CANopen Message Real-Time Optimization Based on Hybrid Scheduling Method

FU Li^{1, a}, TONG Guoxiang^{1, 2, b}

¹ School of Optical-Electrical and Computer Engineering, University of Shanghai for

Science and Technology, Shanghai 200093, China

² Shanghai Key Laboratory of Modern Optical Systems, Shanghai 200093, China ^aemail: alan2010@126.com, ^bemail: guoxiangtong@163.com

Keywords: CAN; CANopen; PDO; Hybrid Scheduling Method; Dynamic Priority Scheduling; EDF

Abstract. According to CANopen Process Data Objects (PDO) transmitting modes and message scheduling conditions, using the static priority distribution strategy and dynamic priority scheduling algorithm, this article introduces a hybrid scheduling method for CANopen message. Theories and experiments show that the hybrid scheduling method could decrease message collision probability and improve message worst-case response time, so it could optimize the CANopen message real-time feature and increase the CAN network utilizing efficiency.

Introduction

Along with the development of industry and network technology, fieldbus technology has been widely applied in the field of industrial control and automation. As one kind of fieldbus, CAN (Controller Area Network) bus is one of the most widely used and researched fieldbus.

The CAN fieldbus only defines the physical layer and data link layer. Application layer functions are not clearly defined. CANopen protocol is an application layer protocol of CAN bus. Its main advantages are simplicity, which permits cheaper implementations, and the availability of many device profiles that define the standardized behavior of several kinds of equipment^[1].

In order to reduce CAN message delay time, there are some strategies for message scheduling. At present the main scheduling mechanism can be divided into two kinds, static and dynamic scheduling. For CANopen, the basic protocol of physical layer and data link layer is still the same with CAN, so it still conforms to the real-time characteristics of CAN messages. To optimize CANopen real-time feature, we consider taking reference of CAN message scheduling strategies.

The scheduling condition of CANopen message

The CANopen message worst-case delay time R_i is the sum of queuing time and transferring time, expressed as follows:

$$R_i = t_i + C_i \tag{1}$$

Where t_i is the queueing time and C_i is the time of transmitting message to the bus.

$$\mathbf{t}_i^{n+1} = B_i + \sum_{\forall j \in hp(i)} \left[\frac{t_i^n + J_j + \tau_{bit}}{T_j} \right] * C_j \tag{2}$$

In this equation, B_i is the longest time of low priority message occupied the bus, hp(i) is the set of all messages whose priorities are higher than message *i*, J_j is the jitter of message *j*. While τ_{bit} is the bus bit time and T_j is the transmitting period of message *j*.

Regard to the equation (2), it is possible to solve this equation using an iterative technique ^[2]. The start value t_i^0 is smaller than t_i^n and we could set $t_i^0 = 0$ here. Iteratively calculate until

 $t_i^{n+1} = t_i^n$ we could get a solution of the equation. Combined with equation (1) and (2) we obtain:

$$\mathbf{R}_{i}^{n+1} = C_{i} + B_{i} + \sum_{\forall j \in hp(i)} \left[\frac{R_{i}^{n} + J_{j} + \tau_{bit}}{T_{j}} \right] * C_{j}$$
(3)

Similarly, by iteration, we could get solution of R_i . It is shown that $R_i^{n+1} \ge R_i^n$ and iteration will be halted if either $R_i^{n+1} > T_i$ or if $R_i^{n+1} > D_i$ (where D_i is the deadline of message *i*).

Set S_i to be the interval of adjacent frames, the worst case of C_i could be expressed as:

$$C_{i} = \left[8 * S_{i} + 34 + 13 + \left(\frac{8 * S_{i} + 34 - 1}{4}\right)\right] * \tau_{bit}$$
(4)

The timing feature constrains of CANopen message could be expressed as:

$$R_i < D_i \le T_i \tag{5}$$

$$U = \sum_{i=1}^{N} \frac{C_i}{T_i} \le 1 \tag{6}$$

Where U is the CAN bus utilization rate and other symbols have been mentioned in the above. To ensure the message transmitting effectively, we need to schedule the messages reasonably.

The priority distribution of CANopen TPDO message

CANopen PDOs could be divided into three modes: event triggered mode, synchronous mode and requested mode. Event triggered PDOs could be used to transfer hard real-time messages, synchronous PDOs could be used to transfer soft real-time messages and requested PDOs used to transfer no real-time message.

Using CANopen synchronization object, we divide the synchronous cycle into two parts: the synchronization window and free time window, as shown in figure 1. Event triggered PDOs could be transmitted at any time; synchronous PDOs may be transmitted at the synchronization window every one or several cycles; requested PDOs will be transmitted at the free time window only.

Through this distribution method, it can reduce the probability of message collision.



Figure 1 CANopen Synchronization and TPDO distribution

The hybrid scheduling strategy involved with dynamic scheduling

On CAN bus a given message is assigned a fixed identifier (and hence a fixed priority) ^[3]. The smaller of the identifier, the higher of message priority would be. In this way, the highest priority message would always win the bus in arbitration. If the bus load increased and collision reached to

a certain degree, the lower priority message may be unable to obtain the bus because of arbitration lost, and finally causes message lost and communication error which would lead to serious influence to the system. In order to solve this problem we could adopt a hybrid priority scheduling method.

We use 29-bits extended identifier for the CANopen message frame, the most significant 4 bits as function code and least significant 7 bits as the node ID. The middle 18 bits are reserved for hybrid scheduling method among which we use the higher 10 bits for dynamic scheduling and lower 8 bits for static priority distribution, as shown if figure 2.

Arbitration Field (29 bits identifier)						
Bit25~bit28	bit15~bit24	bit7~bit14	bit0~bit6			
CANopen Function Code	Dynamic Scheduling	Static Priority Setting	CANopen Node ID			
(4 bits)	(10 bits)	(8 bits)	(7 bits)			

Figure 2 The 29 bits identifier distribution for hybrid scheduling method One dynamic scheduling algorithm is the Earliest Deadline First (EDF) scheduling which works by giving higher priority to the message having the earliest absolute deadline. EDF is the basis of many real-time processor scheduling algorithms because of its high utilization^[4].

The dynamic scheduling algorithm could be expressed as:

$$P_{i}(t) = P_{i}(0) - \frac{k}{D_{i} - t}$$
(7)

$$0 \le P_i(0) \le 2^{10} \tag{8}$$

$$D_i - t > 0 \tag{9}$$

In equation (7), $P_i(t)$ means the dynamic priority of message *i* at time *t* after several arbitration lost. $P_i(0)$ is the initial dynamic priority of message *i* which we would generally set to be 2^{10} as there are 10 bits we defined for dynamic scheduling. The smaller of P_i the higher priority would be. $D_i - t$ means the deadline of message *i* at time *t*, and *k* is a weighted value we will define according to different situations. If $P_i(t)$ reached to 0 that would mean message *i* has got its dynamic highest priority.

Experiments and test results

To check the efficiency of this hybrid scheduling method, we use one master and two slave nodes to construct a CAN network. Calculate and set the bus load to be about 80%, then we do the contrast experiments by using the hybrid scheduling method and without using it.

Set the CAN bus baud rate to be 125kbps, the master sends a synchronization object every 1ms while two slave nodes transmit event triggered PDOs every 1ms and synchronous PDOs every 2ms. The priorities of messages on slave nodes refresh every 200us using hybrid scheduling method. The micro-processor calculates the average response time and records the worst-case response time. The test results are shown in table 1. This test is in a relatively heavy load condition and we can see that using the hybrid scheduling method can improve response time especially for the worst-case response time.

		Node 1		Node 2	
		the average	the worst-case	the average	the worst-case
		response time	response time	response time	response time
Event triggered TPDO	without using scheduling method	740us	1650us	710us	1730us
	used hybrid scheduling method	530us	810us	570us	830us
Synchronous TPDO	without using scheduling method	850us	1870us	840us	1850us
	used hybrid scheduling method	680us	910us	730us	890us

 Table 1
 the contrast test of using hybrid scheduling method and not using it

Conclusion

Based on the analysis of CAN message scheduling conditions, this article introduced a hybrid scheduling method to optimize the CANopen message. Through the reasonable distribution of CANopen TPDOs and dynamic priority scheduling, it could avoid the crowd of message collisions and greatly improve the CANopen message response time, so to achieve a better real-time communication for the industrial applications. The analysis and methods presented here can be referenced and extended to the investigation of other similar systems.

Acknowledgement

In this paper, the research was sponsored by the National High Technology Research and Development Program of China (Grant No. 2006AA03Z348), the Key Project of Chinese Ministry of Education (Grant No. 207033).

References

[1] G. Cena, A. Valenzano. Efficient polling of devices in CANopen networks, Proceedings of the ETFA03 IEEE Conference, 2003, 1: 123-130.

[2] N. Audsley, A. Burns, M Richardson, K. Tindell and A. J. Wellings, Applying new scheduling theory to static priority pre-emptive scheduling [J]. Real-Time Systems, 1995, (1)

[3] Tindell K W, Burns A. Guaranteeing message latencies on controller area network (CAN) [EB / OL]. 2007-09-03.

[4] Natale M D. Scheduling the CAN bus with earliest deadline techniques [C]. 21st IEEE Real-Time System, 2000.

[5] G. Leen, D. Heffernan. Time-triggered controller area network. Computing & Control Engineering Journal, 2001, 12(6): 245-256.

[6] Tindell K W, Hansson H, Wellings A J. Analysing real-time communications: controller area network (CAN)[EB/OL]. 2007-09-03.