

An enhanced distributed control-theoretic time synchronization protocol using sliding mode control for wireless sensor and actuator network

Yeong Chin Koo, Muhammad Nasiruddin Mahyuddin

School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Malaysia

Article Info

Article history:

Received Sep 20, 2018

Revised Dec 22, 2018

Accepted Jan 10, 2019

Keywords:

Distributed

Sliding mode control

Time synchronization

Wireless sensor and actuator network

ABSTRACT

Time synchronization is very important in a wireless sensor and actuator network (WSAN) as it provides a common time notation to the WSAN. To handle the time synchronization issue, this paper presented an enhanced distributed control-theoretic time synchronization protocol for wireless sensor and actuator networks, named Time Synchronization using Distributed Observer algorithm with Sliding mode control element (TSDOS). In this protocol, an augmented sliding mode control element is used for robustification purpose. From the theoretical convergence analysis and simulation results presented in this paper, it proved that the proposed TSDOS is the best if compared to another two algorithms from the literature in terms of cumulative integral absolute error.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Muhammad Nasiruddin Mahyuddin,
School of Electrical and Electronic Engineering,
Universiti Sains Malaysia,
Engineering Campus, 14300 Nibong Tebal, Pulau Pinang, Malaysia.
Email: nasiruddin@usm.my

1. INTRODUCTION

Time synchronization is a crucial issue to be addressed in the implementation of WSANs. Through time synchronization, all the sensors and actuators in the network will follow the same time reference. With the help of same time reference, WSANs can sequence the events and model the physical environment sensed by the sensor nodes accurately [1]. Besides, a same time reference among the sensors is also influential in identify the casual relationships between events in the physical world, to eliminate redundant sensor data, and to generally facilitate sensor and actuator network operation [2].

There are many synchronization protocols available in the literature, for example, Reference Broadcast Synchronization (RBS) [3], Time-sync Protocol for Sensor Network (TPSN) [4], Reachback Firefly Algorithm (RFA) [5], Average TimeSync (ATS) [6], Time Synchronization using Maximum and Average consensus protocol (TSMA) [7] and Proportional-Integral clock Synchronization (PISync) [8].

As pointed out in [8], most of the protocols in the literature, especially those based on least-squares regression, have a drawback where the effect of various error sources will appear as multiplicative noise in the time synchronization error dynamic, which in turn cause the global synchronization error to increase exponentially with the diameter of the network, approximately. This drawback will then cause the poor performance in term of scaling properties. This flaw can be solved by using control theory based approach, where the synchronization is accomplished using linear feedback on the measured local synchronization error. Through this control-theoretic approach, the error sources will appear as additive noise in the error dynamic and the global synchronization error will proximately increase as the square root of the diameter of the network [8].

The time synchronization protocol presented in this paper is a control-theoretic approach. It is called Time Synchronization using Distributed Observer algorithm with Sliding mode control element (TSDOS), which applies a distributed observer/estimator algorithm and sliding mode control element on relative comparative error to estimate the skew rate and relative skew rate to accomplish the purpose of time synchronization.

TSDOS has the following characteristics: distributed, asynchronous, robust and light in term of computation effort. The rest of this paper is organized as follows. Section 2 describes the development of TSDOS algorithm and the convergence analysis of the TSDOS algorithm, followed by the brief details of the simulation setup in the last sub-section of Section 2 while the simulation results are displayed and discussed in Section 3. The simulation results of TSDOS are compared with another two time synchronization protocols from literature: ATS [6], and PISync [8] Finally, Section 4 concludes the findings.

2. TSDOS METHODOLOGY

2.1. Clock Modelling

Every node i in a WSN has its own local clock which is a monotonically non-decreasing function of t . Each local clock dynamic can be modeled as first order dynamics, as shown in (1)

$$\tau_i(t) = \alpha_i t + \beta_i \tag{1}$$

where τ_i is the local clock reading, α_i is the local clock skew which determines the clock speed, and β_i is the local clock offset which is the difference between time reported by clock τ_i and the real time at the initial instant t_0 .

Figure 1 displays relative clock dynamics between a pair of nodes with different skew rate and offset [6]. Different nodes have different skew rate and offset due to the low-cost oscillators in the nodes.

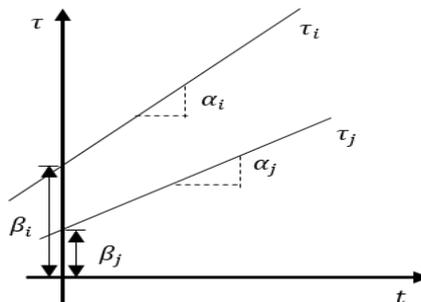


Figure 1. The clock dynamics of two nodes, i and j , with their respective local clock skew rate, α_i and clock offset, β_i

To achieve absolute synchronization in a sensor and actuator network, all nodes $i = 1, 2, \dots, \mathcal{N}$ must precisely compensate for their clock parameters α_i and β_i so that all clocks have same clock skew and zero offset error [9]. Without information on the real world absolute time t available to the nodes in WSN, computing for the clock parameter α_i and β_i is impossible. As reported in [10], to avoid this issue, firstly, we can rearrange (1) so that t is expressed as a shared entity, i.e.

$$t = \frac{\tau_i - \beta_i}{\alpha_i} \tag{2}$$

and then substitute (2) into (1) for node j so that we will reach

$$\tau_j = \frac{\alpha_j}{\alpha_i} \tau_i - \frac{\alpha_j}{\alpha_i} \beta_i + \beta_j \tag{3}$$

As can be observed, as shows in (3), local clock dynamic for node j does not contain the absolute reference time t . The paradigm adopted in [6] is that, we may eventually obtain the clock skew rate and offset of all the nodes in a network by estimating the relative magnitude $\frac{\alpha_j}{\alpha_i}$ and the offset imposed

$(\beta_j - \frac{\alpha_j}{\alpha_i} \beta_i)$. If these magnitudes can be estimated eventually, the clock dynamics estimates $\hat{\tau}_i$ can be furnished indirectly. Therefore, we may express the clock dynamics estimates of each node in terms of these estimated magnitudes,

$$\hat{\tau}_i = \hat{\alpha}_i \tau_i + \hat{\delta}_i \quad (4)$$

where $\hat{\alpha}_i$ is the relative clock skew rate estimate and $\hat{\delta}_i$ is the clock offset estimate. As shows in (4) that each node has its own self-time keeping mechanism by estimating its own virtual time, $\hat{\tau}_i$ before the estimates is passed to its connected neighbor node j . The final objective is to have a virtual reference clock, τ_r ,

$$\tau_r(t) = \alpha_r t + o_r \quad (5)$$

to which all the nodes will refer to and finally be synchronized, i.e.

$$\lim_{t \rightarrow \infty} \hat{\tau}_i = \tau_r, \forall i \quad (6)$$

The end objective and motivation is like the TSMA algorithm [7] whereby, as shows in (6) describes the consensus in achieving the common reference time except that TSMA adopts the maximum averaging of all the virtual time estimates. On the other hand, ATS algorithm in [6] adopts a kind of low pass filter structure from which, in this paper, we have enhanced the algorithm by imposing a sliding-mode like term for fast convergence in relative skew estimates and skew rate estimates.

2.2. TSDOS Algorithm

TSDOS algorithm is an extended algorithm from the work of ATS [6] through an enhancement by the sliding-mode control term. Two assumptions are deduced for the algorithm to work. First, the communication created between neighboring nodes is a single-hop communication which occurs at a very short finite time interval. Second, the communication occurs between a node pair is instantaneous.

TSDOS consists of three main parts: the relative skew rate estimation, the skew rate estimation, and the offset estimation. In this work, each node i is assumed to periodically transmit a packet to all its neighbors with a synchronization period equal to T , i.e., the transmission instant t_ℓ^i

$$t_\ell^i = \frac{\ell T - \beta_i}{\alpha_i} \quad (7)$$

2.2.1 Relative Skew Estimation

Every node i tries to estimate the relative skew rate with respect to their neighbor j . The relative skew rate estimates in [6] is enhanced by incorporate a sliding-mode like term at the relative skew rate. We first define a sliding plane ϵ_i ,

$$\epsilon_i = \frac{\tau_j(t_2) - \tau_j(t_1)}{\tau_i(t_2) - \tau_i(t_1)} \quad (8)$$

from which i.e. $\tau_{i,j}(t_2) - \tau_{i,j}(t_1)$ taken at time t_1 and time t_2 from the data packet for node j is compared with node i . Then the relative comparative error is fed into a low-pass filter structure to estimate the relative skew rate,

$$\hat{\xi}_i(t+1) = \rho_\xi \hat{\xi}_i(t) + (1 - \rho_\xi) \text{sign}(\epsilon_i) \quad (9)$$

where $\rho_\xi \in \{0,1\}$ is the tuning parameter similar to the effects of the filter poles coefficient, i.e. increasing the coefficient may increase the convergence of the filtered signal but may reduce the effects of filtering. A signum function forces the comparative clock pairs to the plane defined in (8).

Theorem 1. Consider the relative skew update (9) where $0 < \rho_\xi < 1$, the transmission events t_ℓ^i are generated according to the (7), and each τ_i evolves according to (1). Then

$$\lim_{t \rightarrow \infty^+} \hat{\xi}_i(t) = 1$$

in finite-time for any initial condition $\hat{\xi}_i(0)$.

Proof. See Appendix A.

2.2.2 Skew Estimation

As stated previously in (4), a node updates its own perceived virtual time. The skew rate estimate is also enhanced by the sliding mode term in the filter,

$$\hat{\alpha}_i(t + 1) = \rho_s \hat{\alpha}_i(t) + (1 - \rho_s) \hat{\xi}_i \text{sign}(\hat{\alpha}_j) \tag{10}$$

where $\rho_s \in \{0,1\}$ is the tuning parameter and $\hat{\alpha}_j$ is the virtual clock skew estimate of the connected neighbor node j .

Theorem 2. Consider the skew update equation given by (10) with initial condition $\hat{\alpha}_i(0) > 0$ and $0 < \rho_s < 1$, where $\hat{\xi}_i(t)$ is updated according to (9) and t_p^i are defined in (7). Then

$$\alpha_r = \lim_{t \rightarrow \infty} \hat{\alpha}_i(t) = 1, \forall i, \alpha_i(0) > 0$$

in finite-time.

Proof. See Appendix B.

2.2.3 Offset Estimation

Last but not least, after all the nodes are running at the same speed, the offset $\hat{\delta}_i$ of the virtual clock can be estimated by,

$$\hat{\delta}_i(t + 1) = \hat{\delta}_i(t) + (1 - \rho_o)(\hat{t}_j - \hat{t}_i) \tag{11}$$

where $\rho_o \in \{0,1\}$ is the tuning parameter and \hat{t}_i, \hat{t}_j are the estimated virtual clock for node i and their neighbor node j , respectively.

Proposition 1 (Theorem 6, [6]). Consider the offset update equation given by (9) with initial condition $\hat{\delta}_i(0) = 0$ and $0 < \rho_o < 1$, where \hat{t}_i , and t_p^i , are defined in (4) and (7), respectively. Then

$$\lim_{t \rightarrow \infty} \hat{t}_i(t) = \hat{t}_j(t), \forall i, j \in \mathcal{N}_j$$

exponentially fast.

Finally, the virtual clock estimator, \hat{t} , for each node i is then obtained by combining the skew estimator and offset estimator value from (10) and (11).

2.3. Simulation Setup

An experimental simulation is designed in this paper to verify the proposed time synchronization protocol, besides comparing the performance with another two control-theoretic protocols from literature, ATS [6] and PISync [8]. The experimental simulations are simulated using MATLABM SimulinkTM where ad-hoc deployment of 5 nodes in a WSN is simulated. Figure 2 shows the connectivity among the 5 nodes in the simulation. The final objective of the simulation is all the 5 nodes will eventually be synchronized to a same time reference.

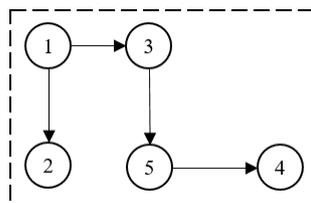


Figure 2. Connectivity of the 5 nodes

The synchronization period in this simulation is set to be 1 milliseconds and is running for a total of 100 seconds. In the simulation, cumulative Integral Absolute Error or cIAE which can be expressed as

$$cIAE = \sum_{i=1}^N \int_0^T |e_i| dt \quad (12)$$

Was computed for the whole nodes in the network as a measure of the computational effort (control effort) of the protocol. N denotes the total number of nodes in a network whose perceived virtual time are to be synchronized. Such computed index also gives an indication of the energy consumption of each node to achieve time synchronization, thus the lesser value is desirable. This is one of the important traits for a wireless sensor node as they need to conserve energy.

3. RESULTS AND DISCUSSION

Figure 3 illustrates the plots of virtual time estimator, $\hat{\tau}$, while Figure 4 displays the plots of virtual time estimation error, $\tilde{\tau}$, for the three algorithms, TSDOS, ATS, and PISync. In general, it can be observed that all the three algorithms are able to synchronize the nodes. As observed in Figure 3, the virtual time estimate of all three algorithms are increasing linearly, which fulfill the first order dynamic stated in (5). In addition, from the plots of virtual time estimation error in Figure 4, it can be deduced that the error of all the 3 algorithms are remain constant and relatively small throughout the synchronization period.

However, more precisely, from Figure 3 (c), it is noticeable that the line width of the virtual time estimate plot of PISync is quite thick compared to another two figures, Figure 3 (a) and Figure 3 (b). Next, from the error plot in Figure 4, it explained the reason of the thickness, where the error of TSDOS and ATS is converged to 0 (Figure 4 (a), 4 (b)), while the error of PISync is converged to a bigger range of value, between -6 to 12 (Figure 4 (c)), which shows the performance of PISync is unfavorable compared to TSDOS and ATS.

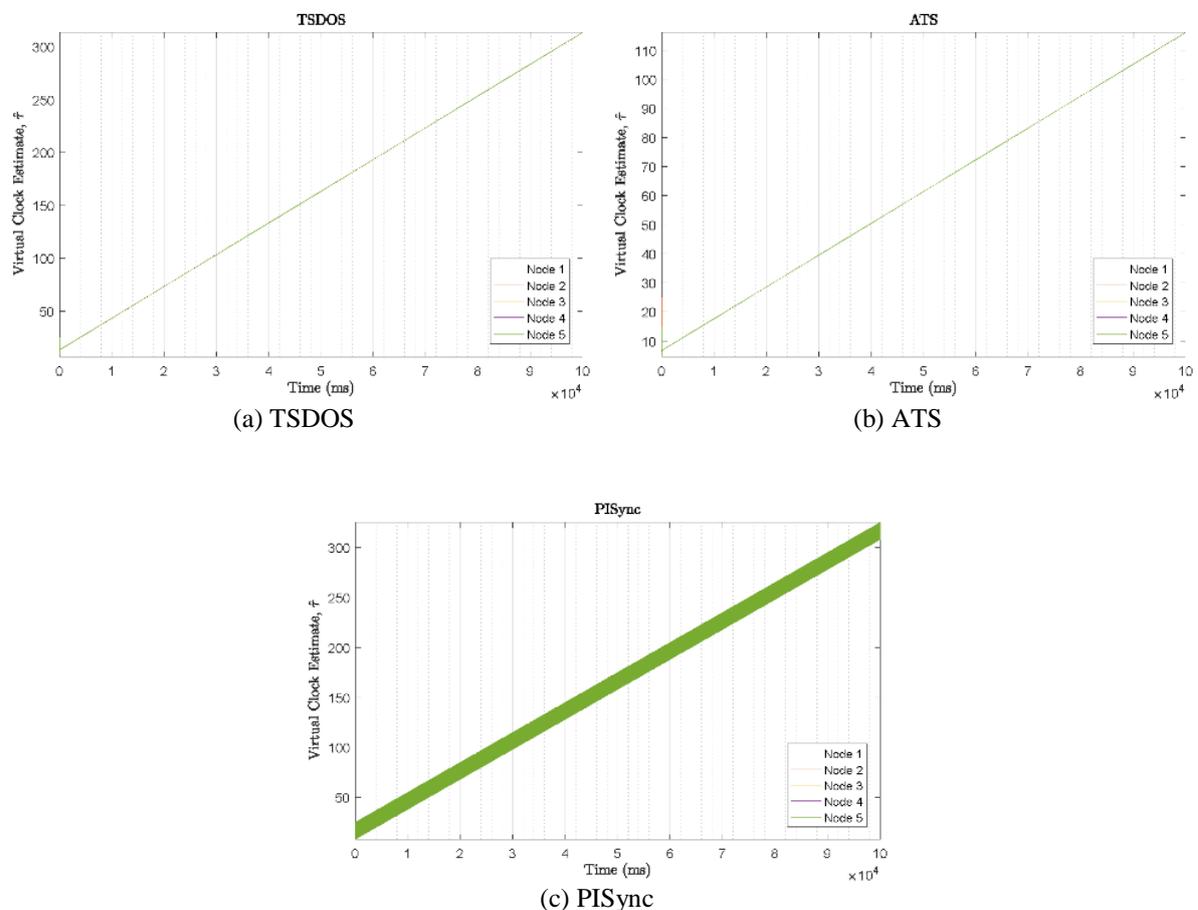
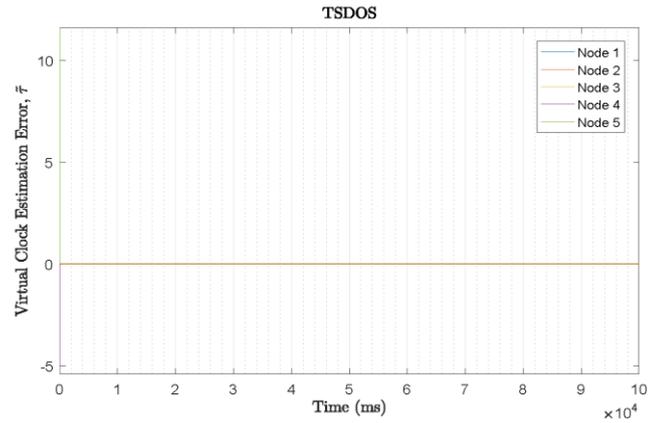
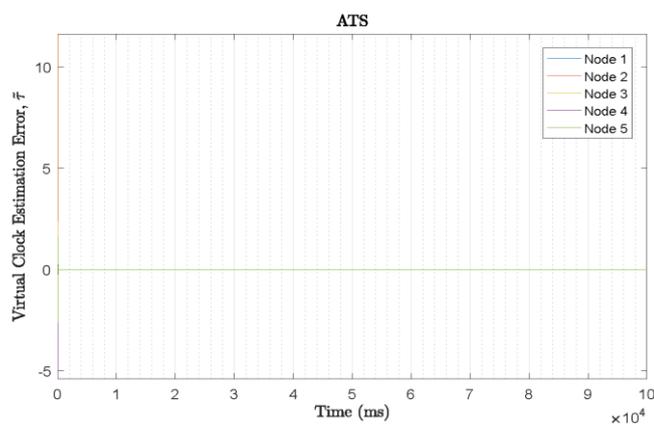


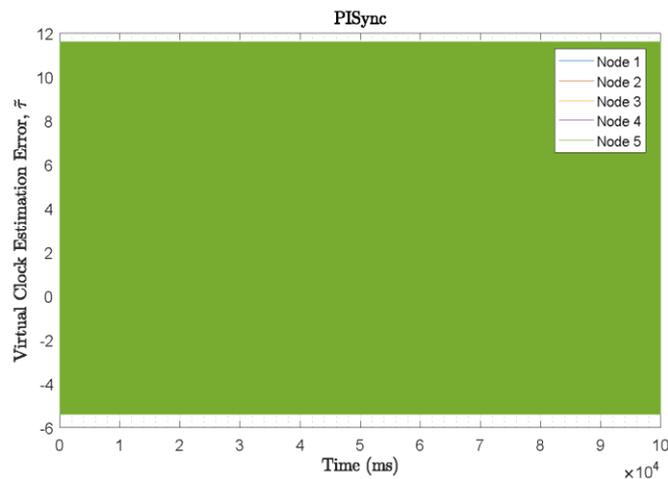
Figure 3. Plots of virtual time estimate of (a) TSDOS, (b) ATS, and (c) PISync



(a). TSDOS



(b). ATS



(c). PISync

Figure 4. Plots of virtual time estimation error of (a) TSDOS, (b) ATS, and (c) PISync

The cIAE of the three algorithms are summarized in Figure 5 to have a clearer comparison on the three algorithms. From Figure 5, it can be clearly seen that TSDOS and ATS once again outperformed PISync, where TSDOS and ATS have a much lower cIAE, 1.1089×10^3 and 1.6653×10^3 , respectively, compared to the cIAE of PISync, 6.1558×10^7 . In addition, the bar charts in Figure 5 also clearly shows that TSDOS has the best performance which can be easily observed from Figure 5 that the bar chart for TSDOS is the lowest among the 3 bar charts. This is because TSDOS has the lowest value in term of cIAE, compared to

the other 2 protocols. As mentioned previously, cIAE also give an indication on the energy consumption of the protocol. Thus, we can conclude that TSDOS has reduced 33.43% and 99.99% of the energy consumption if compared to ATS and PISync, respectively.

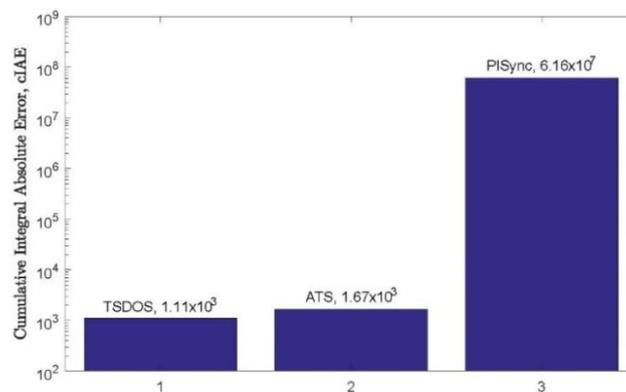


Figure 5. cIAE of TSDOS, ATS, and PISync

4. CONCLUSION

In conclusion, this paper has successfully proposed an enhanced version of distributed time synchronization protocol. Firstly, this protocol is claimed to be completely asynchronous because it does not require synchronous packets transmission between nodes. Secondly, it is a distributed protocol and has low computational energy requirement because each node is run the same light observer/estimator algorithm to estimate the virtual reference time. The proposed TSDOS algorithm has the best performance among the three compared protocols, where TSDOS has the lowest cIAE. Future work will involve the hardware implementation of the proposed time synchronization algorithm on a group of interconnected sensor node network.

ACKNOWLEDGEMENTS

This work is supported by Research University (RUi) Grants (Grant No. 1001/PELECT/8014029) and Bridging Research Grant (Grant No. 304/PELECT/6316106), both awarded by Universiti Sains Malaysia.

REFERENCES

- [1] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, 1st ed. Chichester: John Wiley & Sons, Inc., 2010.
- [2] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*. West Sussex: John Wiley & Sons, Inc., 2010.
- [3] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI '02)*, 2002, vol. 36, no. SI, pp. 147–163.
- [4] S. Ganerwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2003, pp. 138–149.
- [5] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proceedings of the 3rd international conference on Embedded networked sensor systems - SenSys '05*, 2005, pp. 142–153.
- [6] L. Schenato and F. Fiorentin, "Average TimeSync: A consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, Sep. 2011.
- [7] Z. Dengchang, A. Zhulin, and X. Yongjun, "Time Synchronization in Wireless Sensor Networks Using Max and Average Consensus Protocol," *Int. J. Distrib. Sens. Networks*, vol. 9, no. 3, p. 10, 2013.
- [8] K. S. Yıldırım, 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, 2018.
- [9] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Netw.*, vol. 18, no. 4, pp. 45–50, 2004.
- [10] W. Su and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 384–397, 2005.

BIOGRAPHIES OF AUTHORS

	<p>Koo Yeong Chin received his B. Eng Honours degree in Mechatronic Engineering and MSC degree in Aerospace Engineering from Universiti Sains Malaysia in 2011 and 2015, respectively. He is currently pursuing PhD degree in Universiti Sains Malaysia in the area of control system.</p>
	<p>Dr. Muhammad Nasiruddin Mahyuddin is currently a Senior Lecturer at the School of Electrical and Electronics Engineering, Universiti Sains Malaysia and also an Honorary Visiting Fellow in Queens School of Engineering, University of Bristol. He received his PhD in 2014 from University of Bristol in the area of Mechanical Engineering, specialising on Control and Robotics. He received his M.Eng with high distinction in Mechatronic and Automatic Control from the Universiti Teknologi Malaysia in 2006 and obtained his B.Eng with Honours in Mechatronic Engineering from the International Islamic University of Malaysia in 2004. Thereafter, he started his work as an Application Engineer at Agilent Technologies working with Motion Control product. He was appointed as a Senior Associate Teacher by University of Bristol via contract, giving lecture in Nonlinear Control with Application to Robotics from October 2011 to July 2012. He was invited as a Visiting Academician at MIS Lab, Universite de Picardie Jules Verne, France in March and April 2018. His current research interests include distributed adaptive control, cooperative control, nonlinear control and parameter estimation involving mechatronics system and robotics.</p>

APPENDIX A

Proof of Theorem 1

By writing the (9) in recursive formula, it follows that

$$\begin{aligned} \hat{\xi}_i(t) &= \rho_\xi^\ell \hat{\xi}_i(0) + \sum_{h=0}^{\ell-1} \rho_\xi^h (1 - \rho_\xi) \text{sign}(\epsilon_i) \\ &= \rho_\xi^\ell \hat{\xi}_i(0) + (1 - \rho_\xi^\ell) \text{sign}(\epsilon_i) \end{aligned}$$

where $\ell = \lfloor \frac{(t - \bar{\beta}_j)}{\tau_j} \rfloor$, and $\sum_{h=0}^{\ell-1} \rho_\xi^h = \frac{1 - \rho_\xi^\ell}{1 - \rho_\xi}$ (from finite geometric series). Since $0 < \rho_\xi < 1$, then

$$\begin{aligned} \lim_{t \rightarrow \infty^+} \hat{\xi}_i(t) &= \lim_{t \rightarrow \infty^+} \rho_\xi^\ell \hat{\xi}_i(0) + \text{sign}(\epsilon_i)(1 - \rho_\xi^\ell) \\ &= \text{sign}(\epsilon_i) \end{aligned}$$

Given the properties of signum function are

$$\text{sign}(A) = \begin{cases} 1, & A > 0 \\ 0, & A = 0 \\ -1, & A < 0 \end{cases}$$

Since $\epsilon_i = \frac{\tau_j(\tau_2) - \tau_j(\tau_1)}{\tau_i(\tau_2) - \tau_i(\tau_1)}$ and τ is a monotone increasing function, so $\epsilon_i > 0$. Therefore,

$$\lim_{t \rightarrow \infty^+} \hat{\xi}_i(t) = 1$$

and the convergence is in finite-time.

APPENDIX B

Proof of Theorem 2

Following the similar notion as in Theorem 2 and Appendix A.

$$\begin{aligned}\hat{\alpha}_i(t) &= \rho_s^\ell \hat{\alpha}_i(0) + \sum_{h=0}^{\ell-1} \rho_s^h (1 - \rho_s) \hat{\xi}_i(t) \text{sign}(\hat{\alpha}_j) \\ &= \rho_s^\ell \hat{\alpha}_i(0) + \text{sign}(\hat{\alpha}_j) \hat{\xi}_i(t) (1 - \rho_s^\ell)\end{aligned}$$

From Theorem 2, $\lim_{t \rightarrow \infty^+} \hat{\xi}_i(t) = 1$.

$$\begin{aligned}\alpha_r &= \lim_{t \rightarrow \infty^+} \hat{\alpha}_i(t) \\ &= \lim_{\ell \rightarrow \infty^+} \rho_s^\ell \hat{\alpha}_i(0) + \text{sign}(\hat{\alpha}_j) (1 - \rho_s^\ell) \\ &= \text{sign}(\alpha_j) \\ &= 1\end{aligned}$$

and the convergence is in finite-time.