u_{\theta_i}[k] + \bar{\eta}_i$;
    elseif $(\hat{P}_i[k] + u_{\theta_i}[k] + \bar{\eta}_i) \geq$ threshold **then**
      reset counter (counter = 0);
      $send\_pkt(Sync)$;
    end if
  $k = k + 1$;

Through defining $\mathbf{K} = \mathbf{U}^{-1}V$, according to (30), we have

$$\begin{aligned} &\begin{bmatrix} -diag(Q, \rho^2 I) & (G\mathbf{A} + \mathbf{B}V\mathbf{C})^T \\ G\mathbf{A} + \mathbf{B}V\mathbf{C} & -G - G^T + diag(Q, I) \end{bmatrix} \\ &+ \left(\begin{bmatrix} 0 & I \end{bmatrix}^T (G\mathbf{B} - \mathbf{B}\mathbf{U}) \mathbf{U}^{-1}V \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \right) \\ &+ \left(\begin{bmatrix} 0 & I \end{bmatrix}^T (G\mathbf{B} - \mathbf{B}\mathbf{U}) \mathbf{U}^{-1}V \begin{bmatrix} \mathbf{C} & 0 \end{bmatrix} \right)^T \\ &= \begin{bmatrix} -diag(Q, \rho^2 I) & (G\mathbf{A} + G\mathbf{B}\mathbf{K}\mathbf{C})^T \\ G\mathbf{A} + G\mathbf{B}\mathbf{K}\mathbf{C} & -G - G^T + diag(Q, I) \end{bmatrix} < 0. \end{aligned} \tag{31}$$

Based on [24], [25], (31) is the bounded real lemma with the auxiliary variable matrix G. Once the matrix inequality (29) is established, the H_∞ performance ρ of any arbitrary pairwise system (22) is guaranteed. Since the spanning tree network is a directed graph, the H_∞ performance of the networked system

$\mathcal{G}$ is also guaranteed. This means that clock and delay noises in the spanning tree clock synchronisation network possess a small impact (i.e. ρ times) on the output accuracy. □

IV. Simulation Results

To validate the theoretical results in the preceding section, here, we conduct numerical simulations in a (randomly generated) 50-node spanning tree network (see Fig. 3). For the simulations, the initial clock offset $\theta_i[0]$ and initial skew $\gamma_i[0]$ are chosen randomly and uniformly in the corresponding intervals (0.4 s, 0.8 s) and (0 ppm, 50 ppm). The clock offset and skew are subject to random perturbations with the standard deviations $\sigma_{\theta_i} = 1$ μs and $\sigma_{\gamma_i} = 1$ ppm respectively. The synchronisation cycle is 1 second. The standard deviation of packet exchange delay is $\sigma_{\kappa_i} = 4$ μs [11]. This means that the standard deviations of offset and skew measurement noises [see (15)] are, respectively, 4 μs and around 6 μs.

The condition in Theorem 1 is used to design a static output feedback controller, the H_∞ performance $\rho = 14.671$ is obtained under $\zeta = 0.4895$ and $\xi = 0.4937$. The control gain K is equal to

$$K = \begin{bmatrix} 0.7615 & 0 \\ 0 & 0.1253 \end{bmatrix}.$$

In addition, two synchronisation approaches, namely, PISync and TPSN, are also selected for performance comparison. Figs. 4 and 5 respectively show the evolution of offset and skew over time. Clearly, all three solutions let both the clock offset and skew converge to corresponding constant values, and thus the steady synchronised state is achieved in the network. In the PISync and TPSN protocols, since the complete offset estimate is used for clock correction (i.e. $\alpha = 1$), their convergence speed is faster than that of R-PkCOs (see Fig. 4).

Even though the adaptive tuning method is utilised in PISync, the order-of-magnitude of β is still tiny, and is less than 5×10^{-07} (from the simulation results). Thus, such a small value of β cannot overcome the joint effects from the drifting clock frequency (with the standard deviation of 1 ppm) and $\nu_{\gamma_i}[k]$ (with the standard deviation of 6 μs). The failure of clock skew correction (see Fig. 5) also leads to worse precision of around 400 μs, as shown in Fig. 4. For the TPSN protocol, the drifting clock is adjusted by using the full clock skew estimate (i.e. $\beta = 1$), which suffers from large clock skew measurement noise $\nu_{\gamma_i}[k]$ with the standard deviation of 6 μs (while the frequency's standard deviation is only about 1 ppm). As a result, an over-correction occurs on $\gamma_i[k]$ (see Fig. 5), and the synchronisation performance degrades. In R-PkCOs, the control gain K is obtained from Theorem 1, which implies that a small (i.e. less than $\rho = 14.671$ times) effect of clock and delay disturbances is on the output synchronisation accuracy. Thus, the R-PkCOs protocol achieves better precision, compared to PISync and TPSN.

For PISync and TPSN, their corresponding under- and over-correction are also reflected in the evolution of $\aleph_i = \frac{\|z_i\|_2}{\|d_i\|_2}$ (see Fig. 6). The proposed R-PkCOs method guarantees $\aleph_i$ of each node is smaller than 6. However, during steady synchronised state, $\aleph_i$ of PISync and TPSN are only about 50 and 30 respectively.

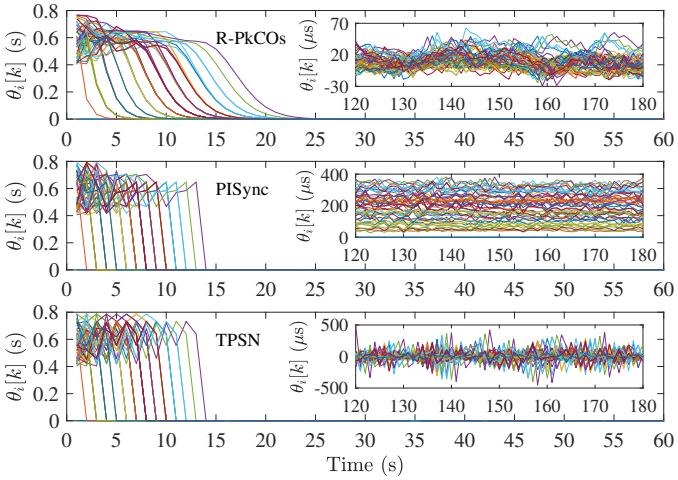


Fig. 4. Evolution of the clock offset under three synchronisation algorithms.

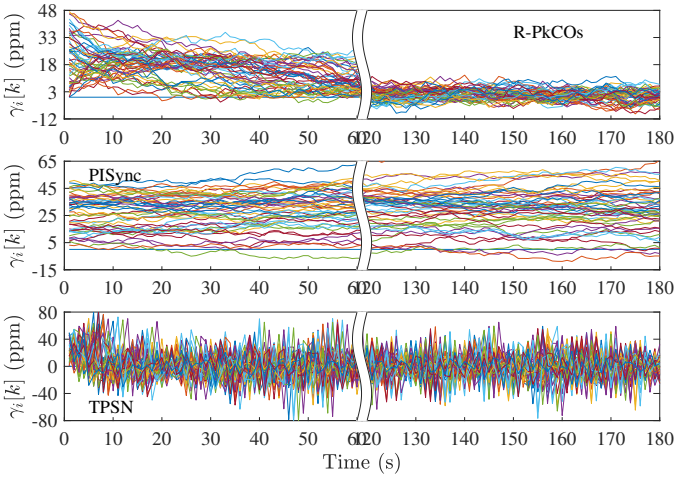


Fig. 5. Evolution of the clock skew via R-PkCOs, PISync and TPSN.

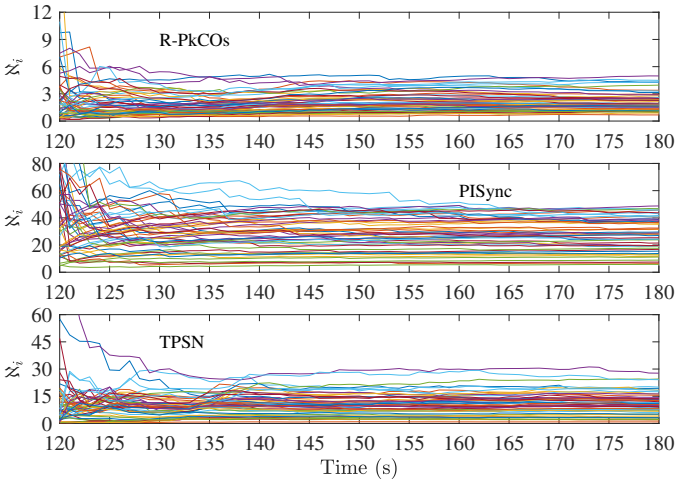


Fig. 6. Evolution of $\aleph_i = \frac{\|z_i\|_2}{\|d_i\|_2}$ under three clock synchronisation methods.

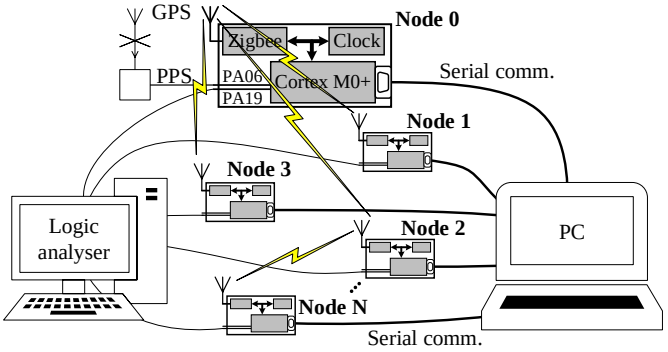


Fig. 7. Architecture of the wireless network hardware testbed.

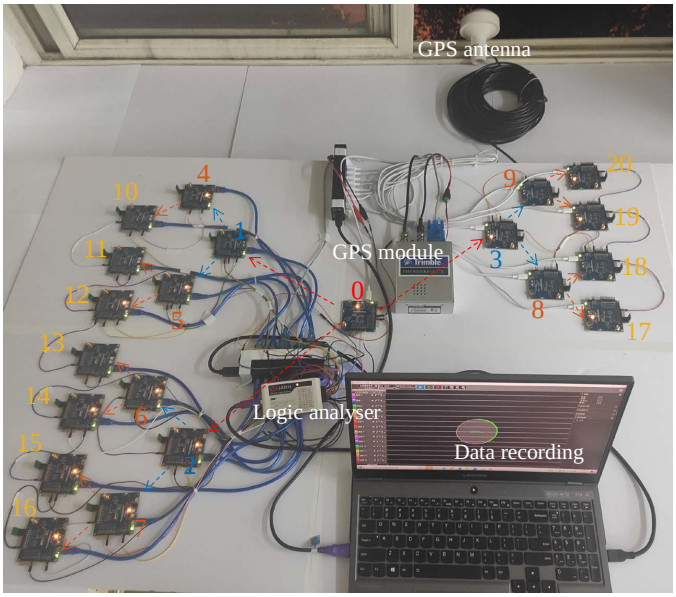


Fig. 8. Hardware testbed (dashed line: *Sync* packet exchange direction).

V. Experimental Evaluation

This section evaluates the performance of the proposed R-PkCOs synchronisation method in a spanning tree network (see Figs. 7 and 8). For the implementation, the clock's time variable is represented by a 32-bit counter register (i.e. COUNT) of the RTC module, which is driven by an external 32.768 MHz crystal oscillator. The threshold register (i.e. COMP) is set to 32767999 to let the embedded clock reset each second. Once COUNT matches COMP, the processor issues a hardware interrupt, where COUNT is reset to zero, meanwhile, a 21-byte *Sync* packet is transmitted. Upon the reception of the wireless packet, the other hardware interrupt [i.e. Address Match Interrupt (AMI)] is triggered to generate a timestamp, which is used for offset calculation and clock correction. In addition, Algorithm 1 presents the pseudocode of R-PkCOs.

During the experiments, the Trimble ThunderBolt E GPS Disciplined Clock [26] is connected to the root node, for providing the reference time [i.e. the Pulse Per Second (PPS) signal] to the network (see Figs. 7 and 8). This means that the synchronisation cycle T is one second. The average values of packet exchange and processing delays are around

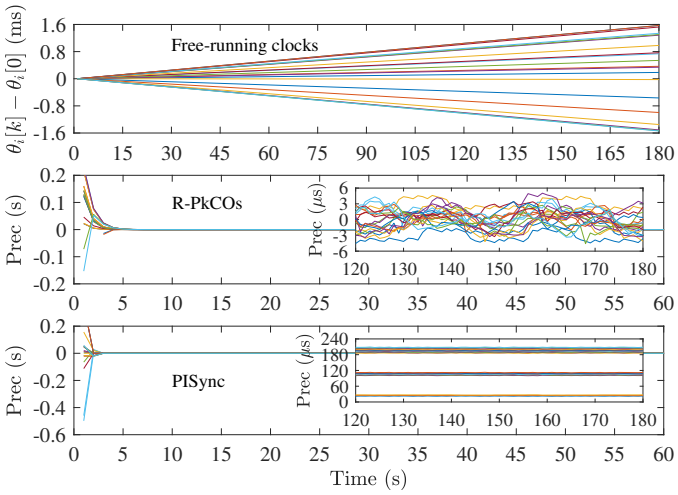


Fig. 9. Evolution of free-running embedded clocks, and the synchronisation precision by using the R-PkCOs and PISync protocols.

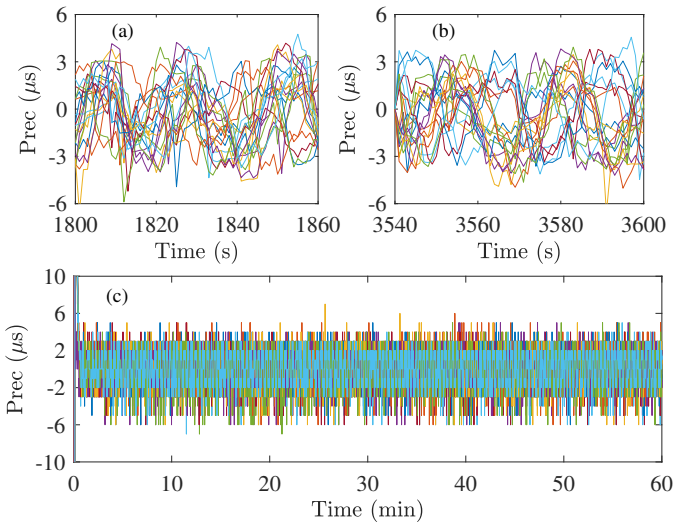


Fig. 10. One-hour performance of the R-PkCOs scheme.

514.25 μs and 117 μs respectively, and the corresponding standard deviations are of 0.3 μs and 0.3 μs. The control gains α and β are $1/1.3$ and $1/8$, respectively, which are the same as the parameters used in the simulations. t_{dp} and t_{sd} are set to 9.15 ms and 3.66 ms, respectively. The logic analyser [27] is used to evaluate the performance, and the synchronisation precision is defined as the time difference between the sensor node clock and root node's clock. Moreover, the PISync protocol is chosen for comparison.

From Fig. 9, it can be seen that each sensor nodes possess the unique frequency drifting characteristic. The clock offset increments of 20 sensor nodes in the experiments are between -1.6 ms and 1.6 ms, if no time synchronisation protocol is applied to the network. By using the proposed R-PkCOs protocol, the synchronisation precision in the spanning tree network is up to 6 μs; while, the first-hop node precision of PISync is about 20 μs, which also coincides with [21].

In addition, we also study the one-hour performance of the R-PkCOs protocol. The logic analyser can only sample 100-

second data; thus, the serial communication method [11] is utilised for data collection and analysis. Fig. 10c shows the time synchronisation precision obtained by using the serial communication method. The precision converges from the initial value to around 6 μs, and this accuracy (from Fig. 10c) is similar to the performance calculated via the logic analyser, as shown in Figs. 10a and 10b. Overall, by using the slot-based one-way *Sync* packet transmission scheme, and the control gain obtained from Theorem 1, R-PkCOs achieves time synchronisation with the precision of about 6 μs during the one-hour experiments.

VI. Conclusion

In this paper, we propose the R-PkCOs protocol to correct both clock skew and offset for improving synchronisation performance, subject to the impacts of drifting frequency, and external perturbations from packet exchange and processing delays. The proposed algorithm not only uses the slot-based one-way *Sync* packet transmission mechanism, but also adopts the H_∞ control method to guide the parameter selection, for letting clock and delay noises possess a tiny (i.e. ρ times) impact on the synchronisation accuracy. Through designing the static controller, the ratio between the modulus of the achieved synchronisation precision and the magnitude of clock and delay noises is always smaller than a given value ρ, thereby guaranteeing robustness of R-PkCOs. The one-hour experimental results show that the proposed protocol is capable of achieving clock synchronisation with the precision of 6 microseconds in a 21-node spanning tree network. Thus, the R-PkCOs synchronisation technology can be applied in the IIoT applications to provide an accurate common sense of timing (up to 6 μs) among wireless nodes.

Acknowledgment

The authors would like to thank Mr Jun Xiong and Prof. Xiao-Heng Chang for providing help in the development of the Linear Matrix Inequality (LMI) program on MATLAB.

References

[1] A. M. Romanov, F. Gringoli, and A. Sikora., "A Precise Synchronization Method for Future Wireless TSN Networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3682-3692, May. 2021.

[2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund., "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724-4734, Nov. 2018.

[3] T. Qiu, Y. Zhang, D. Qiao, X. Zhang, M. L. Wymore, and A. K. Sangaiah., "A Robust Time Synchronization Scheme for Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3570-3580, Aug. 2018.

[4] Z. Hanzalek, and P. Jurcik., "Energy Efficient Scheduling for Cluster-Tree Wireless Sensor Networks With Time-Bounded Data Flows: Application to IEEE 802.15.4/ZigBee," *IEEE Trans. Ind. Informat.*, vol. 6, no. 3, pp. 438-450, Aug. 2010.

[5] T. Qiu, X. Liu, M. Han, H. Ning, and D. Wu., "A Secure Time Synchronization Protocol against Fake Timestamps for Large-Scale Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1879-1889, Dec. 2017.

[6] H. M. Smith., "Synchronous Flashing of Fireflies," *Science*, vol. 82, no. 2120, pp. 151-152, Aug. 1935.

[7] R. E. Mirollo, and S. H. Strogatz., "Synchronization of Pulse-Coupled Biological Oscillators," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1645-1662, 1990.

[8] Y. Zong, X. Dai, P. Canyelles-Pericas, K. Busawon, R. Binns, and Z. Gao., "Modelling and Synchronisation of Delayed Packet-Coupled Oscillators in Industrial Wireless Sensor Networks," in *Proc. 21st IFAC World Congr. (IFAC)*, Jul. 2020.

[9] IEEE Standard for Low-Rate Wireless Networks, IEEE standard 802.15.4-2020, 2020.

[10] Y. Zong, X. Dai, Z. Gao, K. Busawon, R. Binns, and I. Elliott., "Synchronization of Pulse-coupled Oscillators for IEEE 802.15.4 Multi-hop Wireless Sensor Networks. in *Proc. 2018 Conf. Commun. (GlobeCom)*, Dec. 2018.

[11] Y. Zong, X. Dai, and Z. Gao., "Proportional-Integral Synchronisation for Non-identical Wireless Packet-Coupled Oscillators with Delays," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11598-11608, Nov. 2021.

[12] J. Elson, L. Girod, and D. Estrin., "Fine-Grained Network Time Synchronization Using Reference Broadcasts," in *Proc. 5th Symp. Operating Syst. Design Implementation*, 2002, pp. 147-163.

[13] S. Ganeriwal, R. Kumar, and M. B. Srivastava., "Timing-sync Protocol for Sensor Networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, Nov. 2003.

[14] F. Shi, X. Tuo, S. X. Yang, J. Lu, and H. Li., "Rapid-Flooding Time Synchronization for Large-Scale Wireless Sensor Networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1581-1590, Mar. 2020.

[15] K. S. Yildirim, R. Carli, and L. Schenato., "Adaptive Proportional–integral Clock Synchronization in Wireless Sensor Networks," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 2, pp. 610-623, Mar. 2018.

[16] G. J. Pottie, and G. J. Pottie., "Wireless Integrated Network Sensors," *Commun. ACM*, vol. 43, pp. 51-58, May. 2000.

[17] F. Lin, W. Dai, W. Li, Z. Xu, and L. Yuan., "A Framework of Priority-Aware Packet Transmission Scheduling in Cluster-Based Industrial Wireless Sensor Networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5596-5606, Aug. 2020.

[18] H. Wang, L. Shao, M. Li, B. Wang, and P. Wang., "Estimation of Clock Skew for Time Synchronization Based on Two-Way Message Exchange Mechanism in Industrial Wireless Sensor Networks," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4755-4765, Nov. 2018.

[19] X. Huan, K. Kim, S. Lee, E. Lim, and A. Marshall., "A Beaconless Asymmetric Energy-Efficient Time Synchronization Scheme for Resource-Constrained Multi-Hop Wireless Sensor Networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1716-1730, Mar. 2020.

[20] Z. An, H. Zhu, X. Li, C. Xu, Y. Xu, and X. Li., "Nonidentical Linear Pulse-coupled Oscillators Model with Application to Time Synchronization in Wireless Sensor Networks," *IEEE Trans. Ind. Electron.*, vol. 58, no. 6, pp. 2205-2215, Jun. 2011.

[21] Y. Tian, S. Chun, G. Chen, S. Zong, Y. Huang, and B. Wang., "Delay Compensation-Based Time Synchronization Under Random Delays: Algorithm and Experiment," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1, pp. 80-95, Jan. 2021.

[22] Y. Zong, X. Dai, Z. Gao, K Busawon, and J. Zhu., "Modelling and Synchronization of Pulse-Coupled Non-identical Oscillators for Wireless Sensor Networks," in *Proc. 16th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2018.

[23] G. Giorgi, and C. Narduzzi., "Performance Analysis of Kalman-Filter-based Clock Synchronisation in IEEE 1588 networks," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 8, pp. 2902-2909, Aug. 2011.

[24] X. Chang, R. Liu, and J. H. Park., "A Further Study on Output Feedback H_∞ Control for Discrete-Time Systems," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 67, no. 2, pp. 305-309, Feb. 2020.

[25] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan., Linear Matrix Inequalities in System and Control Theory. Philadelphia, PA, USA: SIAM, 1994.

[26] The Trimble Thunderbolt E GPS Disciplined Clock.

[27] Kingst Logic Analyser LA2016.

Yan Zong received the B.Eng. degree in electrical and electronic engineering from Nanjing Normal University, China, in 2016, and the Ph.D. degree in electrical engineering from Northumbria University, U.K., in 2020. His research interests cut across several disciplines, which include machine learning, networked control systems and synchronisation, and their applications to the industrial Internet of Things.

He is currently a research fellow at Cranfield University, U.K. Prior to that he was a FPGA engineer at Keiky Ltd., U.K.

Xuewu Dai received the B.Eng. degree in electronic engineering and the M.Sc. degree in computer science from Southwest University, China, in 1999 and 2003, respectively, and the Ph.D. degree from the University of Manchester, U.K., in 2008. His research interests include robust state estimation, networked control systems and synchronisation, and their applications to the time-sensitive Industrial Internet of Things.

He is a Senior Lecturer with the Department of Mathematics, Physics and Electrical Engineering, Northumbria University. Prior to that he was a postdoc at the Department of Engineering Science, University of Oxford (2011-2013) and at the Department of Electronic and Electrical Engineering, UCL (2009-2011).

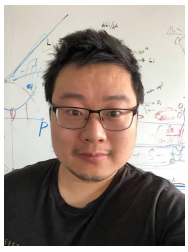
Dr Dai was awarded the Early Career Research Prize by SWIG UK.

Zhuangkun Wei received the bachelor's and master's degrees in electronic engineering from Beijing University of Posts and Telecommunications (BUPT), China, in 2014 and 2018, respectively, and the Ph.D. degree in engineering from the University of Warwick, U.K., in 2021. His research interests cover physical layer security, graph signal processing, molecular communications, and Explainable Artificial Intelligence (XAI).

He is currently a research fellow in the School of Aerospace, Transport and Manufacturing, Cranfield University.

Mengbang Zou received his M.Sc. degree in mechanical science and engineering from Huazhong University of Science and Technology, China, in 2019. He is interested in complex system with nonlinear dynamics including complex system's resilience, synchronization and region of attraction.

He is now a Ph.D. student in the School of Aerospace, Transport and Manufacturing of Cranfield University.



Weisi Guo received his Ph.D. degree from the University of Cambridge, U.K., in 2011. He was an Ass. Professor at the University of Warwick (2012-19) and Turing Fellow (2017-19). His research interests focus on networks, data science, and autonomy.

He is currently a full Professor and head of the Human Machine Intelligence Group at Cranfield University, U.K.

Prof. Guo was a winner of the IET Innovation Award.

Zhiwei Gao received the B.Eng. degree in electrical engineering and automation and the M.Eng. and Ph.D. degrees in systems engineering from Tianjin University, Tianjin, China, in 1987, 1993, and 1996, respectively. His research interests include stochastic control systems, data-driven modelling, estimation and filtering, fault diagnosis, resilient control, intelligent optimisation, power electronics, wind energy systems, electric vehicle batteries, and bioinformatics.

He is currently with the Faculty of Engineering and Environment, Northumbria University, Newcastle upon Tyne, U.K., as a Reader.

Dr Gao serves several leading international journals as academic editors, and organised more than ten special issues in the premier international journals.