Electronic Preprint for Journal of Information Processing Vol.21 No.3

Regular Paper

Slot Multiplexing Optimization for Minimizing the Operating Frequency of a FlexRay Bus under Hard Real-time Constraints

Makoto Sugihara[1,2,a)] Akihito Iwanaga[3,b)]

Received: November 5, 2012, Accepted: April 5, 2013

Abstract: Industrial applications such as automotive ones require a cheap communication mechanism to send out communication messages from node to node by their deadline time. This paper presents a design paradigm in which we optimize slot multiplexing of a FlexRay bus under hard real-time constraints so that we can minimize the operating frequency of the FlexRay bus. The reduction of the operating frequency of a FlexRay bus helps one to choose a slower and cheaper wire harness for building a distributed system. We formulate a slot multiplexing optimization problem under hard real-time constraints. We build an integer linear programming (ILP) model as a means for solving the slot multiplexing optimization problem. Our experimental results show that our design paradigm achieved a 72.7% smaller operating frequency at its best than the naive one.

Keywords: FlexRay, slot multiplexing, hard real-time, bandwidth, cost, time division multiple access, static segment, scheduling.

1. Introduction

The automobile industry has been incessantly meliorating their automotive products in order to achieve low energy consumption, high safety and high comfort. The evolvement of automobiles is attributed largely to that of electronics. In-vehicle electronics will play a more and more important role in developing next-generation automobiles. From the viewpoint of economy, a design paradigm is essential to build an automotive electronic system at a cheap cost as the system consists of expensive electronic devices and wires. A luxury car contains over 100 electronic control units (ECUs) which communicate with one another via a communication network. A car now contains a distributed embedded system.

The controller area network (CAN) is a communication network standard and is employed as an in-vehicle communication network [2]. The CAN suffers from low determinacy with communication latency. The CAN is incapable of implementing automotive applications such as x-by-wire systems [3]. The FlexRay network is another communication network standard which provides determinacy with communication latency as well as flexibility with network bandwidth [4]. The FlexRay communication mechanism is based on a time division multiple access (TDMA) scheme. The FlexRay communication mechanism offers determinacy with communication latency by the static time division multiple access (STDMA) scheme while it does flexibility with bandwidth consumption by the flexible time division multiple access (FTDMA) scheme.

It is a challenging theme to study a design paradigm that builds a FlexRay communication mechanism. This paper proposes a design paradigm in which we optimize slot multiplexing of a FlexRay bus so that we minimize the bandwidth of the bus. The reduction of network bandwidth generally contributes to cutting down the cost of the wire harness. To the best of our knowledge our work is the first design paradigm to minimize the operating frequency of a communication bus by optimally determining slot multiplexing, that is, assigning each and every communication signal to one or more static slots of 1–64 communication cycles so that the cost of the mechanism is reduced.

The remainder of this paper is organized as follows: Section 2 briefly reviews a FlexRay communication system. Section 3 discusses and formulates an operating frequency minimization problem in which we assume that the TDMA parameters and the operating frequency of a bus are variables under hard deadline constraints. Section 4 shows our experiments on operating frequency minimization. Section 5 compares our work with other related work. Section 6 provides concluding remarks.

2. FlexRay Bus

In this section we briefly review the FlexRay bus standard [4]. Further detail should be referred to Ref. [4].

2.1 Communication Cycle

A *communication cycle* is one complete instance of the communication structure that is periodically repeated to comprise the media access method of the FlexRay system. A communication

1 Kyushu University, Fukuoka 819–0395, Japan
2 JST CREST, Chiyoda, Tokyo 102–0075, Japan
3 Bosch Corporation, Shibuya, Tokyo 150–0002, Japan
a) sugihara@slrc.kyushu-u.ac.jp
b) iwanaga@soc.ait.kyushu-u.ac.jp

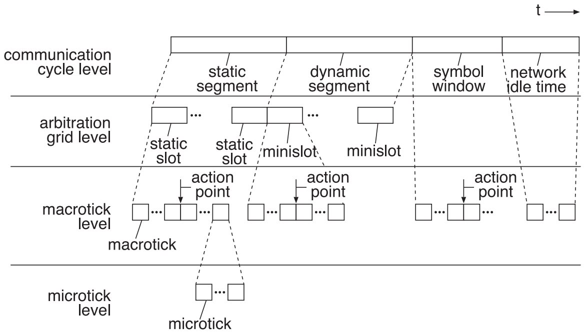

Fig. 1 Timing hierarchy within the communication cycle.

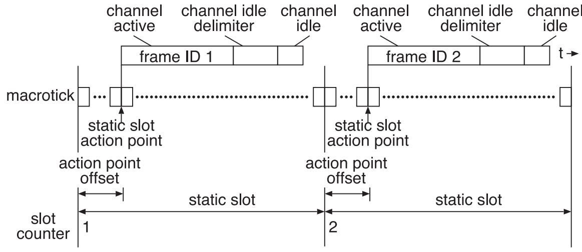

Fig. 2 Timing within a static segment.

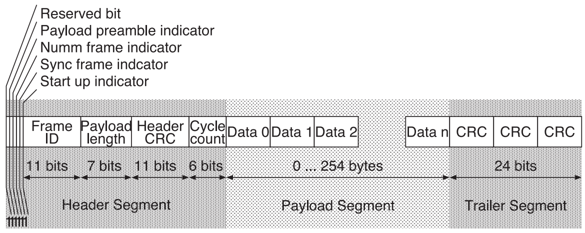

Fig. 3 Frame format.

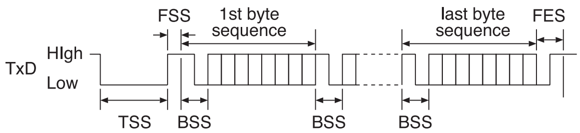

Fig. 4 Frame encoding.

cycle is defined by means of a timing hierarchy. The timing hierarchy consists of four timing hierarchy levels: microtick level, macrotick level, arbitration grid level, communication cycle level, as depicted in **Fig. 1**.

At the communication cycle level, communication cycles are repeatedly executed. A communication cycle contains a *static segment* (*SS*), a *dynamic segment* (*DS*), a *symbol window* (*SW*), and a *network idle time* (*NIT*). This paper focuses only on the static segment. The time division multiple access (TDMA) scheme is generally a channel access method for shared medium networks. It temporally divides a channel into time slots each of which only a permitted communication message(s) can use. Within a static segment a static time division multiple access (STDMA) scheme is used to arbitrate transmissions. Within a dynamic segment the flexible time division multiple access (FTDMA) scheme is used to arbitrate transmissions. A symbol window is a communication period in which a symbol can be transmitted on the network. A network idle time is a communication-free period that concludes each communication cycle.

The next lower level, the arbitration grid level, contains the arbitration grid. In a static segment the arbitration grid consists of consecutive time intervals, called *static slots*. In the dynamic segment the arbitration grid consists of consecutive time intervals, called *minislots*.

The arbitration grid level builds on the macrotick level that is defined by the macrotick. A macrotick represents the smallest granularity unit of the global time. Designated macrotick boundaries in Fig. 1 are called action points. An action point is an instant in time at which a transmitter starts a transmission of a FlexRay frame.

The lowest level in the hierarchy is defined by microticks. A microtick is an interval of time and is a node-local concept.

2.2 Media Access Control

In the FlexRay protocol, media access control is based on a recurring communication cycle. Within one communication cycle FlexRay offers the choice of two media access schemes: a static TDMA scheme and a dynamic mini-slotting based scheme. The former scheme is used in static segments while the latter one in dynamic segments. A communication frame which is a transmission unit like a packet is sent within the static and dynamic segments. We briefly review the TDMA scheme for static segments.

Each and every frame has its frame identifier. A frame identifier is uniquely assigned to a static slot. The frame identifier of a frame determines which transmission slot it uses. Arbitration is based on the unique assignment of a frame identifier to a node. Whenever a new static slot comes, a slot counter increments its value by one. A frame is sent when its frame identifier corresponds with a slot counter. The initial value of a slot counter is one. At the end of a static segment the slot counter is reinitialized with one.

All static slots consist of an identical number of macroticks. The number of macroticks per static slot is a global constant. Detailed timing of static slots is given in **Fig. 2**.

2.3 Frame Format

A FlexRay frame is a container for transmission which consists of three segments: header segment, payload segment and trailer segment. The FlexRay frame format is shown in **Fig. 3**.

The node shall transmit the frame on the network such that the header segment appears first, followed by the payload segment, and then followed by the trailer segment, which is transmitted last. Within the individual segments the node shall transmit the fields in left to right order as depicted in Fig. 3.

2.4 Frame Encoding

A FlexRay frame is encoded with five sequences: a transmission start sequence (TSS), a frame start sequence (FSS), a byte start sequence (BSS), a frame end sequence (FES), and a dynamic trailing sequence (DTS). A TSS and an FSS are inserted at the beginning of a frame. Every byte of a frame has its BSS before it. An FES is added after a frame. A DTS is added in a dynamic segment immediately after an FES of a frame.

Figure 4 shows decoded frames in a static segment. A TSS is used to initiate proper connection setup through the network. A

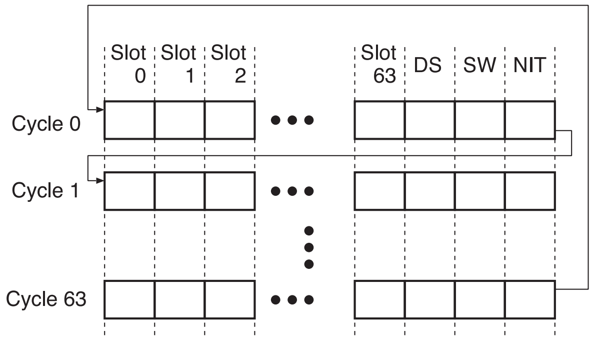

Fig. 5 Slot multiplexing.

transmission node generates a TSS that consists of a continuous LOW for a period given by a FlexRay parameter. An FSS follows the TSS. An FSS is used to compensate for a possible quantization error in the first byte start sequence after the TSS. An FSS consists of one HIGH bit time. The node shall append an FSS to the bit stream immediately following the TSS of a transmitted frame. A BSS is used to provide bit stream timing information to the receiving devices. The BSS shall consist of one HIGH bit time followed by one LOW bit time. Each byte of frame data shall be sent on the channel as an extended byte sequence that consists of one BSS followed by eight data bits. An FES is used to mark the end of the last byte sequence of a frame. The FES shall consist of one LOW bit time followed by one HIGH bit time. The node shall append an FES to the bit stream immediately after the last extended byte sequence of the frame.

2.5 Slot Multiplexing

The FlexRay standard specifies *cycle-independent slot assignment* as the method of assigning, for a given channel, the set of all communication slots having a specific slot number to a node (i.e., on the given channel, slots with the specific slot number are assigned to the node in all communication cycles). The FlexRay standard specifies *cycle-dependent slot assignment* as the method of assigning, for a given channel, an individual slot (identified by a specific slot number and a specific cycle counter number) or a set of slots (identified by a specific slot number and a set of communication cycle numbers) to a node. The FlexRay standard specifies *slot multiplexing* (*SM*) as the technique of assigning, for a given channel, slots having the same slot identifier to different nodes in different communication cycles. **Figure 5** shows an example of slot multiplexing. The FlexRay standard specifies that one may use up to 64 (= N_{cycles}) communication cycles for slot multiplexing. The communication cycles are recurringly executed with an identical period that depends on the number of communication cycles. This paper assumes that one uses 64 communication cycles for slot multiplexing.

3. Operating Frequency Minimization Problem

The reduction of the operating frequency of a FlexRay communication mechanism contributes to cost reduction as low operating frequency enables one to choose a cheap wire harness for building a communication system. This paper presents a design paradigm in which we synthesize an optimal FlexRay bus whose operating frequency is the minimum under hard real-time constraints. In this section we define a slot multiplexing optimization problem in which we optimize slot multiplexing of a FlexRay bus so that we can minimize the operating frequency of the bus under constraints of hard real-time deadlines. We also show an ILP problem model for the operating frequency minimization problem as a means to solution.

3.1 Problem Formulation

We define a *communication signal* as the behavior of a node in requesting the sending of a constant size of data by its deadline periodically or aperiodically. We also define a *communication message* as an instance of a communication signal which a node requests to send via the FlexRay bus. From an aspect of implementation, a communication message is split into one or more frames which are data structures for transmission via a bus. For simple problem formulation we assume a network system in which all communication messages are requested to be sent periodically. Without loss of generality an aperiodic network system can be transformed to a periodic one because an aperiodic communication signal can be treated as a periodic one by regarding the minimal interval between successive communication signal instances as its period. A communication signal is defined by a 4-tuple (N, C, D, B) where N is a sender node, C is a period between successive communication messages, D is a relative deadline from its request, and B is the size of a communication message in bits. We assume that Node N requests to send a datum of B bits every C time units and finish sending it within duration D from its request.

We now discuss a time-triggered network system in which a set of N_{sig} communication signals, $\mathbb{S} = \{S_1, S_2, \cdots, S_{N_{\mathrm{sig}}}\}$, are transmitted via static segments each of which consists of q ($\leq N_{\mathrm{sig}}$) static slots. We treat the number of static slots, q, as a variable. Communication signal S_i is defined by a 4-tuple (N_i, C_i, D_i, B_i). We assume that a communication signal may use no more than a static slot within a communication cycle. We assume that the time-triggered network works at w Hz and its bandwidth is w b/s. The objective function in this paper is to minimize w. The minimization of the operating frequency of a bus contributes to reducing the cost of building network hardware because a lower operating frequency offers flexibility in choosing a wire harness and results in choosing a cheap wire harness.

A communication message is sent in the form of frames. A frame consists of header, payload, and trailer segments as shown in Fig. 3. We use notations B_{hd} and B_{tl} for the header size and the trailer size in bits respectively. The size of a payload segment shall be fixed and identical among all frames in a static segment. We use notation P for the payload size in the number of 16-bit words. The standard for FlexRay states that the size of a payload shall be an even number in bytes between 0 to 254. The size of a frame in bits, F, is shown as follows:

$$F = B_{\mathrm{hd}} + B_{\mathrm{tl}} + 16P. \quad (1)$$

The size of an encoded frame in bits, F_{enc}, is formulated as follows:

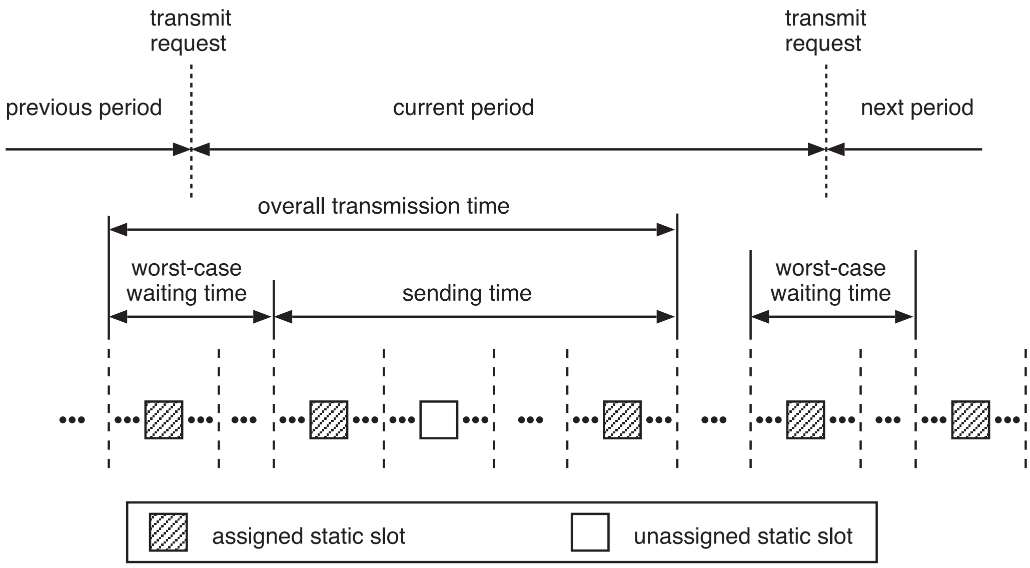

Fig. 6 Total transmission time for a communication message.

$$\begin{aligned}F_{\text{enc}} &= F + \left(B_{\text{TSS}} + B_{\text{FSS}} + \frac{F}{8} \cdot B_{\text{BSS}} + B_{\text{FES}} + B_{\text{DLM}} + B_{\text{IDL}}\right) \\ &= \left(1 + \frac{B_{\text{BSS}}}{8}\right) F + B_{\text{TSS}} + B_{\text{FSS}} + B_{\text{FES}} + B_{\text{DLM}} + B_{\text{IDL}} \\ &= (16 + 2B_{\text{BSS}})\, P + O \end{aligned} \quad (2)$$

$$O = \left(1 + \frac{B_{\text{BSS}}}{8}\right)(B_{\text{hd}} + B_{\text{tl}}) + B_{\text{TSS}} + B_{\text{FSS}} + B_{\text{FES}} + B_{\text{DLM}} + B_{\text{IDL}},$$

where B_{TSS}, B_{FSS}, B_{BSS}, B_{FES}, B_{DLM}, and B_{IDL} are the numbers of bit time for a TSS, an FSS, a BSS, an FES, a channel idle delimiter, and a channel idle respectively.

For a simple explanation, we assume that only hard real-time communication signals are given and that a communication cycle consists of a static segment in order to satisfy their deadline constraints. The problem that will be defined in this section is easily applied to a network system that has a static segment, a dynamic segment, a symbol window, and network idle time as far as the length for a dynamic segment, a symbol window, and network idle is treated as a constant. Message scheduling for dynamic segments must, however, be done separately from that for static segments. Our approach is dedicated to scheduling communication messages only over static segments. Message scheduling for dynamic segments should be discussed separately from our approach. We send a communication message in the form of frames. **Figure 6** shows how a communication message is sent with frames. From the assumption, a communication cycle t_{cc} is equal to the time spent for all static slots and is formulated as follows:

$$t_{\text{cc}} = \frac{q \cdot F_{\text{enc}}}{w}, \quad (3)$$

where q is an integer variable standing for the number of static slots constituting a communication cycle and w is a real variable for the operating frequency of the FlexRay bus.

The FlexRay standard states that one may freely choose static slots of 64 communication cycles on designing a system for sending a communication message. The configuration of static slots of a communication cycle is distinguished from that of another communication cycle. One may build 64 configurations on static slots. The 64 communication cycles appear recurringly. The freedom of choice generally makes design space huge. For simple problem formulation we assume that a communication signal consumes 2^j $(0 \leq j \leq 6)$ static slots out of 64 communication cycles and that it uses a static slot every 2^{6-j} communication cycles. We introduce the following binary variable $a_{i,j}$ to indicate how many communication cycles are available out of 64 ones.

$$a_{i,j} = \begin{cases} 1 & \text{if communication signal } i \text{ consumes } 2^j \text{ communication cycles out of 64 ones and uses one every } 2^{6-j} \text{ communication cycles,} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

The following constraints are introduced for the above variables.

$$\sum_{j=0}^{6} a_{i,j} = 1, \forall i. \quad (5)$$

It takes $\left\lceil \frac{B_i}{16P} \right\rceil$ static slots for communication signal i to send a communication message. The transmission time to send a communication message is required to be no more than the period of a communication message. The following constraint is, therefore, introduced.

$$\left\lceil \frac{B_i}{16P} \right\rceil \left(\sum_{j=0}^{6} 2^{6-j} a_{i,j} \right) t_{\text{cc}} \leq C_i, \forall i. \quad (6)$$

The above constraint is transformed using Eq. (3) as follows.

$$F_{\text{enc}} \left\lceil \frac{B_i}{16P} \right\rceil \left(\sum_{j=0}^{6} 2^{6-j} \frac{a_{i,j} q}{w} \right) \leq C_i, \forall i. \quad (7)$$

There are 2^{6-j} ways for a communication signal to periodically use 2^j communication cycles out of 64 ones because there are 2^{6-j} communication cycles for the first communication cycle of the communication signal. We introduce the following variable $c_{i,l}$ $(1 \leq l \leq 64)$ to indicate the phase of communication cycles.

$$c_{i,l} = \begin{cases} 1 & \text{if the communication cycle which communication signal } i \text{ first uses is the } l\text{-th communication cycle,} \\ 0 & \text{otherwise.} \end{cases}$$

The following constraint is introduced for the above variable.

$$a_{i,j} = 1 \Rightarrow \sum_{l=1}^{2^{6-j}} c_{i,l} = 1 \text{ and } c_{i,l} = 0\ (\forall l > 2^{6-j}), \forall i, \forall j. \quad (8)$$

Once it is determined which communication cycle each and every communication signal first uses, it is fixed which communication cycles a communication message utilizes. We introduce the following binary variable $d_{i,m}$ to indicate which communication cycles a communication signal uses.

$$d_{i,m} = \begin{cases} 1 & \text{if a communication message of communication signal } i \text{ uses } m\text{-th communication cycle,} \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

From the assumption that a communication signal uses a communication cycle every 2^{6-j} cycles, the following constraint is introduced.

$$a_{i,j} = 1 \text{ and } c_{i,l} = 1 \Rightarrow d_{i,m} = 1 \text{ and } m = l + h \cdot 2^{6-j}, \forall i, \forall j, \forall l \leq 2^{6-j}, 0 \leq \forall h \leq 2^j - 1. \quad (10)$$

The number of static slots in a communication cycle, q, is determined by the maximum number of communication signals assigned to a communication cycle among communication cycles as follows.

$$q = \max_m \sum_i d_{i,m}. \quad (11)$$

Waiting time occurs before the first frame is sent by the corresponding static slot. Let the worst-case waiting time be equal to the interval between assigned slots:

$$\left(\sum_{j=0}^{6} 2^{6-j} a_{i,j}\right) t_{\mathrm{cc}} \quad (12)$$

on the assumption that a communication signal is allocated to any static slot in a communication cycle. Time for sending frames follows the waiting time. Let us conservatively assume that the time for sending all frames is equal to

$$\left(\left\lceil \frac{B_i}{16P} \right\rceil - 1\right)\left(\sum_{j=0}^{6} 2^{6-j} a_{i,j}\right) t_{\mathrm{cc}}.$$

The worst-case latency to transmit a communication message of communication signal i, t_i, is the summation of the worst-case waiting time and the time for sending a communication message as shown in the following equation.

$$t_i = \left\lceil \frac{B_i}{16P} \right\rceil \left(\sum_{j=0}^{6} 2^{6-j} a_{i,j}\right) t_{\mathrm{cc}}. \quad (13)$$

The worst-case latency t_i must be no more than D_i in order to satisfy the given hard deadline constraint. From Eq. (13) the following constraint is introduced.

$$\left\lceil \frac{B_i}{16P} \right\rceil \left(\sum_{j=0}^{6} 2^{6-j} a_{i,j}\right) t_{\mathrm{cc}} \leq D_i, \forall i.$$

The above constraint is transformed using Eq. (3) as follows.

$$F_{\mathrm{enc}} \left\lceil \frac{B_i}{16P} \right\rceil \left(\sum_{j=0}^{6} 2^{6-j} \frac{a_{i,j} q}{w}\right) \leq D_i, \forall i. \quad (14)$$

The mathematical model is shown as follows.

Minimize the cost function w
subject to
(i) $\sum_{j=0}^{6} a_{i,j} = 1, \forall i.$
(ii) $F_{\mathrm{enc}} \left\lceil \frac{B_i}{16P} \right\rceil \left(\sum_{j=0}^{6} 2^{6-j} \frac{a_{i,j} q}{w}\right) \leq C_i, \forall i.$
(iii) $a_{i,j} = 1 \Rightarrow \sum_{l=1}^{2^{6-j}} c_{i,l} = 1$ and $c_{i,l} = 0\ (\forall l > 2^{6-j}), \forall i, \forall j.$
(iv) $a_{i,j} = 1$ and $c_{i,l} = 1 \Rightarrow d_{i,m} = 1$ and $m = l + h \cdot 2^{6-j}$, $\forall i, \forall j, \forall l \leq j, 0 < \forall h \leq 2^j - 1.$
(v) $q = \max_m \sum_i d_{i,m}.$
(vi) $F_{\mathrm{enc}} \left\lceil \frac{B_i}{16P} \right\rceil \left(\sum_{j=0}^{6} 2^{6-j} \frac{a_{i,j} q}{w}\right) \leq D_i, \forall i.$
Variables
- w is a real variable.
- q is an integer variable.
- $a_{i,j}$ is a binary variable.
- $c_{i,l}$ is a binary variable.
- $d_{i,m}$ is a binary variable.

Bounds
- $0 \leq q \leq N_{\mathrm{sig}}.$

3.2 Linearization

In this subsection we linearize the mathematical model shown in the previous subsection.

We assume that various but finite operating frequencies are available to use. The finite set of N_{bw} operating frequencies, $\mathbb{W} = \{W_1, W_2, \cdots, W_{N_{\mathrm{bw}}}\}$, is given. We introduce the following binary variable e_n to indicate whether or not the operating frequency W_n is adopted.

$$e_n = \begin{cases} 1 & \text{if } W_n \text{ is the operating frequency of the bus,} \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

The operating frequency of a time-triggered bus is now formulated as follows.

$$w = \sum_n W_n e_n. \quad (16)$$

A single operating frequency must be chosen for the time-triggered bus and the following constraint is introduced.

$$\sum_n e_n = 1. \quad (17)$$

From Eqs. (16) and (17), the inverse of variable w is formulated as follows.

$$\frac{1}{w} = \sum_n \frac{e_n}{W_n}. \quad (18)$$

From Eq. (18), constraints (7) and (14) are transformed as follows.

$$F_{\mathrm{enc}} \left\lceil \frac{B_i}{16P} \right\rceil \sum_{j,n} \frac{2^{6-j}}{W_n} a_{i,j} e_n q \leq C_i, \forall i. \quad (19)$$

$$F_{\mathrm{enc}} \left\lceil \frac{B_i}{16P} \right\rceil \sum_{j,n} \frac{2^{6-j}}{W_n} a_{i,j} e_n q \leq D_i, \forall i. \quad (20)$$

A nonlinear term $a_{i,j}e_n$ is linearized using the standard technique [1]. We introduce a new binary variable $g_{i,j,n}$ to linearize $a_{i,j}e_n$ as follows.

$$g_{i,j,n} = a_{i,j}e_n, \quad (21)$$
$$g_{i,j,n} - a_{i,j} \le 0,$$
$$g_{i,j,n} - e_n \le 0,$$
$$g_{i,j,n} - a_{i,j} - e_n \ge -1, \forall i, \forall j, \forall n.$$

A nonlinear term $a_{i,j}e_nq$ (= $g_{i,j,n}q$) is linearized using the standard technique [1]. We introduce a new integer variable $o_{i,j,n}$ to linearize $a_{i,j}e_nq$ as follows.

$$o_{i,j,n} = a_{i,j}e_nq = g_{i,j,n}q, \quad (22)$$
$$o_{i,j,n} \le N_{\text{sig}}g_{i,j,n},$$
$$o_{i,j,n} \le q,$$
$$o_{i,j,n} \ge q - N_{\text{sig}}(1 - g_{i,j,n}),$$
$$o_{i,j,n} \ge 0, \forall i, \forall j, \forall n$$

Constraints (19) and (20) are transformed using Eq. (22) as follows.

$$F_{\text{enc}}\left\lceil \frac{B_i}{16P} \right\rceil \sum_{j,n} \frac{2^{6-j}}{W_n} o_{i,j,n} \le C_i, \forall i. \quad (23)$$

$$F_{\text{enc}}\left\lceil \frac{B_i}{16P} \right\rceil \sum_{j,n} \frac{2^{6-j}}{W_n} o_{i,j,n} \le D_i, \forall i. \quad (24)$$

A nonlinear constraint (8) is linearized as follows.

$$\sum_{l=1}^{2^{6-j}} c_{i,l} - 1 - M_{i,l}(1 - a_{i,j}) \le 0, M_{i,l} = 2^j - 1, \forall i, \forall j.$$
$$\sum_{l=1}^{2^{6-j}} c_{i,l} - 1 - m_{i,l}(1 - a_{i,j}) \ge 0, m_{i,l} = -1, \forall i, \forall j.$$
$$c_{i,l} + a_{i,j} \le 1, \forall i, \forall j, \forall l > 2^j.$$

A nonlinear constraint (10) is linearized as follows.

$$z_{i,j,l} = a_{i,j}c_{i,l}, \quad (25)$$
$$z_{i,j,l} - a_{i,j} \le 0,$$
$$z_{i,j,l} - c_{i,l} \le 0,$$
$$z_{i,j,l} - a_{i,j} - c_{i,l} \ge -1, \forall i, \forall j, \forall l \le 2^{6-j}.$$
$$d_{i,m} - z_{i,j,l} \ge 0,\ m = l + h \cdot 2^{6-j},$$
$$\forall i, \forall j, 1 \le \forall l \le 2^{6-j}, 0 \le \forall h \le 2^j - 1.$$

A nonlinear constraint (11) is linearized as follows.

$$q \ge \sum_i d_{i,m}, \forall m.$$

The integer linear programming (ILP) model is given as follows.

Minimize the cost function $\sum_n W_ne_n$
subject to
(i) $\sum_{j=0}^{6} a_{i,j} = 1, \forall i.$
(ii) $\sum_n e_n = 1.$
(iii) $F_{\text{enc}}\left\lceil \frac{B_i}{16P} \right\rceil \sum_{j,n} \frac{2^{6-j}}{W_n} o_{i,j,n} \le C_i, \forall i.$
(iv) $F_{\text{enc}}\left\lceil \frac{B_i}{16P} \right\rceil \sum_{j,n} \frac{2^{6-j}}{W_n} o_{i,j,n} \le D_i, \forall i.$
(v) $g_{i,j,n} - a_{i,j} \le 0, \forall i, \forall j, \forall n.$
(vi) $g_{i,j,n} - e_n \le 0, \forall i, \forall j, \forall n.$
(vii) $g_{i,j,n} - a_{i,j} - e_n \ge -1, \forall i, \forall j, \forall n.$
(viii) $o_{i,j,n} \le N_{\text{sig}}g_{i,j,n}, \forall i, \forall j, \forall n.$
(ix) $o_{i,j,n} \le q, \forall i, \forall j, \forall n.$
(x) $o_{i,j,n} \ge q - N_{\text{sig}}(1 - g_{i,j,n}), \forall i, \forall j, \forall n.$
(xi) $o_{i,j,n} \ge 0, \forall i, \forall j, \forall n.$
(xii) $\sum_{l=1}^{2^{6-j}} c_{i,l} - 1 - M_{i,l}(1 - a_{i,j}) \le 0, M_{i,l} = 2^j - 1, \forall i, \forall j.$
(xiii) $\sum_{l=1}^{2^{6-j}} c_{i,l} - 1 - m_{i,l}(1 - a_{i,j}) \ge 0, m_{i,l} = -1, \forall i, \forall j.$
(xiv) $c_{i,l} + a_{i,j} \le 1, \forall i, \forall j, \forall l > 2^j.$
(xv) $z_{i,j,l} - a_{i,j} \le 0, \forall i, \forall j, \forall l \le j.$
(xvi) $z_{i,j,l} - c_{i,l} \le 0, \forall i, \forall j, \forall l \le j.$
(xvii) $z_{i,j,l} - a_{i,j} - c_{i,l} \ge -1, \forall i, \forall j, \forall l \le j.$
(xviii) $d_{i,m} - z_{i,j,l} \ge 0,\ m = l + h \cdot 2^{6-j}, \forall i, \forall j, 1 \le \forall l \le j, 0 \le \forall h \le 2^j - 1.$
(xix) $q \ge \sum_i d_{i,m}, \forall m.$

Variables
- q is an integer variable.
- $a_{i,j}$ is a binary variable.
- $c_{i,l}$ is a binary variable.
- $d_{i,m}$ is a binary variable.
- e_n is a binary variable.
- $g_{i,j,n}$ is a binary variable.
- $o_{i,j,n}$ is an integer variable.
- $z_{i,j,l}$ is a binary variable.

Bounds
- $0 \le q \le N_{\text{sig}}.$
- $0 \le x_{i,j,k,n} \le N_{\text{sig}}.$

In the operating frequency minimization problem, the deadline of a communication signal has a great impact on both the operating frequency of a bus and the number of static slots consumed by the corresponding communication signal.

4. Experiment

4.1 Experimental Setup

We utilized a network parameter set which had been shown by Park et al. [7] as shown in **Table 1**. We assumed that there was no channel idle time just after a channel idle delimiter.

We used the SAE benchmark which gave communication requirements in a distributed automotive control system. The detailed table for the SAE benchmark was given by Kutlu et al. [6]. The SAE benchmark contains 53 types of communication signals. The SAE benchmark is common among researchers in the research domain of designing an in-vehicle network. The utilization of the SAE benchmark signal set facilitates comparison with other work.

We virtually made 53 benchmark sets using the SAE benchmark set by picking up communication signals from 53 ones of the SAE benchmark set. All communication signals in the SAE benchmark set have their own ID from 1 to 53. The benchmark

Table 1 Network parameters.

Factor	Length
Header w/o BSS	5 B/frame
Header w BSS	45 bits/frame
Trailer w/o BSS	3 B/frame
Trailer w BSS	27 bits/frame
TSS	9 bits/frame
FSS	1 bit/frame
FES	2 bits/frame
Idle delimiter	11 bits/frame
Action point offset	1 MT/frame, 10 bits/frame
BSS	2 bits/frame byte

Table 2 Given operating frequencies for a bus [MHz].

Operating frequencies	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2

set of N communication signals contains Signals 1 to N of the SAE benchmark signals.

We assumed a set of available operating frequencies as shown in **Table 2**. We assumed many more operating frequencies than the FlexRay standard specifies in order to examine the potential for reducing the operating frequency. The FlexRay specification version 3.0.1 specifies three standard bit rates, 2.5, 5.0, and 10.0 Mb/s.

We developed a program that generated an ILP problem instance in the Perl language. The input of the program was the specification of communication signals and the output is an LP format file. We used IBM ILOG CPLEX 12.4 as an ILP solver [5]. The computation platform is an Intel Core i7-3960X-installed PC that is equipped with 64 GB memory. The operating frequency for the Core i7-3960X is 3.30 GHz.

4.2 Optimization Strategy

We adopted three optimization strategies in order to discuss the advantages and disadvantages of the conventional and proposed approaches.

The first optimization strategy *"w/o SM"* obtains a TDMA schedule without slot multiplexing (SM) optimized by solving an ILP problem instance that was given in Ref. [9].

The second one *"w. SM (freq. opt.)"* obtains a TDMA schedule with SM optimized by solving an ILP problem instance whose model was shown in Section 3. All the available operating frequencies are given as $\mathbb{W}$ that constitutes an ILP problem instance. We configured a CPLEX parameter that limited computation time to 16,000 seconds during which the CPLEX could optimize. The CPLEX returns a feasible solution if it obtains it within the limited time. The CPLEX does not return a feasible solution unless it obtains it within the limited time.

The third one *"w. SM (freq. enum.)"* obtains a TDMA schedule with SM optimized by solving enumerated ILP problem instances whose model was shown in Section 3. Operating frequencies and sets of communication signals are enumerated to generate and solve ILP problem instances. We solved enumerated ILP problem instances in ascending order of operating frequencies and the number of communication signals. An optimal solution to an ILP problem instance of a set of communication signals determines the lower bound of the operating frequency of its superset of communication signals. We limited computation time to 14,400 seconds during which the CPLEX could optimize for a single ILP problem instance. The pseudo-code for the third optimization strategy is shown in **Fig. 7**.

4.3 Experimental Results

TDMA schedules were obtained with the three optimization strategies. Operating frequencies that were obtained with the three optimization strategies are shown in **Fig. 8**. We assumed that the size of a payload segment of a frame was 16 bytes in our experiments. In the figure, a circle, an X, and a square indicate optimization strategies *"w/o SM"*, *"w. SM (freq. opt.)"*, and *"w. SM (freq. enum.)"* respectively. The first strategy *"w/o SM"* is a conventional approach and the others are based on our proposal. The first strategy *"w/o SM"* was found to require a higher operating frequency of a bus than the others because the first strategy *"w/o SM"* does not adopt SM. The second strategy *"w. SM (freq. opt.)"* and the third strategy *"w. SM (freq. enum.)"* optimized SM. Both the second strategy *"w. SM (freq. opt.)"* and the third strategy *"w. SM (freq. enum.)"* achieved a lower operating frequency than the first strategy *"w/o SM"*. The second strategy *"w. SM (freq. opt.)"*, however, resulted in a higher operating frequency for the numbers of communication signals, 34, 37, 38, 40, 43, 45, 47, 48, and 51 than the third strategy *"w. SM (freq. enum.)"*. The second strategy was not able to obtain a feasible solution for a set of 53 communication signals within the configured time limit, that is, 16,000 seconds. The third strategy *"w. SM (freq. enum.)"* achieved the lowest operating frequency in any set of communication signals.

Pseudo-code for the optimization strategy *"w. SM (freq. enum.)"*

```
Procedure EnumerateILP(𝕊,𝕎_available)
    // 𝕊 (= {S_1, S_2, ··· , S_Nsig}) is a set of communication signals.
    // 𝕎_available (= {W_1, W_2, ··· , W_Nbw}) is a set of available operating
    // frequencies of a bus. We assume that W_i < W_j if i < j.
begin
    w_index = 1;
    for i = 1 to N_sig do
        𝕊_tmp = {S_1, ··· , S_i};
        feasibility_flag = false;
        for j = w_index to N_bw do
            𝕎_tmp = {W_j};
            Generate and solve an ILP problem instance for 𝕊_tmp and 𝕎_tmp.
            if a feasible solution is found then
                w_index = j;   feasibility_flag = true;
                break;
            endif
        endfor
        if feasibility_flag == false then
            break;
        endif
    endfor
    return (feasibility_flag == true ? w_index : -1);
end
```

Fig. 7 Pseudo-code for the optimization strategy *"w. SM (freq. enum.)"*.

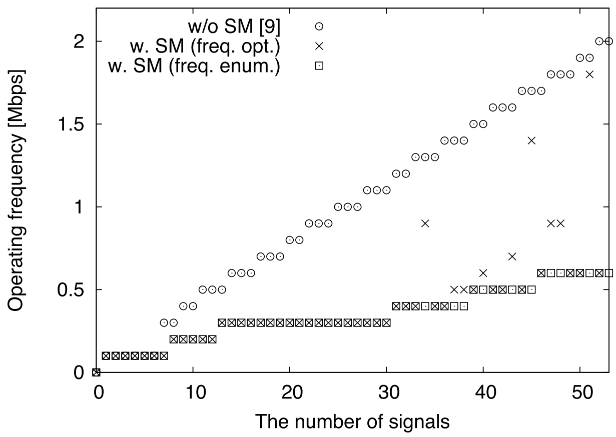

Fig. 8 Operating frequencies obtained with the three optimization strategies.

Figure 9 shows the computation time which each optimization strategy took to search for a feasible solution within a limited

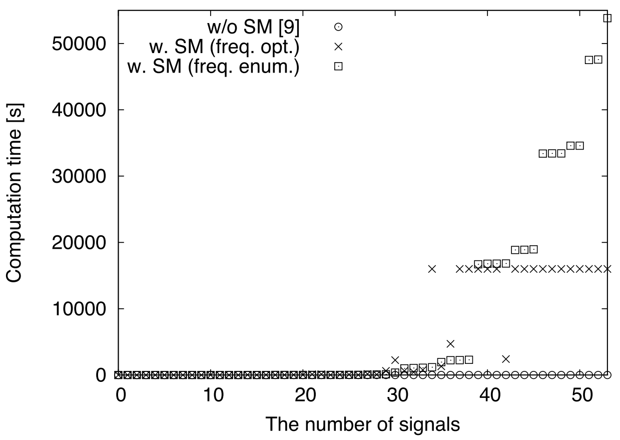

Fig. 9 Computation times of the three optimization strategies.

computation time. The first strategy *"w/o SM"* finished any optimization within 0.04 seconds. The strategy *"w/o SM"* obtained an optimal solution more quickly than any other optimization strategy. The second strategy *"w. SM (freq. opt.)"* obtained an optimal solution for all ILP problem instances of 33 communication signals or less. The second strategy *"w. SM (freq. opt.)"* encountered aborts of optimization, that is, the time limit of 16,000 seconds for many of ILP problem instances of 34 communication signals or more. The third strategy *"w. SM (freq. enum.)"* took about 53,830 seconds to optimize a TDMA schedule of 53 communication signals. The third strategy *"w. SM (freq. enum.)"* encountered an optimization abort that was caused by the time limit of 14,400 seconds for sets of 39 and 46 communication signals. Any other enumerated problem instances were solved within the given time limit.

5. Related Work

Several design approaches were proposed for a time-triggered network in FlexRay networking systems.

Schmidt et al. proposed a message scheduling approach for the static segment of the FlexRay protocol [8]. They assumed cycle-dependent slot assignment. The approach optimized a schedule so that they maximized bandwidth utilization. They assumed that the period and deadline of each and every communication signal are given in not absolute time but communication cycles. This assumption makes it hard to minimize the operating frequency of a bus. Their approach is similar to our approach in considering bandwidth. Their approach, however, is different from our approach in the matter of regarding network bandwidth as a constant. As far as network bandwidth is regarded as a constant, the cost of building a network system cannot be reduced.

Park et al. presented a network parameter optimization approach which optimized the payload length and time for a communication cycle so that the extent of real-time communications was maximized [7]. Their approach is a soft real-time one as it does not necessarily guarantee real-time deadlines.

Zeng et al. proposed a schedule optimization approach for time-triggered systems so that they maximized design extensibility measured by the number of free communication slots and also maximized performance [10]. Their approach put a focus on extensibility for future designs while our approach focuses on the

Table 3 The orders of variables and constraints.

	Variables	Constraints
Conventional model [9]	$O(N_{\mathrm{bw}})$	$O(N_{\mathrm{sig}})$
Proposed model	$O(N_{\mathrm{sig}} N_{\mathrm{bw}} \log^2(N_{\mathrm{cycles}}))$	$O(N_{\mathrm{sig}} N_{\mathrm{bw}} \log^2(N_{\mathrm{cycles}}))$

cost of building a system. We think that their extensibility would not be useful for the mass production of consumer cars.

We previously presented a TDMA scheduling approach that obtained a minimal operating frequency of a bus [9] on the assumption that the slot multiplexing technique was not used. The counterpart of the mathematical model shown in Section 3 is given as follows on the assumption that a payload size is constant.

Minimize the cost function $w = \sum_j W_j x_j$
subject to
(1) $\sum_j x_j = 1$.
(2) $F_{\mathrm{enc}} \left(N_{\mathrm{sig}} \left\lceil \frac{B_i}{P} \right\rceil + 1 \right) \sum_j \frac{x_j}{W_j} \leq D_i,\ 1 \leq \forall i \leq N_{\mathrm{sig}}$.
(3) $F_{\mathrm{enc}} N_{\mathrm{sig}} \left\lceil \frac{B_i}{P} \right\rceil \sum_j \frac{x_j}{W_j} \leq C_i,\ 1 \leq \forall i \leq N_{\mathrm{sig}}$.
Variable
- x_j is a binary variable, $1 \leq \forall j \leq N_{\mathrm{bw}}$.

From the mathematical models shown in Section 3 and mentioned above, the orders of the numbers of variables and constraints are summarized in **Table 3**. In the conventional mathematical model, the number of operating frequency variations N_{bw} affects only the number of variables and the number of communication signals N_{sig} affects only the number of constraints. In the proposed mathematical model, the number of operating frequency variations N_{bw} affects both the numbers of variables and constraints and that of communication signals N_{sig} also affects both the numbers of variables and constraints. The number of communication signals becomes a major factor to determine the size of problem instances when the number of operating frequency variations N_{bw} and that of recurring cycles N_{cycles} are regarded as constant. Our technique proposed in this paper requires a longer optimization time than the conventional one [9] as shown in Fig. 9. Our technique, however, has the potential to achieve a smaller operating frequency of a bus than the conventional one as shown in Fig. 8. We achieved a 72.7% smaller operating frequency at its best than the conventional approach. The comparison between the above orders of variables and constraints suggests that the size of the ILP problem shown in this paper grows more rapidly than the ILP model of the conventional technique [9]. There exists a trade-off between computation time and the operating frequency of a bus.

6. Conclusion

We proposed a design technique in which we optimize cycle-dependent slot assignment so that we can minimize the operating frequency of a FlexRay bus under hard deadline constraints. The operating frequency minimization contributes to cost reduction because system designers can choose slower wire harness. Even if our design methodology requires more computing power for solving the operating frequency minimization problem, the fabrication cost reduced by adopting cheaper parts is much higher than the price of a high-performance computing platform. Our experimental results showed that our optimization technique achieved a

72.7% smaller operating frequency of a bus for given benchmark signal sets at the best than the conventional one [9].

The approach based on ILP has a limitation on the number of communication signals. In our experiment we barely obtained the network system in which the number of communication signals was up to 53. It would be difficult for our ILP-based approach to obtain a good solution for a higher number of communication signals. Our future work includes the development of algorithm for faster computation even for a higher number of communication signals using metaheuristics. Development of metaheuristics for solving the operating frequency minimization problem will contribute to the reduction of optimization time as well as the cost reduction of a large-scale automotive network.

References

[1] Bisschop, J.: AIMMS –optimization modeling (Apr. 2011).

[2] CAN: Control Area Network, available from ⟨http://www.can.bosch.com/⟩.

[3] Cena, G. and Valenzano, A.: Achieving round-robin access in controller area networks, *IEEE Trans. Ind. Electron.*, Vol.49, No.6, pp.1202–1213 (Dec. 2002).

[4] FlexRay Consortium: FlexRay Communications System Protocol Specification Version 3.0.1 (Dec. 2005).

[5] The IBM Corp.: The IBM ILOG CPLEX Optimization Studio V12.4 documentation, available from ⟨http://publib.boulder.ibm.com/infocenter/cosinfoc/v12r4/⟩.

[6] Kutlu, A., Ekiz, H. and Powner, E.T.: Performance Analysis of MAC Protocols for Wireless Control Area Network, *Proc. International Symposium on Parallel Architectures, Algorithms, and Networks*, pp.494–499 (June 1996).

[7] Park, I. and Sunwoo, M.: FlexRay Network Parameter Optimization Method for Automotive Applications, *IEEE Trans. Industrial Electronics*, Vol.58, No.4, pp.1449–1459 (Apr. 2011).

[8] Schmidt, K. and Schmidt, E.G.: Message Scheduling for the FlexRay Protocol: The Static Segment, *IEEE Trans. Vehicular Technology*, Vol.58, No.5, pp.2170–2179 (June 2009).

[9] Sugihara, M. and Iwanaga, A.: Minimization of FlexRay Bus Bandwidth for Hard Real-Time Applications, *IPSJ Journal of Information Processing*, Vol.21, No.1, pp.46–52 (Jan. 2013).

[10] Zeng, H., Matale, M.D., Ghosal, A. and Sangiovanni-Vincentelli, A.: Schedule Optimization of Time-Triggered Systems Communicating over the FlexRay Static Segment, *IEEE Trans. Industrial Informatics*, Vol.7, No.1, pp.1–17 (Feb. 2011).

Makoto Sugihara was born in Tokyo, Japan in 1974. He received his B.E., M.E. and Ph.D. degrees in Computer Science and Communication Engineering from Kyushu University, Japan in 1996, 1998 and 2001, respectively. He joined the Fujitsu Laboratories, Japan, in 1998. From 2002 to 2003, he was a visiting researcher at Duke University, NC, USA. From 2003 to 2007, he was a Researcher at the ISIT, Fukuoka, Japan, and was also a Visiting Associate Professor at Kyushu University, Fukuoka, Japan. From 2007 to 2009, he was a Lecturer at Toyohashi University of Technology, Aichi, Japan. From 2009 to 2011, he was an Associate Professor at Toyohashi University of Technology, Aichi, Japan. He is currently an Associate Professor at the System LSI Research Center, Kyushu University, Fukuoka, Japan. His research interests include design methodologies for VLSI chips and embedded systems. He is a member of the IEEE and the IEEE Computer Society, the IEICE, and the IPSJ. He received the IPSJ 40th Anniversary Best Paper Award, the Hiroshi Ando Memorial Award, and the Funai Information Science Encouragement Award in 2001, 2008, and 2009 respectively.

Akihito Iwanaga was born in Yamaguchi, Japan in 1988. He received his B.E. degree in Computer Science and Electronic Engineering from the National Institution for Academic Degrees and University Evaluation, Japan in 2011. He also received his M.E. degree from the Department of Automotive Science, Graduate School of Integrated Frontier Science, Kyushu University, Japan in 2013. His research activities include designing a hard real-time system and an automotive network. He is a member of the IPSJ and the JSAE.