CANopen Message Real-Time Optimization Based on Hybrid Scheduling Method

FU Li¹,², TONG Guoxiang¹,²,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:

$$R_i = t_i + C_i$$  \hspace{1cm} (1)

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

$$t^{n+1}_i = B_i + \sum_{j \in hp(i)} \left( \frac{t^n_i + J_j + \tau_{bit}}{T_j} \right) \cdot C_j$$  \hspace{1cm} (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 $\tau_{bit}$ is the bus bit time and $T_j$ is the transmitting period of message $j$.

Regarding the equation (2), it is possible to solve this equation using an iterative technique [2]. The start value $t^0_i$ is smaller than $t^n_i$ and we could set $t^0_i = 0$ 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:

\[ R_{i}^{n+1} = C_{i} + B_{i} + \sum_{j \in \text{set}(i)} \left[ \frac{R_{i}^{n} + J_{j} + \tau_{\text{bit}}}{T_{j}} \right] * C_{j} \] (3)

Similarly, by iteration, we could get solution of \( R_{i} \). It is shown that \( R_{i}^{n+1} \geq R_{i}^{n} \) 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:

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

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

\[ R_{i} < D_{i} \leq T_{i} \] (5)

\[ U = \sum_{i=1}^{n} \frac{C_{i}}{T_{i}} \leq 1 \] (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](image)

**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](image)

<table>
<thead>
<tr>
<th>Bit25~bit28 CANopen Function Code (4 bits)</th>
<th>bit15~bit24 Dynamic Scheduling (10 bits)</th>
<th>bit7~bit14 Static Priority Setting (8 bits)</th>
<th>bit0~bit6 CANopen Node ID (7 bits)</th>
</tr>
</thead>
</table>

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)

\[ 0 \leq P_i(0) \leq 2^{10} \]  
(8)

\[ D_i - t > 0 \]  
(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.
<table>
<thead>
<tr>
<th>Event triggered TPDO</th>
<th>Node 1</th>
<th>Node 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>without using scheduling method</td>
<td>740us</td>
<td>1650us</td>
</tr>
<tr>
<td>used hybrid scheduling method</td>
<td>530us</td>
<td>810us</td>
</tr>
<tr>
<td>Synchronous TPDO</td>
<td>850us</td>
<td>1870us</td>
</tr>
<tr>
<td>without using scheduling method</td>
<td>840us</td>
<td>1850us</td>
</tr>
<tr>
<td>used hybrid scheduling method</td>
<td>680us</td>
<td>910us</td>
</tr>
<tr>
<td>the average response time</td>
<td>710us</td>
<td>830us</td>
</tr>
<tr>
<td>the worst-case response time</td>
<td>1730us</td>
<td>890us</td>
</tr>
</tbody>
</table>

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**


