DOI: 10.23977/jeis.2023.080607
ISSN 2371-9524 Vol. 8 Num. 6

TSN Time Synchronization Based on Kalman Filtering

Li Kai[1,a], Zhu Yuan[1,b], Sun Zhipeng[2,c,*]

[1]School of Automotive Studies, Tongji University, Shanghai, China
[2]Nanchang Intelligent New Energy Vehicle Research Institute, Nanchang, China
[a]lk123qwert@163.com, [b]yuan.zhu@tongji.edu.cn, [c]sunzhipeng@naiine.com
[*]Corresponding author

Keywords: IEEE 802.1AS protocol, Kalman filtering, AUTOSAR

Abstract: With the continuous development of the automotive industry and network technology, automotive Ethernet is facing the need for real-time and high-precision synchronization of data transmission. In this paper, the IEEE 802.1AS protocol gPTP based on the precise time protocol is studied and analysed in depth, focusing on the design and functional implementation of the time synchronization subsystem in the time-sensitive network system, and a method to improve the accuracy of time synchronization between nodes in the AUTOSAR protocol stack based on Kalman filter is given, and is implemented on the hardware platform AURIX TC397 and NXP SJA1110. Finally, the Kalman filter algorithm is used to correct the time synchronization error. Experimental results show that the proposed algorithm can significantly reduce the time deviation between the master-slave clocks.

1. Introduction

In recent years, with the continuous development of the automotive industry and industrial control, along with the gradual integration of computer network technology and automatic control technology, the on-board network, as a product of the convergence of these emerging technological fields, has also continued to expand. Simultaneously, automobiles are advancing towards greater intelligence. Due to the continuous innovation of in-vehicle navigation, security systems, multimedia, autonomous driving and other applications, higher requirements are put forward for network reliability, bandwidth, and synchronization performance[1-3]. Time-Sensitive Networking (TSN) was proposed and developed to meet the requirements of real-time data transmission applications (e.g., multi-axis synchronization, transportation industry, etc.) for data transmission, latency, and packet loss related indicators [4-7], and to improve the interoperability, configuration, and monitoring capabilities of real-time Ethernet. In TSN, the IEEE802.1AS protocol is used to provide timing and synchronization capabilities for nodes in the network, which is referred to as the precision clock timing and synchronization protocol[8-10].

2. An overview of the principles of the IEEE 802.1AS protocol

The IEEE 802.1AS protocol requires that every node within its defined synchronous gPTP domain be a time-aware system. There are two types of time-aware systems, namely time-aware terminal stations and time-aware bridges. Among them, only one port (master port or slave port) of the terminal node is connected to the gPTP domain. A bridge node is typically a switch that has more than two ports connected to the gPTP domain. In addition, since the IEEE 802.1AS protocol itself is defined at Layer 2 of the OSI model, the timestamps of all packet entry and exit ports are recorded by hardware [11]. The synchronization process of the gPTP domain is that the master clock sends time synchronization information to all time synchronization systems directly connected to it, and the time-aware system that receives the synchronization information should add the path transmission delay to correct the time deviation to complete its own synchronization. These time-aware systems then forward the corrected synchronization information to other time-aware systems connected to their primary ports, thus achieving synchronization of the entire gPTP domain. Obviously, the measurement process in synchronization is divided into path delay measurement and time synchronization measurement. The following sections describe the messages and measurement principles involved in these two measurement processes, and finally describe and analyze the role of the other messages[12].

2.1. Path delay measurement

The path delay measurement mechanism in the IEEE 802.1AS protocol is a peer-to-peer delay measurement proposed in IEEE 1588-2008, which is completed by using Pdelay_Req, Pdelay_Resp and Pdelay_Resp_Follow_Up packets, as shown in Figure 1.

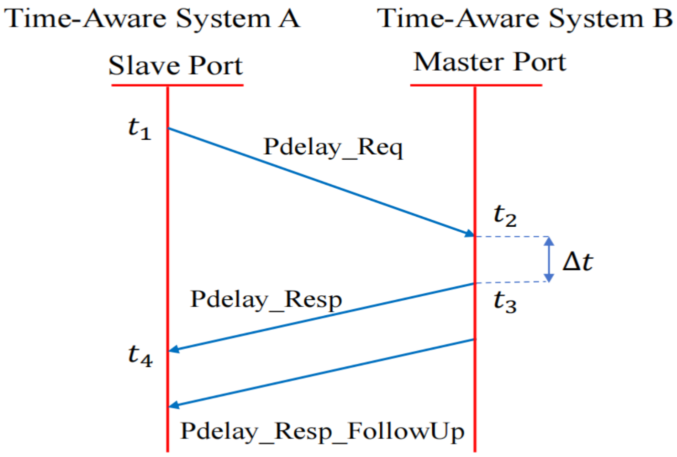

Figure 1: Schematic diagram of the peer-to-peer delay measurement process.

The measurement steps are as follows:

(1)The slave port of time-aware system A sends a Pdelay_Req request response message to the master port of time-aware system B, and records the timestamp t_1 at the time of sending. When the message arrives on port B, the master port of the time-aware system records time t_2.

(2)The master port of time-aware system B then sends a Pdelay_Resp response message, records the time t_3 when the message leaves the port, and calculates Δt. In this case, the slave port of time-aware system A records the time when the message arrives at port t_4.

(3) Finally, port B sends a Pdelay_Resp_Follow_Up message to port A, which contains a time difference Δt.

After the above three steps, the times t_1, t_4 and Δt are known for port A, where t_1, t_4 are measured by the clock of port A and Δt is measured by the clock of port B. It is very likely that the clock frequencies of ports A and B are skewed, and at the micro level, they do not see the same speed or speed of the observed time. Therefore, the IEEE 802.1AS protocol introduces frequency ratios to eliminate this error. If r is used to denote the frequency ratio of port B to port A, the equation for equivalence delay is shown in Formula (1).

$$Pdelay = \frac{t_4 - t_1 - r * \Delta t}{2} \qquad (1)$$

From this formula, it can be seen that the peer-to-peer delay mechanism considers the link delay between port A and port B to be equal, so the link delay is measured by averaging.

2.2. Time synchronization process

The IEEE 802.1AS protocol uses a point-to-point method to transmit time synchronization information, and the path delay measurement and time synchronization process are separated, making the time synchronization process more stable and accurate. The protocol clearly stipulates that only the two-step mode is used, that is, the timestamp of the SYNC message is carried Follow_Up the message, so it needs to use both Sync and FollowUp messages to complete, and the synchronization process under a full-duplex Ethernet link is shown in Figure 2.

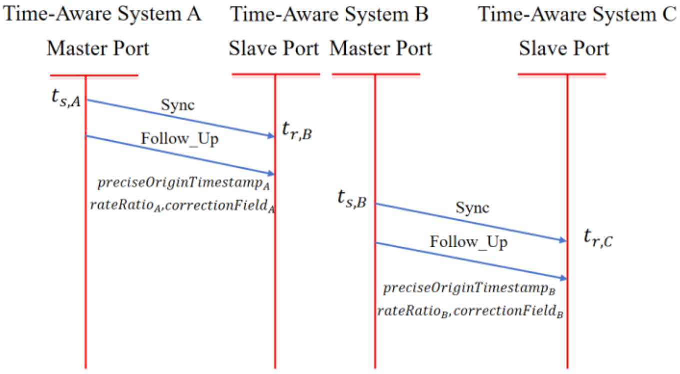

Figure 2: Time synchronization process.

After the calculation and setting of the above important fields, based on B completing the synchronization with A, the time awareness system C also maintains the synchronization with the time awareness system A. The IEEE 802.1AS protocol relies on the above-mentioned successive synchronization method to keep the clock nodes in sync with the master clock throughout the network.

3. TSN clock synchronization algorithm based on Kalman filtering

Due to the aging problem of crystal oscillators and the influence of ambient temperature, the clock drift is not constant, and the main causes of clock errors are clock stamping and delay jitter, i.e., time offset and time offset rate. In order to improve the timestamp accuracy of the TSN time synchronization node and reduce the delay jitter, the clock of the node is adjusted to make the clocks of each node as consistent as possible, and the time synchronization scheme based on Kalman filter is adopted, and the local clock is adjusted by Kalman filtering on the clock offset and clock offset rate, so that the clock synchronization is more accurate.

3.1. Establishment of a clock state model

Assuming that the sampling time of the time deviation between nodes is T, and the true value of the clock deviation at kT is represented by $T'_{offset}(k)$, and the measured value of the time deviation at kT is represented by $T_{offset}(k)$, then there is an observation model as follows:

$$T'_{offset}(k) = T_{offset}(k) + W(k) \tag{2}$$

where $W(k)$ denotes the measurement error, assuming that the measurement error conforms to a normal distribution, its mean is 0, and the variance is σ_w^2 white noise, which can be obtained from the test data using statistical principles.

In the above three time-aware systems A, B, and C, time-aware system A is the master node, assuming that the clock frequency of its clock is f and there is no noise influence, that is, the clock frequency of the master clock is unchanged, and $f(k)$ is the clock frequency of the slave node at kT time, $f(k)'$ is the frequency drift rate of the slave node, and the formula for the frequency drift rate is as follows:

$$f(k)' = r(k) + V(k) \tag{3}$$

Among them, $r(k)$ is the normal drift of the frequency, which is affected by the external environment such as temperature and crystal aging, and $V(k)$ is the random drift of the frequency, which can be regarded as a zero mean and a variance of σ_V^2 white noise. Under normal operating conditions, $r(k)$ can be regarded as a constant value for a certain period of time in close proximity, so the expressions for the relationship between the true value of clock deviation and frequency and the relationship between frequency and frequency drift are shown:

$$T'_{offset}(x) = \int_0^x [f(k) - f]\, d\tau + \mu_0 \tag{4}$$

$$f(x) = \int_0^x f'(k)\, d\tau + \delta_0 \tag{5}$$

μ_0 is the initial value of the clock offset, δ_0 is the initial value of the frequency, the two formulas are discretized, and then they are discretized, and the expression can be obtained as follows:

$$\begin{cases} \begin{bmatrix} T'_{offset}(k+1) \\ f(k+1) - f \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T'_{offset}(k) \\ f(k) - f \end{bmatrix} + \begin{bmatrix} 0.5T^2 \\ T \end{bmatrix} r(k) + \begin{bmatrix} 0.5T^2 \\ T \end{bmatrix} V(k) \\ T_{offset}(k+1) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} T'_{offset}(k) \\ f(k) - f \end{bmatrix} + W(k) \end{cases} \tag{6}$$

The system state $X(k+1)$ defined at the sampling time $(k+1)T$ is the true value and frequency drift of the clock deviation, $Y(k+1)$ is the observation signal of the corresponding state, the state transition matrix is A, the control matrix and the noise-driven matrix are B, τ, and the observation matrix is H.From the discrete expression, the state-space equation of the Kalman criterion can be deduced as follows:

$$\begin{cases} X(k+1) = AX(k) + Br(k) + \tau V(k) \\ Y(k+1) = HX(k) + W(k) \end{cases} \tag{7}$$

The parameters are as follows: $\begin{cases} X(k+1) = \begin{bmatrix} T'_{offset}(k+1) \\ f(k+1) - f \end{bmatrix} \\ Y(k+1) = [T_{offset}(k+1)] \end{cases}$, $A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$, $B = \tau = \begin{bmatrix} 0.5T^2 \\ T \end{bmatrix}$, $H = [1 \quad 0]$.

3.2. Kalman filtering process

According to the state-space equation, the time deviation can be filtered, and the filtering is mainly divided into two steps, the first step is to predict the state at the $(k+1)T$ time, and the second step is to update the state at the $(k+1)T$ time.

(1) Prediction of the state at $(k+1)T$ time:

$$\begin{cases} X(k+1|k) = AX(k|k) + Bu(k) \\ P(k+1|k) = AP(k|k)A^T + Q \end{cases} \tag{8}$$

where $X(k+1|k)$ is the prior state prediction at $(k+1)T$ time, and $P(k+1|k)$ is the prior estimate of the covariance, which is an unreliable prediction based on the results of the previous iteration and is the input in the state update equation. $X(k|k)$ is the posterior state prediction at kT time. $P(k|k)$ is the posterior estimation covariance, which is calculated according to the posterior state prediction and posterior estimation covariance at $(k-1)T$ time. The prior state prediction and prior estimation covariance at kT time, which is the optimal solution of kT time, that is, the final filtering result. As for the initial value of the state-space equation, as long as it is not set to 0. $u(k)$ is the control quantity at kT time, and there is $u(k) = r(k) + V(k)$, Q is the process noise covariance matrix, and $Q = \begin{bmatrix} \sigma_W^2 & 0 \\ 0 & \sigma_V^2 \end{bmatrix} T$.

(2) Status updates at $(k+1)T$ time:

$$K(k+1) = P(k+1|k)H^T[HP(k+1|k)H^T + R]^{-1} \tag{9}$$

$$\begin{cases} X(k+1|k+1) = X(k+1|k) + K(k+1)(Y(k+1) - HX(k+1|k)) \\ P(k+1|k+1) = (I - K(k+1)H)P(k+1|k) \end{cases} \tag{10}$$

where $K(k+1)$ denotes the Kalman gain at time $(k+1)T$, and R depends on the measurement noise covariance σ_m^2 associated with the measurement device. $X(k+1|k+1)$ represents the posterior state prediction at time $(k+1)T$, and $P(k+1|k+1)$ is the posterior estimated covariance, which is the optimal prediction at time $(k+1)T$.

The whole process of Kalman filtering is shown in Figure 3, the specific steps of Kalman filtering are to first give the initial state, that is, the clock deviation, frequency drift value and covariance at the time of $k = 0$, and the initial value can be arbitrarily given. Then, by the recursive Formula (10), the prior state prediction and prior estimated covariance of $(k+1)T$ time are obtained, and then the Kalman gain of $(k+1)T$ time is updated according to the prior estimated covariance, and the Kalman gain determines the weights of the prior state prediction and the posterior state prediction of time at $(k+1)T$ time. Finally, according to Formula (12), the posterior state prediction and posterior estimation covariance of the $(k+1)T$ moment are obtained, and a filtering process is completed.

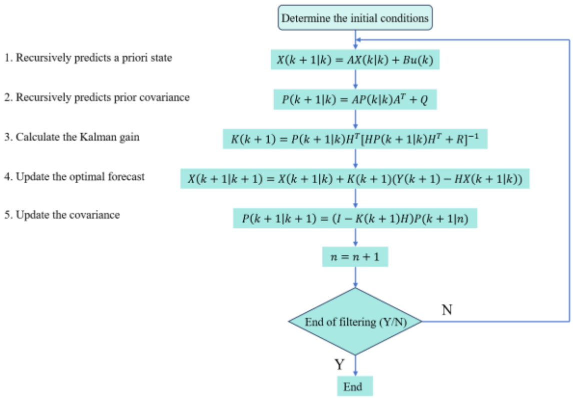
Determine the initial conditions
1. Recursively predicts a priori state
$X(k+1|k) = AX(k|k) + Bu(k)$
2. Recursively predicts prior covariance
$P(k+1|k) = AP(k|k)A^T + Q$
3. Calculate the Kalman gain
$K(k+1) = P(k+1|k)H^T[HP(k+1|k)H^T + R]^{-1}$
4. Update the optimal forecast
$X(k+1|k+1) = X(k+1|k) + K(k+1)(Y(k+1) - HX(k+1|k))$
5. Update the covariance
$P(k+1|k+1) = (I - K(k+1)H)P(k+1|n)$
$n = n+1$
End of filtering (Y/N)
N
Y
End

Figure 3: Kalman filtering process.

4. Experiment and Result Analysis

4.1. TSN time synchronization based on the AUTOSAR specification

AUTOSAR (Automotive Open system Architecture) is an open automotive electronic system architecture that improves development efficiency and product reliability by standardizing the connection and interaction between hardware and software of automotive electronic systems. In AUTOSAR, time synchronization is a very important function that ensures the synchronization of clocks between the various ECUs (Electronic Control Units) in order to implement various time-related functions such as timestamps, timeout calculations, etc. In the AUTOSAR standard specification, the *EthTSyn* module is a unique module used to implement the time synchronization protocol based on automotive Ethernet, which cooperates with the time synchronization management module *StbM* module to jointly implement a complete automotive Ethernet time synchronization scheme.

4.2. Build a hardware platform

The hardware used in this lab is an Infineon Aurix2G TC397 development board and an NXP SJA1110 switch. Using the Vector toolchain, configure the AUTOSAR-based time synchronization feature on Davinci and the SJA1110 switch project on the S32 Design Studio IDE.

4.3. Analysis of results

The working frequency of the master-slave clock is set to 150MHz, and the synchronization period is selected as 125ms, the time deviation between the two nodes can be observed by Lauterbach on the host computer, and the time deviation can also be observed from the PPS output of the two development boards with an oscilloscope, as shown in Figure 4.

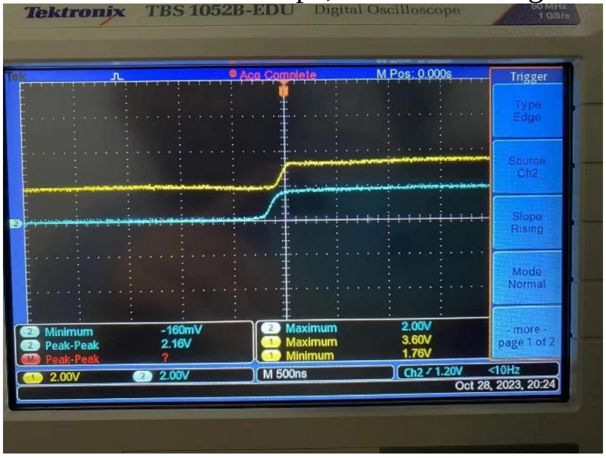

Figure 4: Oscilloscope observation synchronization accuracy.

The time deviation before and after the Kalman filter is observed in Figure 5 below. The estimated value of the state is obtained by Kalman filter estimation, so that the deviation of the slave clock is compensated, and the clock model is corrected, and finally the synchronization of the slave clock and the master clock is realized.

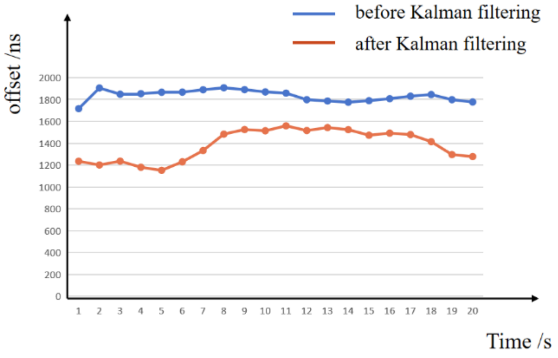

Figure 5: Time deviation before and after Kalman filtering.

5. Conclusion

The traditional clock synchronization system based on software timestamp has a simple synchronization process and low cost, but there are problems such as low synchronization accuracy and serious noise in the measurement process, and the synchronization accuracy is affected by clock offset and clock offset rate. Based on real-time Ethernet, this paper deeply explores the principle of clock synchronization, proposes a clock synchronization optimization algorithm based on Kalman filter, and uses Kalman filter to estimate the clock offset and clock offset rate, and corrects the clock to make each node realize clock synchronization, which further improves the synchronization accuracy.

References

[1] Shi Z Y. (2020). Research on real-time Ethernet clock synchronization system based on improved Kalman filter algorithm. (Master's thesis, Zhejiang Sci-Tech University).

[2] Xu, M K. (2020). Implementation and testing of IEEE 802.1AS protocol. (Master's thesis, Harbin Institute of Technology).

[3] Gardiner, Eric. The Avnu Alliance Theory of Operation for TSN-enabled Industrial Systems [J]. IEEE Communications Standards Magazine, 2018, 2(1):5-5.

[4] Lin Zhao, Feng He, et al. Comparison of Time Sensitive Networking (TSN) and TTEthernet [C]. 2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC), 23-27 Sept. 2018

[5] Kevin B. Stanton. Distributing Deterministic, Accurate Time for Tightly Coordinated Network and Software Applications: IEEE 802.1 AS, the TSN profile of PTP [J]. IEEE Communications Standards Magazine, 2018.

[6] Nikolaus Kerö, Andreas Puhm, et al. Performance and Reliability Aspects of Clock Synchronization Techniques for Industrial Automation [J]. Proceedings of the IEEE, June 2019.

[7] Nsaibi S, Leurs L, Schotten H D. Formal and Simulation-based Timing Analysis of Industrial-Ethernet Sercos III over TSN[C]. 2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT). IEEE, 2017.

[8] Zhao Q L, Dong X Z, et al. A Universal Method for Implementing IEEE 1588 with the 1000M Ethernet Interface [J]. IEEE AUTOTESTCON, 12-15 Sept. 2016.

[9] Wang H, Shao L, et al. A Global Clock Skew Estimation Scheme for Hierarchical Wireless Sensor Networks [J]. IEEE Access, 2017(5):20333-20338.

[10] Wang H, Zeng H, Wang P. Linear Estimation of Clock Frequency Offset for Time Synchronization Based on Overhearing in Wireless Sensor Networks [J]. IEEE Communications Letters, 2016, 20(2):288-291.

[11] Zhang B H. Improved Time Synchronization Algorithm Based on TPSN [J]. Computer Engineering, 2010, 36(9):109-110.

[12] Garone E, Gasparri A, Lamonaca F. Clock Synchronization Protocol for Wireless Sensor Networks with Bounded Communication Delays[J]. Automatica, 2015, 59(C):60-72.