Research and Design of Hybrid Scheduling Algorithm Based on LIN Bus

Zhanwei Liang¹, Hongyu Shi¹, Hao Yan¹, Dongmei Ma¹ and Jie Li¹

¹Chongqing University of Posts and Telecommunications Automation College, Chongqing 400065
E-mail: 2820429366@qq.com

Abstract: In the LIN (Local Interconnect Network) bus communication transmission, there is a problem of data transmission delay in the process of sending messages by the master-slave node ECU (Electronic Control Unit). Aiming at this kind of delay problem, this paper analyzes the characteristics of static scheduling and dynamic scheduling algorithm of LIN bus, optimizes the scheduling strategy of LIN bus message, and proposes a hybrid scheduling algorithm based on time slice scheduling. At the same time, the simulation model of LIN bus master-slave node message transmission is established, and the simulation experiment verification platform of LIN bus hardware-in-the-loop is built. It is verified that the algorithm reduces the delay by about 30%.

1. INTRODUCTION

Nowadays, the traditional data transmission method based on LIN (Local Interconnect Network) bus is the static scheduling method based on the static scheduling table to realize the message transmission. This kind of transmission method is simple and stable, but it has the problem of high transmission delay when the master node has event message to transmit. The LBFP (left-handed optimal matching algorithm) proposed by Jiang Yinfeng¹ mainly solves the problem of response time delay of message transmission between master and slave nodes ECUs, but does not solve the problem of transmission delay between messages in the transmission process. The LBFP-GA (genetic algorithm) proposed by Yin Jianli mainly solves the problem of response time delay in the transmission between messages and the problem of delay time in the receiving of messages by nodes, and also does not solve the problem of transmission delay between messages in the transmission process.

Firstly, through the research and analysis of the traditional static scheduling algorithm data transmission method, the scheduling matrix of the static scheduling table is optimized, and a hybrid scheduling algorithm is designed. The main idea is to static in the scheduling matrix in the traditional static scheduling method. The time slice is divided into a basic time slice and a periodic scheduling time slice. The time slice interleaving method is used, so that when the active node needs to transmit the event type message, the latest time slice can be found, thereby reducing the delay.

2. Analysis of traditional static scheduling algorithm

2.1. Defect analysis of traditional scheduling algorithm

As shown in Figure 1, the data transmission characteristics of the LIN bus are in the form of master-slave. The host computer sends the frame header or frame header plus data in the form of information exchange with the lower computer. After the lower computer receives the message sent by
the upper computer, it defines the database analyzes the ID (Identity Document) or ID plus data, performs corresponding operations according to the parsed result, and returns feedback data. The host computer receives the message sent by the lower computer and parses its ID and data. The upper computer functions to define the message transmission sequence and time of the LIN bus.

Fig 1. Schematic diagram of data transmission between the master and slave nodes of the LIN bus

The time that a message is transmitted on the LIN bus is determined by two factors, delay time and response time. The delay factors of the message mainly include the following parts: the message sending waiting time \( T_q \), the sending average delay time \( T_d \), the message transmission factor is \( \mu_1 \), and the verification factors of the message delay time are positive or negative. Load rates \( \rho_a \) and \( \rho_b \).

2.2. Delay analysis of traditional scheduling algorithm
The message sent from the host computer to the lower computer is a forward transmission message, and the message sent from the lower computer to the upper computer is a reverse transmission message. In the LIN network, the forward message sent by the upper computer is \( M_1, M_2, ..., M_n, M_{n+1}, ..., M_m \), and consider that all forward packets of \( M_1 \sim M_n \) are event-type messages, \( M_{n+1} \sim M_m \) all forward messages For the periodic message, the reverse message sent by the lower computer is \( S_1, S_2, S_3, ..., S_n \) and considers that all reverse messages are periodic messages. The transmission factor of the bus message is \( \mu_1, \mu_2, \mu_3, ..., \mu_n \), the transmission factor represents the average waiting delay coefficient of each frame of the message in transmission. The expressions of the positive and negative load rates \( \rho_a \) and \( \rho_b \) in the LIN network are:

\[
\rho_a = 1.4 \times \left\{ \sum_{i=1}^{n} \left[ 34 + 10N_i \right] \mu_i + \sum_{j=n+1}^{m} \left[ 36 + 10N_j \right] / M_j \right\} \times 100\% \tag{1}
\]

\[
\rho_b = 1.4 \times \left\{ \sum_{k=1}^{n} \left[ 34 + 10(N_k + 1) \right] / S_k \right\} \times 100\% \tag{2}
\]

In the above formula, the data length of \( N_i \) and \( N_j \) forward transmission messages, and \( N_k \) is the data length of the reverse transmission message.

3. Hybrid scheduling algorithm design

3.1. Hybrid scheduling algorithm design ideas
Since the data transmission mode based on the LIN bus is based on the static scheduling of the static schedule, when the event message needs to be transmitted by the host computer, it still needs to wait for the corresponding sequence in the static scheduling to transmit, which reduces the delay of the event type message, so this paper proposes a hybrid scheduling algorithm to improve this situation. The scheduling diagram of the hybrid scheduling algorithm is as follows:
The basic time slice is sent in the forward transmission

\[ \rho = \frac{3}{T_s - T_a} \]

The average delay time of the periodic message sent by the hybrid schedule is as follows:

\[ T_d = \frac{\rho_d T_a}{m} + \rho_b T_s \]

It can be seen from the above equation that the average message transmission waiting time of the hybrid schedule is almost determined by the periodic message transmission time.

The average delay time \( T_d \) of the periodic message sent by the hybrid schedule is as follows:

\[ T_d = \frac{\rho_d T_a}{m} + \rho_b T_s \]
The average delay time \( T_l \) of the event type message sent by the hybrid schedule is as follows:

\[
T_l = nT_s + \sum_{i=1}^{n} C_i
\]

(7)

4. Hybrid scheduling algorithm delay optimization verification

The communication mode of LIN bus is the main form. In this paper, the communication between the upper computer and the lower computer is used for analog verification. The upper computer is PC and the lower computer is an ECU controller. This paper takes an ECU controller as an example and the upper computer. The LIN bus communication environment is built with CANcaseXL, and the hybrid scheduling algorithm designed in this paper is used to verify the algorithm feasibility.

In this system, the baud rate is 19200 bps, the base time slice \( T_s \) takes 20 ms, the hybrid scheduling table \( T_x \) takes 500 ms, and the basic message transmission basic cycle \( T_a \) takes 100 ms.

Figure 3 shows the test sample topology. The master node is BCM, the slave node is DPWM, the BCM optional periodic message has 8 frames, the optional event type message has 5 frames, and the DPWM periodic message has 7 frames. The specific frame contents are listed below.

Table 1 shows the information of the 8-frame period packet sent by the host computer in the system, including the packet name, ID, packet length, and period. The BCM_DDAPWL_01 and BCM_DDAPWL_02 are periodic monitoring request frame headers. Then, the monitoring frame header is sent, BCM_01 is the Global message of the ECU, and is used for the ECU remote controller control, and BCM_Check_02 to BCM_Check_06 are the periodic type verification request frame headers, and if the verification is performed, the verification frame header is transmitted.

Table 1 Host machine cycle type message

| Message name       | ID  | Message length | cycle /ms |
|--------------------|-----|----------------|-----------|
| BCM_DDAPWL_01      | 0x27| 8              | 20        |
| BCM_DDAPWL_02      | 0x28| 8              | 20        |
| BCM_01             | 0x31| 8              | 20        |
| BCM_Check_02       | 0x22| 8              | 20        |
| BCM_Check_03       | 0x23| 8              | 20        |
| BCM_Check_04       | 0x24| 8              | 20        |
| BCM_Check_05       | 0x25| 8              | 20        |
| BCM_Check_06       | 0x26| 8              | 20        |

Table 2 shows the information of the 5-frame event-type packet sent by the host computer in the system, including the packet name, ID, packet length, and period. The BCM_02 to BCM_06 are event-type request packets.
Table 2 Host computer event type message

| Message name | ID   | Message length | cycle /ms |
|--------------|------|----------------|-----------|
| BCM_02       | 0x32 | 8              |           |
| BCM_03       | 0x33 | 8              |           |
| BCM_04       | 0x34 | 8              |           |
| BCM_05       | 0x35 | 8              |           |
| BCM_06       | 0x36 | 8              |           |

Table 3 shows the information of the 7-frame periodic packet sent by the lower-level device in the system, including the packet name, ID, packet length, and period. DDAPWL_01 and DDAPWL_02 are monitoring response packets. After the frame header, the monitoring response message is sent, and CLI_01 to CLI_05 are the verification reply message. If the host computer checks the frame header, it sends a verification reply message.

Table 3 lower machine cycle type message

| Message name | ID   | Message length | cycle /ms |
|--------------|------|----------------|-----------|
| CLI_01       | 0x18 | 8              | 20        |
| CLI_02       | 0x19 | 8              | 20        |
| CLI_03       | 0x1A | 8              | 20        |
| CLI_04       | 0x1B | 8              | 20        |
| CLI_05       | 0x1C | 8              | 20        |
| DDAPWL_01    | 0x10 | 8              | 20        |
| DDAPWL_02    | 0x11 | 8              | 20        |

Figure 4 shows the actual hybrid scheduling table of the system. Each row contains 5 basic time slices. Each row consists of 3 frame periodic messages and 2 basic time slices. The basic time slice is interspersed with periodic message settings. The basic time slice $T_s$ The value is 20 ms, the hybrid scheduling period $T_x$ of the hybrid schedule is 500 ms, and the basic packet transmission basic period $T_a$ takes 100 ms.

Figure 5 is the effect of the delay time of the event type message under the load rate of different periodic types of messages. The figure shows the delay time curve of the event type message during the change of the periodic message load rate from 0% to 95%. The graph shows the changes in the delay profile of event-type messages in three different cases.
Figure 5 Effect of delay time of event-type packets under different periodic packet load rates

Table 4 shows the average theoretical delay time of periodic packets sent by equations (8), (9), and (10) at different load rates. The average delay time of event-type messages in the above waveform is analyzed. The actual average delay time of the event-type message in the static schedule and the hybrid schedule can be calculated. The calculation can conclude that the hybrid scheduling algorithm in this paper will be compared with the traditional static scheduling algorithm in the event-type packet transmission delay. The delay has increased by about 30%.

Table 4 Average delay time of event-type packets under different periodic packet load rates.

| Delay time /ms | Periodic message load rate |
|----------------|----------------------------|
|                | 25% | 50% | 75% | 95% |
| Theoretical delay | 2.5 | 4.6 | 7.9 | 15.1 |
| Static scheduling | 3.65 | 6.05 | 9.95 | 18.15 |
| Hybrid scheduling | 3.31 | 5.51 | 9.21 | 17.01 |

5. Conclusion
This paper presents a hybrid scheduling algorithm based on blank time slice and periodic time slice. Through the establishment of hardware in the loop simulation test system, two different scheduling algorithms are compared and tested to verify the timeliness of the hybrid scheduling algorithm. Through the analysis of experimental data, it is concluded that the hybrid scheduling algorithm is 30% more efficient than the static scheduling algorithm in terms of the transmission delay of event-based messages, and in the LIN bus message transmission process, with the increase of the bus load rate The timeliness of data transmission is better.

Acknowledgement
We want to thank Professor Cheng Anyu for his support and tremendous help in this study. This research project was supported by the Chongqing Municipal Education Commission Science and Technology Project (KJ1722377) and the Chongqing Municipal Education Commission's joint construction project.

REFERENCES
[1] Yinfeng. Research on Static Segment Scheduling Algorithm of Vehicle LIN Bus, Master thesis of Hunan University, 2016
[2] Jianli Yin, Research on Dynamic Segment Scheduling and Optimization Configuration of Automotive LIN Network, Master thesis of Hefei University of Technology, 2014
[3] Homchaudhuri B., Hierarchical control strategies for energy management of connected hybrid electric vehicles in urban roads[J]. Transportation Research Part C Emerging Technologies, 2016,62:70-86
[4] Bu Qiao, Distributed Networked Time Lag Analysis and Control for Distributed Drive Electric Vehicles, Beijing Institute of Technology Master thesis, 2016

[5] Ma C, Ko S Y, Jeong K Y, et al. Design methodology of component design environment for PHEV[J]. International Journal of Automotive Technology, 2013, 14(5): 785-795

[6] Lee D H, Kim N W, Jeong J R, et al. Component sizing and engine optimal operation line analysis for a plug-in hybrid electric transit bus [J]. International Journal of Automotive Technology, 2013, 14(3): 459-469

[7] Zeng X, Wang J. A Parallel Hybrid Electric Vehicle Energy Management Strategy Using Stochastic Model Predictive Control With Road Grade Preview [J]. IEEE Transactions on Control Systems Technology, 2015, 23(6): 2416-2423