Int. J. Simul. Multidisci. Des. Optim. **15**, 21 (2024)

https://doi.org/10.1051/smdo/2024018

IJSMDO
Available online at:
https://www.ijsmdo.org

RESEARCH ARTICLE OPEN ACCESS

Hierarchical data synchronous interaction in nonlinear complex systems

Lufeng Yuan*, Shijie Gao, Xin He, Changnian Liu, Xilei Ren, and Zhichao Fan

Beijing China-Power Information Technology Co., Ltd, Beijing 100192, China

Received: 22 March 2024 / Accepted: 2 September 2024

Abstract. Nonlinear complex systems are widely used in various industries, and the complexity and dynamism of the systems pose enormous challenges to data synchronization and interaction. This study proposes a method based on signal synchronization for data synchronization and interaction in nonlinear complex systems. The synchronization interaction path of data was analyzed, and signals were used as the carrier for data synchronization. The data synchronization interaction was carried out by combining the main synchronization signal and the auxiliary synchronization signal. Furthermore, mean filtering was introduced for signal filtering, and the main synchronization process was divided into two parts: coarse and fine tuning. The experimental results showed that the main synchronization frequency deviation of the research method was 303 Hz when the main synchronization completion rate was 100% in a −15 dB signal-to-noise ratio environment. In the analysis of data synchronization success rate, the research method achieved a highest success rate of 99.7% when the data transmission density was 50 pieces per minute. The experiment shows that the research method can effectively improve the quality and efficiency of data synchronization and interaction in nonlinear complex systems.

Keywords: Nonlinear complex systems / data synchronization / signal synchronization / mean filtering / frequency offset

1 Introduction

The continuous development of science and technology has led to the widespread application of nonlinear complex systems (NCS) in many fields, such as communication, control, bioinformatics, etc. [1,2]. In these systems, synchronous interaction of data (SIoD) is a key factor in achieving efficient operation. However, due to the dynamic characteristics and uncertainty of NCS, data synchronization and interaction face many challenges [3]. In practical applications, NCS usually has a high degree of dynamism and uncertainty, which makes SIoD very difficult. For example, in communication systems, due to the time-varying characteristics of the channel and noise interference, the data at the receiving end often experiences distortion and delay, which affects the normal operation of the system [4]. Common SIoD methods include direct synchronization and sliding window based methods. The former has convenience in implementation, but has performance bottlenecks when dealing with complex business. The latter has high complexity in window selection [5]. Signal synchronization technology (SST) can utilize the relationship between the sender and receiver to achieve synchronous interaction of data, but there is a certain level of noise interference. Mean filtering (MF) can eliminate noise effects through data smoothing. In this context, this study attempts to propose a SIoD method based on SST and introduces MF for optimization. It is expected to propose an innovative NCS hierarchical data synchronization interaction (HDSI) method, improve the quality of data synchronization, and provide a certain technical reference for the communication industry.

The research content is mainly divided into four parts. Section 2 discusses the relevant research results of SIoD and SST. Section 3 has designed the HDSI method based on SST. Section 4 is an analysis of the effectiveness of the research method. The fourth part discusses and summarizes the entire text.

2 Related works

The advancement of computer technology has made the data content and functional modules contained in more and more control and management systems more complex. Experts and scholars are beginning to realize the technical importance of complex systems. Some scholars have conducted relevant research on SIoD. González Zapata

* e-mail: yuanlufeng@mju-edu.cn

et al. proposed a method based on artificial neurons to address the problem of machine image SIoD. This method used chaotic neurons to achieve machine to machine proxy, and had good data transmission performance [6]. Scholars such as Cai proposed a method based on non volatile memory for data synchronization and recovery in memory. Its garbage collection of database memory leakage information could effectively improve the performance of data synchronization [7]. Zhou et al. proposed a method based on intermittent control for data synchronization in complex networks. This method used the complex generalized inventory cycle law formula to prevent the separation of real and imaginary parts of the data, and had good data synchronization efficiency and quality [8]. Arokia Samy and other experts proposed a method based on non fragile sampling data control to address the issue of data synchronization in circuit systems. This scheme analyzed the system state under time-varying delays and could effectively improve data synchronization consistency [9]. Hu et al. proposed a method based on projection matrix system to address the problem of projection data synchronization. This method transformed the synchronization problem into an error system adjustment problem, constructed the control input using a constructor, and recognized the dynamic characteristics of the position using a recognizer. Experiments have shown that the proposed method can effectively reduce data synchronization errors [10].

Some scholars have conducted relevant research on SST. Scholars such as Lu proposed a signal synchronization based method to address the security issues of complex systems, collecting binomial model parameters and observation times during the process, and utilizing signal synchronization clock bias for attack mining. The proposed method could effectively reduce the threat of system attacks [11]. Gu et al. proposed an SST-based method to address the issue of data attacks in remote wide area networks. This method incorporated an integrated gateway to track the natural frequency deviation of terminal devices, which can effectively eliminate network anomalies [12]. Ge et al. designed an SST-based method to study the current signal propagation process in feedforward neural networks for information transmission. This method simplified the structure of feedforward neural networks and had good information transmission performance [13]. Kumar and other experts constructed an SST-based way for controlling chaotic satellite systems, which combines the presence of chaos to achieve feedback adaptive control of satellite systems. This method had good satellite control quality [14]. Aliabadi et al. proposed an SST-based plan to handle the matter of encrypted data communication. It used a graph text quantum neural network to analyze the uncertainty of the system receiver, and calculated the linear relationship between the data output and parameters using Taylor series expansion. The results have verified that the designed method can effectively improve the communication accuracy of encrypted data [15].

In summary, SST has been studied and applied in many fields, and some studies have proven that SST can be applied in data related fields. However, there is currently a lack of research linking SST with SIoD. In view of this, this study explores HDSI technology under NCS based on SST, in order to provide feasible technical references for data synchronization.

3 Design of HDSI method based on SST under NCS

HDSI, as a key technology, is of great significance for achieving reliable information transmission. This section will focus on the technical content of researching and designing HDSI methods under NCS.

3.1 HDSI problems and signal generation techniques in NCS

In NCS, the amount of data is usually very large, which can lead to increased delays in data synchronization and interaction, thereby affecting the real-time and stability of the system [16,17]. SST can effectively process a large amount of data. By decomposing the data into multiple parts and processing them separately, it can effectively handle complex data structures and reduce data processing latency [18,19]. This study takes the Internet of Things (IoT) system as the object of NCS, and studies the SIoD method under NCS based on SST. Figure 1 shows the synchronous interaction path of data in the IoT downlink.

In Figure 1, when the data starts synchronous interaction, it first enters the transmitter for processing. Based on the data, cyclic redundancy check is used to generate segment fixed value check codes, which is convenient for detecting data errors in the future. Afterwards, channel encoding is implemented using tail biting convolutional encoding. Then perform data block subframe retransmission and pseudo random code scrambling. After wireless transmission, the data enters the receiver and is first pre decoded and de layered. Perform multiple operations in the opposite direction to the transmitter to complete data reception. When conducting data downlink synchronous transmission, cellular networks are used to divide the data into multiple levels. When performing orthogonal phase shift keying modulation, data synchronization is carried out in a symbol by symbol format, and the transmitted baseband signal is equation (1).

$$s(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N_u-1} c_n \exp\left(\frac{j2\pi kn}{N}\right). \tag{1}$$

In equation (1), $s(k)$ represents the baseband signal. N represents the number of inverse Fourier transform points. N_u represents the number of subcarriers. c_n represents the modulation data on the n-th subcarrier. j is an imaginary unit. k represents a time variable. If there is a timing error during SIoD, and the cyclic prefix length (CPL) is greater than the timing error, the receiver receives the frequency domain signal (FDS) as shown in equation (2).

$$\widehat{\mathrm{S}}(\mathrm{k}) = \exp\frac{(2\pi\xi_K\Delta\theta)}{\mathrm{N}}\mathrm{S}(\mathrm{k})\mathrm{H}(\mathrm{k}) + \mathrm{W}(\mathrm{k})\,. \tag{2}$$

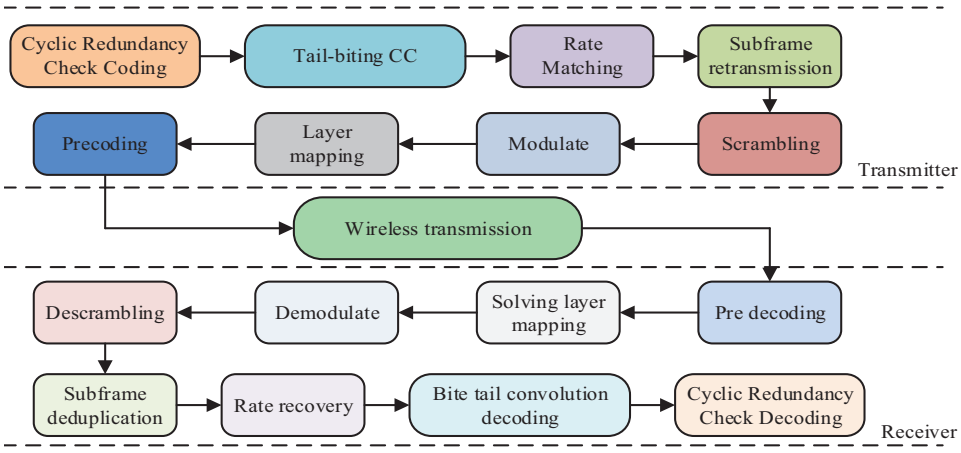

Fig. 1. Downlink data synchronization interaction path.

In equation (2), $\hat{s}(k)$ represents the receiver receiving FDS. $H(k)$ represents the channel system function. $S(k)$ represents the frequency domain baseband signal. $W(k)$ represents white noise interference. $\Delta\theta$ represents timing error. When the CPL is less than the positioning error, the receiver needs to add signal attenuation when receiving FDS, as shown in equation (3).

$$\alpha(\Delta\theta) = \sum_i |h_i(t)|^2 \frac{N - \Delta\theta}{N}. \tag{3}$$

In equation (3), $\alpha(\Delta\theta)$ represents the attenuation of the signal by the channel. i represents the number of multipath channels. $h_i(t)$ represents the channel impulse response. When performing signal synchronization, the device needs to first detect the main synchronization signal (MSS), generate a signal sequence from the data, perform resource mapping, and then generate a baseband signal. The generation of MSS is equation (4).

$$d_l(n) = s(l) \cdot e^{-j\frac{nun(n+1)}{11}}. \tag{4}$$

In equation (4), $d_l(n)$ represents MSS. $s(l)$ represents code coverage. l is a symbol number between 3 and 13. u represents the root sequence number. Figure 2 is an example mapping of frequency domain resources for the main signal.

In Figure 2, the main signal resource mapping number in the frequency domain starts from 0 and ends at 11, corresponding to the subcarriers in the frequency domain. Subcarrier numbers are divided into a location area every 7. The main signal is distributed on subframe 5 of the wireless frame 10. In the generated example graph, the blue block represents a code coverage value of 1, and the green block represents a value of −1. The generation function of the auxiliary synchronization signal (ASS) is related to the position of the data device, as shown in equation (5).

$$d(n) = b_q(m) e^{-j^{2\theta\pi f^n}} e^{-j\frac{nun'(n'+1)}{131}}. \tag{5}$$

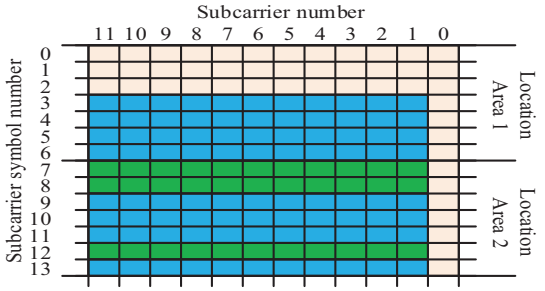

Fig. 2. Main signal frequency domain resource mapping.

In equation (5), $d(n)$ represents ASS. $b_q(m)$ represents the Hadamard matrix row. θ_f represents the cyclic displacement. Figure 3 shows the generation process of ASS.

In Figure 3, during the generation of ASS, scrambling sequences and Zadoff-chu sequences are generated from the location of the data device, and the length of the Zadoff-chu sequence is extended. After combining two sequences and adding frame timing for cyclic displacement, the data is divided into multiple sequences for subcarrier mapping, inverse Fourier transform, and cyclic prefix operation. SIoD is completed by combining MSS and ASS.

3.2 Optimization design of HDSI method by introducing MF and partitioning processing

When there is a large frequency offset in the SIoD process, the zero autocorrelation characteristic in the Zadoff-chu sequence will be destroyed, resulting in data synchronization errors [20]. Before performing SIoD, the large frequency offset group (LFOG) can be preset. After completing the preset, the processing flow of the LFOG is Figure 4.

In Figure 4, when performing LFOG processing, it is first necessary to determine the frequency offset range of the working environment, and then divide the LFOG and generate a reference signal. Search for the time offset

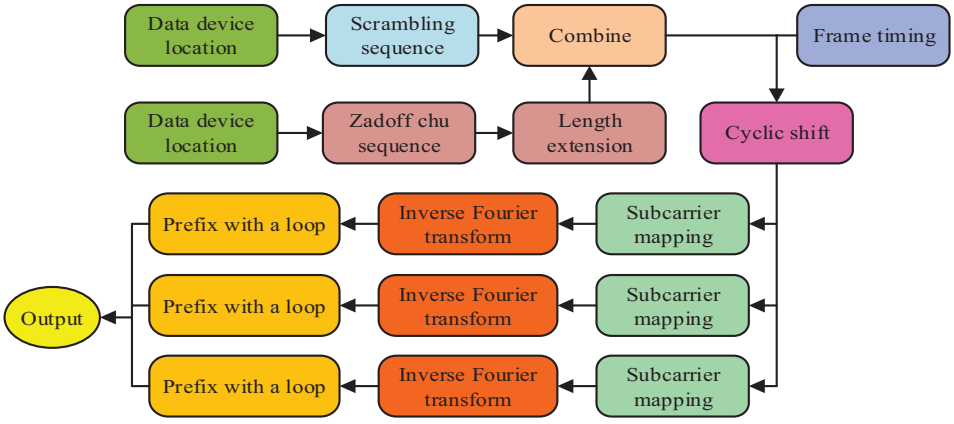

Fig. 3. ASS generation process.

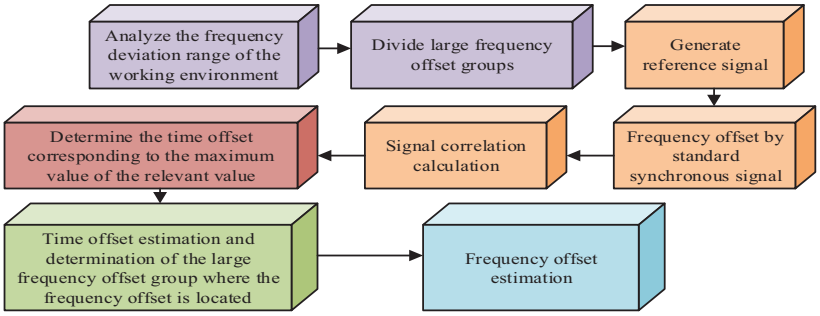

Fig. 4. Large frequency offset group processing flow.

corresponding to the maximum value of the relevant value, then perform time offset estimation and determine the LFOG where the frequency offset is located, and finally complete the frequency offset estimation (FOE). Eliminating the estimation error of SIoD through the estimation of FOE and LFOG. During signal transmission, the noise present in the transmission environment can cause interference to the data. If the interference is too severe, it may cause the signal to be annihilated. Although the Zadoff chu sequence can perform simple processing on noise interference, when the signal-to-noise ratio (SNR) is too low, there may also be significant estimation errors in frequency and time offset, leading to SIoD failure. So signal filtering operation is also required during data synchronization. This study selected MF with lower complexity for signal filtering. The implementation of MF is equation (6).

$$y(n) = \frac{1}{n}\sum_{k=0}^{N} x(n-k). \tag{6}$$

In equation (6), $y(n)$ represents the MF result. x represents the sampled signal. N represents the window size. The frequency response of the mean filter is equation (7).

$$H(e^{j\omega}) = \frac{1}{N}\cdot\frac{1-e^{-j\omega N}}{1-e^{-j\omega}}. \tag{7}$$

In equation (7), $H(e^{j\omega})$ represents the mean filter frequency response. ω represents the signal angle. In response to the computational complexity and accuracy issues during SIoD, this study divides the main synchronization process into two parts: coarse and fine tuning. Figure 5 shows the coarse tuning main synchronization (CTMS) process.

In Figure 5, when conducting CTMS, it is necessary to first perform sliding cross correlation (SCC) calculation. Afterwards, using a cost function to weight the autocorrelation results of each symbol, and judge whether the peak has reached the threshold based on the cost function value. After the peak reaches the threshold, estimate the signal time and frequency offset to complete the approximate positioning of the MSS. When calculating SCC, the SCC calculation for each sampling point is equation (8).

$$y(t) = \sum_{n=0}^{N-1} x[t+n]NPSS^{*}_{symbol}\exp\left(\frac{2j\pi}{N}\cdot f_n(t+n)\right). \tag{8}$$

In equation (8), $y(t)$ represents the SCC value. t represents the relative offset between the selected sampling point and the actual MSS position. $NPSS_{symbol}$ represents the content of a subcarrier symbol length in MSS. To split and calculate SCC, as shown in equation (9).

$$y(\tau,k) = \sum_{n=0}^{N-1} x[\tau+kN+n]NPSS^{*}\exp\left(\frac{2\pi j}{N}\cdot f_n(\tau+kN+n)\right). \tag{9}$$

In equation (9), τ represents the decimal time deviation. When the sum of the product of sampling points and offset plus decimal time offset is 0, the term unrelated to

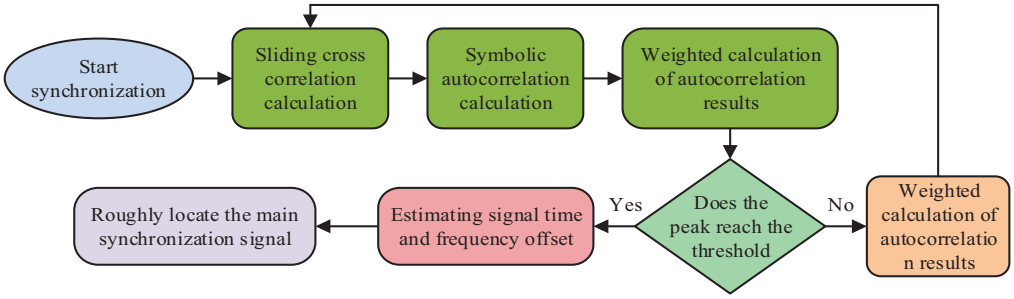

Fig. 5. Roughly adjust the main synchronization process.

frequency offset in the split calculation reaches its maximum value. At this point, the time offset estimation result can be obtained through SCC calculation. The autocorrelation calculation of each symbol when the time offset is 0 is equation (10).

$$S(m) = (11 - m)\exp(2\pi jmf_n). \tag{10}$$

In equation (10), $S(m)$ represents the corresponding code coverage value of a subcarrier symbol in MSS. m represents the interval between autocorrelation calculation values. The cost function in weighted calculation is equation (11).

$$\rho(t) = S(1) + 3S(2)S^*(1) + S(3)S^*(2). \tag{11}$$

In equation (11), $\rho(t)$ represents the cost function. The cost function is used to determine whether the peak has reached the threshold, as shown in equation (12).

$$T = \rho_{\max} - \rho_{left} - \rho_{right}. \tag{12}$$

In equation (12), T represents the relationship between peak and threshold. When it is greater than 0, it is determined that the peak has reached the threshold. ρ_{left} represents the cost function sum of the sampling points on the left side of the sampling point corresponding to the maximum value. ρ_{right} represents the cost function sum of the sampling points on the right side of the sampling point corresponding to the maximum value. Roughly to adjust MF as shown in equation (13).

$$x(n) = \alpha x_{t+1}(n) + (1 - \alpha)x_t(n). \tag{13}$$

In equation (13), $x(n)$ represents the result of coarse adjustment of MF. α represents the MF factor. The coarse tuning part reduces the estimation complexity of time and frequency offsets, and then fine tuning the main synchronization is carried out on the basis of CTMS. Firstly, perform fine-tuning symbol merging, use vectors to represent the sampled sample set, and add code coverage to the vectors and sum them, as shown in equation (14).

$$A(\tau) = \frac{1}{11}\sum_{m=1}^{11} s(m)R_m. \tag{14}$$

In equation (14), $A(\tau)$ represents the sum of sub vectors. $s(m)$ represents the sub vector code coverage value. R_m represents the sampling sample. Afterwards, the sampling point position is obtained through cross correlation calculation, and then time offset and FOE are performed. When performing auxiliary synchronization, the cross correlation results of the received FDS and ASS are analyzed, and the noise from the multi frame detection and multi transmission process is processed. The average cross correlation value is calculated as shown in equation (15).

$$\mathrm{X}(\overline{\mathrm{NID}}) = 2\mathrm{Nf}\sum \mathrm{nf} = 0\mathrm{NfX}(\mathrm{nf}, \mathrm{NID}) \tag{15}$$

In equation (15), X(NID) represents the average cross correlation value of multiple consecutive frames. Figure 6 shows the SIoD process for this design.In Figure 6, after starting SIoD, the device first sends data, while performing both primary and secondary synchronization stages. When performing auxiliary synchronization, it is necessary to pre store the possible starting position of the ASS, then locate the starting position of the ASS in the received signal, and output the detection result after cross correlation calculation. During the main synchronization, CTMS is performed first. During the circuit comparison, the peak value is input into FOE and the MSS position is directly sent to the symbol merging step of the fine tuned main synchronization. After completing the CTMS, fine tune the main synchronization, eliminate noise, and reduce time and frequency offsets. Then, synthesize the main synchronization results and auxiliary synchronization results to complete the HDSI under NCS.

4 Effectiveness analysis of SST-based HDSI method under NCS

The HDSI of NCS involves multiple levels of data transmission and processing, which is of great significance for achieving efficient and accurate synchronization. This section will analyze the effectiveness of research and design HDSI methods from both performance testing and application analysis.

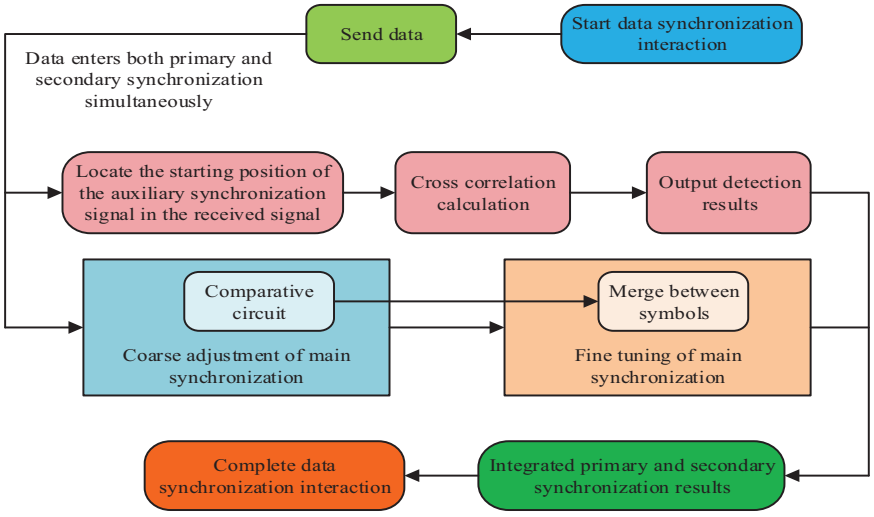

Fig. 6. Data synchronization interaction process.

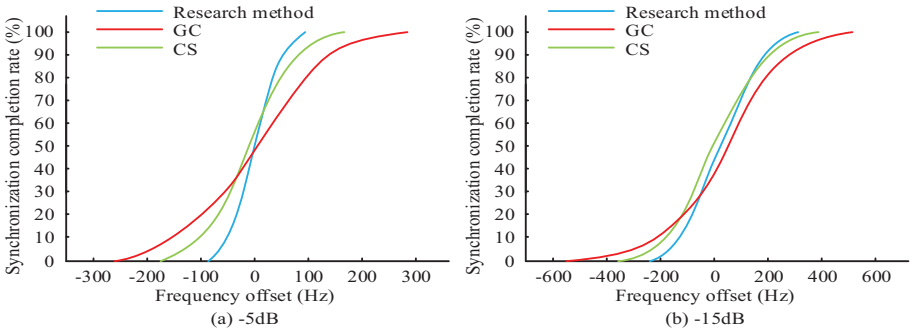

Fig. 7. Main step frequency offset estimation performance.

4.1 Performance testing of SST-based HDSI method under NCS

In order to test the effectiveness of the designed analysis technology synchronous interaction method under NCS, simulation analysis technology was used in performance testing in this study. Two sets of tests with SNRs of −5 dB and −15 dB were set for the environment. To reduce data anomalies caused by randomness during simulation, 100 simulation data were selected for comprehensive analysis [21]. Comparing with Group Communications (GC) and Cell Search (CS) methods during analysis. The GC and CS methods have been widely applied in existing communication systems, representing mature technologies in the fields of group communication and cell search, respectively. And these two methods have been fully validated in the research and practice of the communication field, with high technological maturity, which can provide a stable and reliable comparative basis for research, and have high representativeness and reference value. Test the performance of the main synchronization FOE, as shown in Figure 7.

In Figure 7, during the main synchronization FOE, the main synchronization frequency deviation (MSFD) within a certain range of different methods remains within a certain range during the increase in synchronization completion rate. In the −5 dB SNR environment, the MSFD of GC is 281 Hz when the main synchronization completion rate (MSCR) is 100%. The research method has an MSFD of −89 Hz at 0% MSCR and 93 Hz at 100% MSCR. In a −15 dBSNR environment, the MSFD of CS at MSCR of 100% is 389 Hz. The research method has an MSFD of −234 Hz at 0% MSCR and 303 Hz at 100% MSCR. This indicates that the research method can maintain lower bias when performing main synchronization FOE. Figure 8 shows the test results of the number of frames used for main synchronization.

In Figure 8, the main synchronization accuracy of different methods increases as the number of frames increases. In the −5 dBSNR environment, the accuracy of CS increased to 100% after 48 frames, while the accuracy of GC increased to 100% after 22 frames. The initial synchronization accuracy of the research method is 0%, which increases to 100% after 5 frames. In the −15 dBSNR environment, the initial synchronization accuracy of GC is 11%, which increases to 94% after 50 frames. The initial synchronization accuracy of the research method is 0%, which increases to 100% after 33 frames. This indicates that the research method has better accuracy and efficiency. Use an Intel i5-12400 computer processor with 16 Gb of RAM at 3200 MHz and a 512 Gb solid-state drive for hardware resource utilization testing. Ensure sufficient

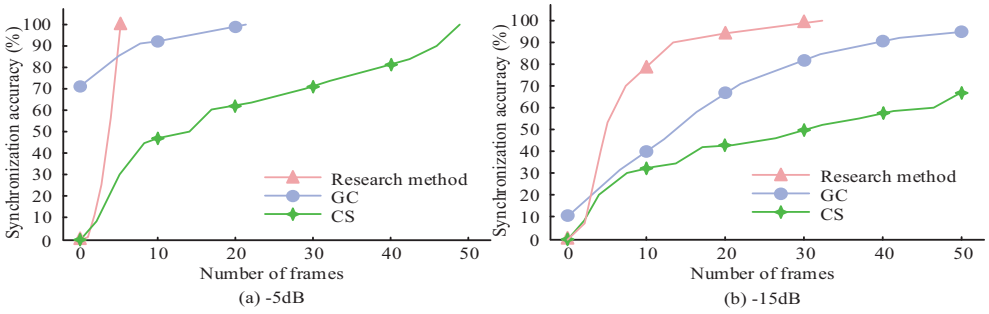

Fig. 8. Main synchronization frame count test.

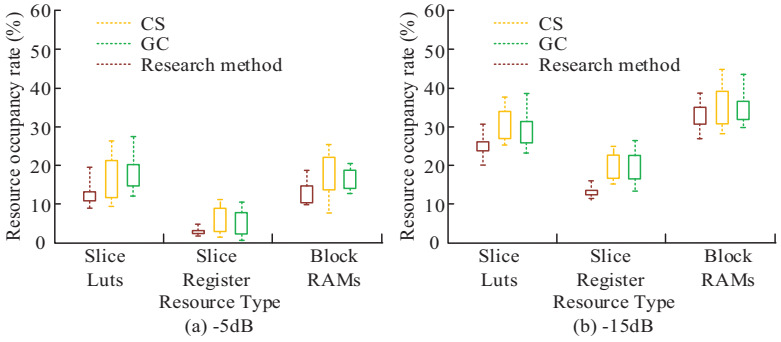

Fig. 9. Hardware resource utilization test.

power supply during testing and use a water cooling system to ensure good heat dissipation of the equipment. Figure 9 is a test of the hardware resource utilization during different method runs.

In Figure 9a, the resource occupancy rate in the −5 dBSNR environment is approximately 5.5% for CS's Slice Register and 18.0% for Block RAMs. The Slice Luts of GC is approximately 17.8%, and the Block RAMs are 16.7%. The research method's Slice Luts is 11.3%, Slice Register is 2.1%, and Block RAMs is 12.1%. In Figure 9b, the resource occupancy rate in the −15 dBSNR environment is approximately 30.7% for CS Slice Luts and 34.9% for Block RAMs. GC's Slice Luts is 28.7%, and Slice Register is 19.7%. The research method's Slice Luts is approximately 25.1%, Slice Register is 12.5%, and Block RAMs is 13.2%. The data indicates that the research method consumes less resources during runtime.

4.2 Application analysis of SST based HDSI method under NCS

When conducting application analysis, two NCS systems containing multiple devices in wireless interconnected network environments were selected for practical application, labeled as Alpha and Bravo, respectively. The Alpha system consists of intelligent sensors, data collectors, mobile terminals, and central processing units. Intelligent sensors are responsible for collecting environmental data, data collectors are responsible for preliminary processing of sensor data, mobile terminals provide user interaction interfaces, and central processing units are responsible for final data processing and decision-making. Bravo system is composed of intelligent surveillance camera, wireless router, edge computing node and cloud server. Intelligent surveillance cameras are used to capture video data in real time. Wireless routers provide network connectivity between devices. Edge computing nodes process data locally to reduce latency. ECS is used for large-scale data storage and complex computing tasks. Use RF modules for signal transmission to achieve HDSI in the environment. Analyze the circuit power consumption (CPC) during HDSI, as shown in Figure 10.

In Figure 10, the CPC of different methods in HDSI increases with the increase of data transmission density (DTD). Regarding the CPC of DTD in the Alpha system, CS increases to 1.31 W when it reaches 250 lines per minute (L/min). GC rose to 0.89 W when it reached 250 bars per minute. The power consumption of the research method is 0.09 W at 50 bars per minute, and it increases to 0.41 W when it reaches 250 bars per minute. In the Bravo system, the power consumption of the DTD circuit increases to 1.19 W when the density increases to 250 L/min, and to 1.08 W when the density increases to 250 L/min. The power consumption of the research method is 0.15 W at a density of 50 L/min, and increases to 0.40 W when it reaches 250 L/min. This indicates that the research method has better power consumption performance in actual operation. Analyze the success rate of data synchronization, as shown in Figure 11.

In Figure 11, the success rate of data synchronization (SRoDS) for different methods decreases as DTD increases during data synchronization. In the Alpha system, the SRoDS of CS and GC decreased to 86.2% and 78.8% at a DTD of 300 L/min. The SRoDS of the research method was

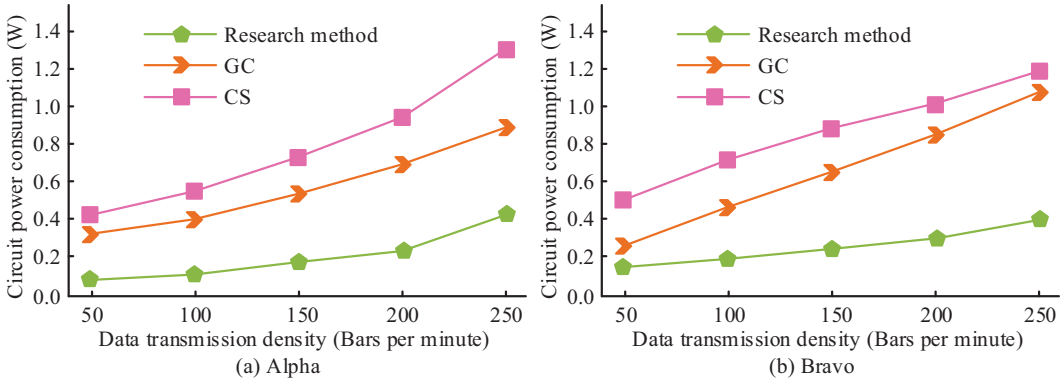

Fig. 10. Circuit power consumption.

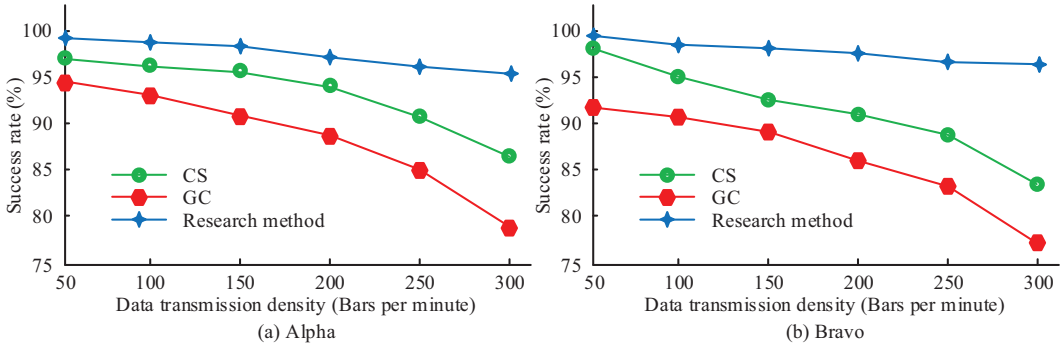

Fig. 11. Data synchronization success rate.

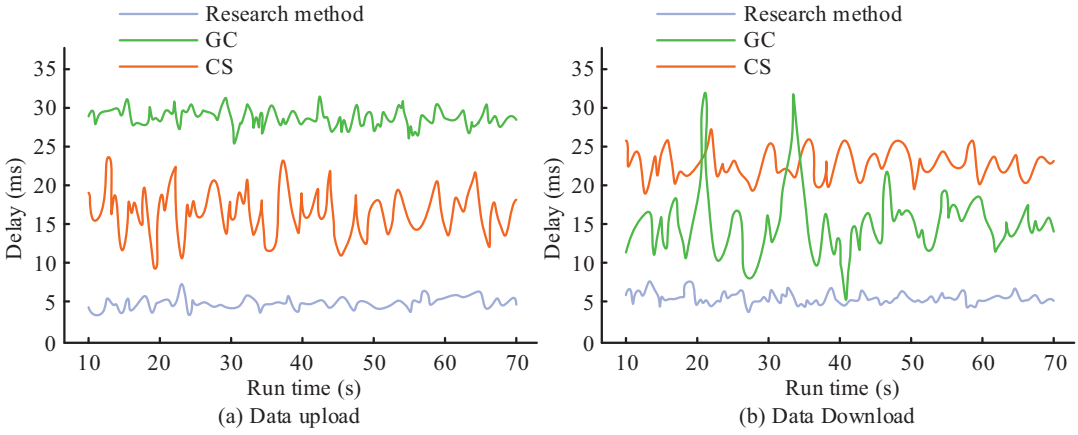

Fig. 12. Data synchronization interaction delay.

99.1% at a DTD of 50 L/min, and decreased to 95.2% at 300 L/min. In the Bravo system, the SRoDS of CS and GC decreased to 83.5% and 77.2% at a DTD of 300 L/min. The SRoDS of the research method was 99.7% at a DTD of 50 L/min, and decreased to 96.3% at 300 L/min. The above data indicates that research methods can better maintain the success rate of SIoD. Analyze the delay during SIoD in the Alpha system, as shown in Figure 12.

In Figure 12, the SIoD delay of different methods fluctuates within a certain range. In Figure 12a, the delay of CS and GC during data upload fluctuates between 9.2 ms to 23.8 ms, 25.2 ms to 31.0 ms, and the research method fluctuates between 3.9 ms and 7.1 ms. In Figure 12b, the latency of CS and GC during data download fluctuates between 19.1 ms to 27.1 ms, 5.2 ms to 32.0 ms, and the research method fluctuates between 4.0 ms and 7.6 ms. This indicates that research methods can complete SIoD tasks in a faster and more stable manner.

5 Conclusion

To better understand and optimize NCS, research on SIoD has become a crucial issue. This study proposed a method based on signal synchronization to handle the SIoD

problem in NCS. The HDSI problem under NCS was analyzed and a signal generation technique for signal synchronization was designed. Then, the LFOG and filtered the signal were presented. Finally, the effectiveness of the research method was verified. The experiment showed that the synchronization accuracy of the research method increased to 100% after 33 frames in a −15 dBSNR environment when testing the number of frames used for main synchronization. In the hardware resource utilization test, the research method achieved a Slice Luts resource utilization rate of approximately 11.3% in the −5 dBSNR environment. When conducting CPC analysis, the research method maintained CPC below 0.41 W at a DTD of 250 L/min. When conducting SRoDS testing, the research method maintained a SRoDS of over 95.2% at a DTD of 300 L/min. The download delay of the research method fluctuated between 4.0 ms and 7.6 ms. The above results indicated that the research method has better SIoD speed and can ensure more accurate SIoD results. However, this study was only tested in a small data volume environment, and the stability in a large data volume environment has not been analyzed. In the future, the experimental scope will be expanded to enrich the experimental results and optimize the methods.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflicts of interest

The author declares that there is no conflicts of interest.

Data availability statement

The data used to support the findings of the research are available from the corresponding author upon reasonable request.

Author contribution statement

Lufeng Yuan: Study design, data collection, statistical analysis, visualization, writing the manuscript. Shijie Gao: data collection, statistical analysis. Xin He: visualization. Changnian Liu: statistical analysis, revised the manuscript. Xilei Ren: statistical analysis, revised the manuscript. Zhichao Fan: Revised the manuscript, led and supervised this study.

References

1. G. Zhang, J. Zhang, W. Li, C. Ge, Y. Liu, Exponential synchronization of delayed neural networks with actuator failure using stochastic sampled-data control, Int. J. Control Automat. Syst. **20**, 691–701 (2022)
2. H. Wang, Y. Ni, J. Wang, J. Tian, C. Ge, Sampled-data control for synchronization of Markovian jumping neural networks with packet dropout, Appl. Intell. **53**, 8898–8909 (2023)
3. X. Liang, C. Zhang, Y. Luo, M. Cui, K. Qiu, Secure key distribution and synchronization method in an OFDM-PON based on chaos, Opt. Express **30**, 18310–18319 (2022)
4. Y. Qin, F. Li, J. Wang, H. Shen, Extended dissipative synchronization of reaction-diffusion genetic regulatory networks based on sampled-data control, Neural Process. Lett. **55**, 3169–3183 (2023)
5. L. Li, Y. Wang, K.Y. Lin, Preventive maintenance scheduling optimization based on opportunistic production-maintenance synchronization, J. Intell. Manufactur. **32**, 545–558 (2021)
6. A.M. González-Zapata, E. Tlelo-Cuautle, I. Cruz-Vega, W.D. León-Salas, Synchronization of chaotic artificial neurons and its application to secure image transmission under MQTT for IoT protocol, Nonlinear Dyn. **104**, 4581–4600 (2021)
7. S. Cai, K. Chen, M. Liu, X. Liu, Y. Wu, W. Zheng, Garbage collection and data recovery for N2DB, Tsinghua Sci. Technol. **27**, 630–641 (2021)
8. H. Zhou, Z. Liu, W. Li, Sampled-data intermittent synchronization of complex-valued complex network with actuator saturations, Nonlinear Dyn. **107**, 1023–1047 (2022)
9. S. Arockia Samy, R. Ramachandran, P. Anbalagan, Y. Cao, Synchronization of nonlinear multi-agent systems using a non-fragile sampled data control approach and its application to circuit systems, Front. Information Technol. Electr. Eng. **24**, 553–566 (2023)
10. W. Hu, L. Gao, T. Dong, Event-based projective synchronization for different dimensional complex dynamical networks with unknown dynamics by using data-driven scheme, Neural Process. Lett. **53**, 3031–3048 (2021)
11. Q. Lu, X. Feng, C. Zhou, A detection and weakening method for GNSS time-synchronization attacks, IEEE Sens. J. **21**, 19069–19077 (2021)
12. C. Gu, L. Jiang, R. Tan, M. Li, J. Huang, Attack-aware synchronization-free data timestamping in Lorawan, ACM Trans. Sensor Networks (TOSN) **18**, 1–31 (2021)
13. M.Y. Ge, G.W. Wang, Y. Jia, Influence of the Gaussian colored noise and electromagnetic radiation on the propagation of subthreshold signals in feedforward neural networks, Sci. China Technolog. Sci. **64**, 847–857 (2021)
14. S. Kumar, A.E. Matouk, H. Chaudhary, S. Kant, Control and synchronization of fractional-order chaotic satellite systems using feedback and adaptive control techniques, Int. J. Adaptive Control Signal Process. **35**, 484–497 (2021)
15. F. Aliabadi, M.H. Majidi, S. Khorashadizadeh, Chaos synchronization using adaptive quantum neural networks and its application in secure communication and cryptography, Neural Comput. Appl. **34**, 6521–6533 (2022)
16. G. Zhang, Y. Liu, H. Yang, D. Qian, Efficient detection of silent data corruption in HPC applications with synchronization-free message verification, J. Supercomput. **78**, 1381–1408 (2022)
17. S. Choudhuri, S. Adeniye, A. Sen, Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation, Artif. Intell. Appl. **1**, 43–51 (2023)

18. H. Cao, Y. Wu, Y. Bao, X. Feng, S. Wan, C. Qian, UTrans-Net: a model for short-term precipitation prediction, Artif. Intell. Appl. **1**, 106–113 (2023)
19. J.C.J. Koelemeij, H. Dun, C.E.V. Diouf, E.F. Dierikx, G.J. Janssen, C.C. Tiberius, A hybrid optical-wireless network for decimetre-level terrestrial positioning, Nature **611**, 473–478 (2022)
20. J. Mao, Y. Sun, X. Yi, H. Liu, D. Ding, Recursive filtering of networked nonlinear systems: a survey, Int. J. Syst. Sci. **52**, 1110–1128 (2021)
21. M. Rinott, N. Tractinsky, Designing for interpersonal motor synchronization, Human-Computer Interact. **37**, 69–116 (2022)

Cite this article as: Lufeng Yuan, Shijie Gao, Xin He, Changnian Liu, Xilei Ren, Zhichao Fan, Hierarchical data synchronous interaction in nonlinear complex systems, Int. J. Simul. Multidisci. Des. Optim. **15**, 21 (2024)