# FlexRay Network Parameter Optimization Method for Automotive Applications

Inseok Park, *Student Member, IEEE*, and Myoungho Sunwoo, *Member, IEEE*

***Abstract*—This paper describes a new FlexRay network parameter optimization (NPO) method consisting of two straightforward steps: design the lengths of the static (ST) slot and the communication cycle. In the first step, we formulate an optimal problem for designing the ST slot length. The bandwidth limitation, composed of protocol overhead and unused network resources, is considered as optimal criterion for efficient network usage. With the exploration of design space (i.e., ST slot lengths which satisfy the constraints of the FlexRay specifications), the optimal ST slot length is determined. Based on the result of the first step, the optimal communication-cycle length is designed in the next step. In this step, we proposed an algorithm for analyzing the worst case response times (WCRTs) for the ST and dynamic frames. Using this algorithm, the optimal communication-cycle length with minimum network delays is derived. The NPO method formalizes the FlexRay network configuration process in two steps for simplicity. Furthermore, the method enables the efficient usage of FlexRay bus with minimum WCRT. Finally, vehicle chassis control system based on FlexRay network is designed using the proposed NPO method. In this example system, we verified the parameter-optimization algorithm, and analysis results are confirmed.**

***Index Terms*—Fieldbus, FlexRay, in-vehicle network, real-time distributed control system.**

## I. Introduction

IN THE LAST few decades, as the number of electronic control units (ECUs) in vehicles has increased, conventional point-to-point links have been replaced with linear bus network, such as controller area network (CAN) and local interconnect network, for the purpose of network efficiency [1], [2]. Currently, it is common to have about 70 ECUs exchanging around 2500 signals in a luxury car [3]. Because of the various advantages of CAN, such as robustness and flexibility, it is the most popular network protocol in the automotive industry. With the increasing number of ECUs in a vehicle, network usage is growing at a considerable rate. However, this growth has caused a significant problem due to the bandwidth limitation of CAN [4], [5]. Additionally, CAN suffers from a low degree of determinism and fault tolerance for advanced automotive applications such as x-by-wire systems [6].

Manuscript received September 11, 2009; revised January 28, 2010; accepted March 9, 2010. Date of publication May 10, 2010; date of current version March 11, 2011. This work was supported in part by the Brain Korea 21 Project in 2010, by the Energy Resource R&D Program (2006ETR11P091C) under the Ministry of Knowledge Economy, Republic of Korea, and by the Korea Institute for Advancement in Technology (KIAT) through the Workforce Development Program in Strategic Technology Project.

I. Park is with the Automotive Control and Electronics Laboratory, Department of Automotive Engineering, Hanyang University, Seoul 133-791, Korea.

M. Sunwoo is with the Department of Automotive Engineering, Hanyang University, Seoul 133-791, Korea (e-mail: msunwoo@hanyang.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2010.2049713

Therefore, fast, deterministic, and fault-tolerant network protocols are required. In order to satisfy these requirements, a consortium composed of a number of car manufacturers and suppliers proposed the FlexRay protocol in 2004 [7]. FlexRay is a heterogeneous protocol consisting of static (ST) and dynamic (DYN) segments. In an ST segment, a high degree of determinism can be provided by using a time division multiple access (TDMA) scheme. Optionally, a DYN segment can be used for enhancing the flexibility of communication. FlexRay combines the advantages of time-triggered and event-triggered communication protocols: determinism and flexibility. Furthermore, with a bandwidth of 10 Mb/s, FlexRay has ten times faster transmission rate, theoretically. FlexRay also supports redundant channels for fault-tolerant communication. At present, FlexRay is expected to enable various automotive applications, such as active safety, passive safety, powertrain management, integrated chassis control, and driver-assistance systems [8].

Although FlexRay presents appropriate features for advanced automotive applications, its complicated operation makes it difficult to configure [9]. FlexRay requires configuration of about 70 parameters, so it is regarded as a difficult and time-consuming task. Today, there are a few commercial tools for designing FlexRay-based control systems (e.g., dSPACE, Elektrobit, and Vector) that provide functions for fault-free configuration, but design optimization is not guaranteed. Therefore, a method that can provide design optimization with regard to performance of the system is required [10]–[12].

Several studies have presented methods to optimize the configuration of a FlexRay network. Pop *et al.* proposed a heuristic method based on the FlexRay timing model [10]. They defined a general FlexRay response-time model and proposed a network parameter design process; however, their approach is limited by their time-triggered system model. Although the time-triggered architecture can improve the control performance due to the high degree of determinism, it has disadvantages such as the lack of flexibility and the restrictive design procedure. Furthermore, it requires time-triggered operating systems (e.g., OSEKtime [11]) which can synchronize the schedules of task and communication. Unfortunately, most legacy automotive applications are implemented using an event-based architecture. This incompatibility means that a large amount of development cost and time are required to migrate from conventional CAN-network-based applications to a time-triggered architecture [13]. Therefore, the model proposed by Pop *et al.* is not suitable for automotive applications. Furthermore, bandwidth utilization was not considered.

Hagiescu *et al.* suggested a compositional framework for the performance analysis of ECUs interconnected by a FlexRay bus [14]. They provided a method to analyze the performance of the FlexRay bus using the maximum end-to-end delay, required amount of buffer, and utilization. However, they only considered the DYN segment in the performance analysis. Therefore, this exclusion imparted a significant limitation on the study as the ST segment plays a dominant role in FlexRay communication.

Although the previous studies of Pop and Hagiescu presented promising directions, they are not suitable for automotive applications. In contrast to the previous studies, we suggest a novel configuration method by considering both the network utilization and the ST segment. Furthermore, our method is based on a practical system model that assumes that tasks are not synchronized with the communication protocol (e.g., tasks of each node are executed by OSEK-OS, which does not provide synchronization service between tasks and communication protocol) [12]. Because it does not require synchronization to communication protocol, CAN-based legacy applications can easily be migrated to FlexRay-based system.

Our configuration method is composed of two steps for determining the optimal ST slot and communication cycle length. In the first step, the ST slot length is determined in order to maximize network bandwidth. Based on the results of the first step, a procedure to determine the optimal communication cycle length to minimize worst case response time (WCRT) is introduced. These two steps are defined as a network parameter optimization (NPO) method in this paper. Using our method, it is possible to design the optimal configuration of FlexRay networks.

This paper is structured as follows. In Section II, the basic principles of FlexRay are described. Section III introduces the NPO method for FlexRay configuration. A case study using the NPO method is presented in Section IV. Finally, we conclude this paper in Section V.

## II. Basic Principles of FlexRay

The principles of FlexRay can be classified into timing hierarchy, frame format, coding element, and media-access control (MAC). Here, we give an overview of FlexRay to understand the NPO method. Further description of this protocol is explained in [7] and [15].

### A. Timing Hierarchy

The timing hierarchy of the FlexRay protocol is composed of six levels: microtick ($\mu$T), macrotick (MT), slot, segment, single communication cycle, and 64 communication cycles. The bottom level in Fig. 1 is the $\mu$T level, which is generated by the local clock of each network node. A group of $\mu$Ts builds an MT. In a FlexRay bus, the MT presents global time unit. Therefore, the timing hierarchy which is above the MT level is represented by the number of MTs.

In a slot level, there are a number of ST slots and minislots. The ST slot is used for periodic data communication. Each ST slot has a frame that transmits at every communication cycle.

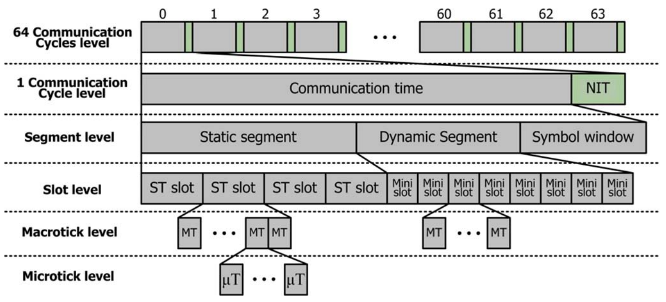


Fig. 1. FlexRay timing hierarchy.

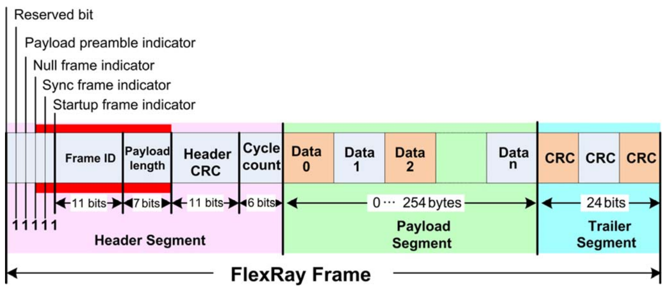


Fig. 2. Frame format.

Otherwise, minislots describe a unit length of DYN slots, which are used for nonperiodic communication.

A single communication cycle consists of ST, DYN segments, symbol window, and network idle time (NIT). The same communication cycle is repeated from cycle counter 0 to 63, as shown in Fig. 1.

### B. Frame Format

The frame format of FlexRay is classified into three segments: header, payload, and trailer, as shown in Fig. 2. The header segment includes specific information about each frame. It consists of four indicators: the frame identifier (ID), the payload length, the header's cyclic redundancy check (CRC), and the cycle count. The payload segment contains up to 254 B of communication data. The trailer segment includes a 24-b CRC that can detect errors in the header and payload segments.

### C. Coding Element and Protocol Overhead

In order to transmit frames to the FlexRay bus, the frames must be encoded with five elements: a transmission start sequence (TSS), a frame start sequence (FSS), a byte start sequence (BSS), a frame end sequence (FES), and a DYN trailing sequence (DTS). These are automatically encoded by the communication controller. An example of encoded frame of ST segment is shown in Fig. 3. The TSS and FSS are placed in front of the frames. Every byte of the frame has a BSS. The FES is located at the end of each frame. In the case of the DYN segment, the DTS is added after the FES.

In this paper, all frame elements except for the payload are regarded as protocol overheads. When a set of data attempts to transmit through the FlexRay bus, additional coding elements (i.e., protocol overheads) are required for communication. Table I denotes the protocol-overhead list for this paper.

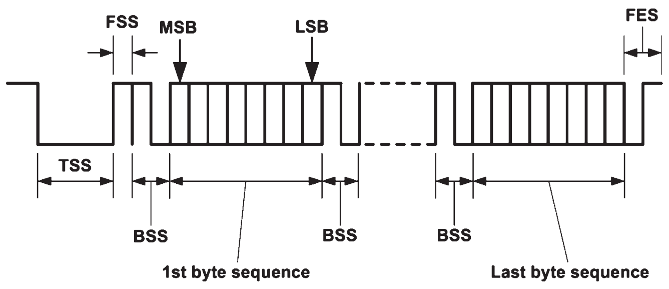


Fig. 3. Coding element.

TABLE I
PROTOCOL OVERHEAD ELEMENTS

| Element | Length |
|---|---|
| Header and Trailer (BSS include) | 80 bits/frame |
| TSS | 9 bits/frame |
| FSS | 1 bit/frame |
| FES | 2 bits/frame |
| DTS | 2 bits/frame |
| Idle delimiter | 11 bits/frame |
| Action point offset | 1 MT/frame, 10 bits/frame |
| BSS | 2 bits/payload byte |

## *D. MAC*

The MAC of FlexRay is classified by TDMA and flexible FTDMA. In an ST segment, bus access is controlled by TDMA principles. ST frames to be transmitted in the ST segment are assigned to a specific node and slot during the configuration phase.

Each ST slot has the same time duration, and all of the communication activities in an ST segment are controlled by an ST schedule. Therefore, periodic network signals are generally assigned to ST frames. Otherwise, the communication activities in a DYN segment are dependent on different uses for diagnosis, flash downloading, and calibration. Therefore, the lengths of the DYN slots are determined by each frame.

The most important rule in the MAC of FlexRay is the match between frame ID and slot counter value. In Fig. 4, this rule is shown with a timing diagram of the task execution and FlexRay communication cycle. Suppose that there are a single FlexRay bus, two network nodes, and four frames. In this example, we assumed that each frame transmission is requested at the end of the specific task execution. For example, the task $Task_{ST2}$ requests transmission of the ST frame $frST_2$ at the end of task execution, and $frST_2$ is transmitted at ST slot 2.

Although $frST_2$ is transmitted in the first cycle, $frST_1$ is delayed until the second communication cycle because $frST_1$ is not prepared to transmit before the beginning of ST slot 1. It has to wait until the ST slot 1 of the next cycle.

DYN frames can also be transmitted when their IDs are identical to the value of the DYN slot counter. However, they differ from the ST frames in that they are prioritized according to frame ID, where a lower ID has a higher priority. In Fig. 4, two DYN frames (i.e., $frDYN_6$ and $frDYN_7$) are requested in the ST segment of the first communication cycle. Although $frDYN_7$ is requested prior to $frDYN_6$, it is transmitted after $frDYN_6$. Therefore, it is possible that the low-priority DYN frames cannot be transmitted in a cycle.

## III. NPO METHOD

### *A. System Model*

In this paper, we suppose a system model including nodes and a FlexRay bus. Each node is composed of a host processor, a controller host interface (CHI), and a FlexRay communication controller.

In terms of synchronization between the application layer (i.e., tasks in a host) and the communication layer, which includes CHI and the communication controller, the system model is classified into synchronous (SYNC) and asynchronous (ASYNC) models. In the case of the SYNC system model, the task scheduling and FlexRay communication are synchronized. In the ASYNC system, the task and communication schedules are decoupled. An example system is shown in Fig. 5. This system consists of a distributed single control loop which includes two nodes, three tasks, and two network signals.

In the case of a SYNC system, the task scheduling is synchronized with a FlexRay communication cycle, as shown in Fig. 6. At the beginning of every cycle, the $Task_{Sensor}$ is activated. After the execution of the task, $Task_{Sensor}$ requests the transmission of $frST_2$. $Task_{Controller}$ computes the control input signal using the received sensor signal (i.e., $frST_2$). The control input signal is transferred through $frST_5$. Using the information in $frST_5$, $Task_{Actuator}$ controls an actuator. These operations are repeated in every communication cycle without any exception. As described in this example, the entire design process should be carried out offline. Task scheduling, the worst case execution time of tasks, and the communication schedule have to be strictly defined during the design phase of the SYNC system. Although this type of system model can provide a high degree of determinism and composability, it has difficulties with the migration of legacy applications and considerable redesign cost for a time-triggered platform [13], [16].

The ASYNC system, however, is based on fixed-priority scheduling, and its scheduling is decoupled from the FlexRay communication. Fig. 7 shows an example that describes the behavior of the ASYNC system model. Because the scheduling of tasks is not synchronized with the FlexRay communication cycle, there are jitters in the activation timings of the tasks. In this ASYNC example, it is possible that $Task_{Sensor}$ is activated slightly after the start of the communication cycle. Due to the activation of $Task_{Sensor}$, $frST_2$, which is generated in the first cycle, cannot be transferred in that cycle. In the next cycle, the waiting $frST_2$ is skipped due to the newly generated $frST_2$ of the second communication cycle, as shown in Fig. 7.

As illustrated in this example, network delays and signal loss are possible. This behavior can induce the performance degradation of networked control systems [17]–[21]. However, this makes it easy to reuse CAN-based event-triggered legacy applications for FlexRay-based control systems because the platform is identical. Therefore, the ASYNC system model is a more practical solution than the SYNC system, so the ASYNC system was selected as the target system in this paper.

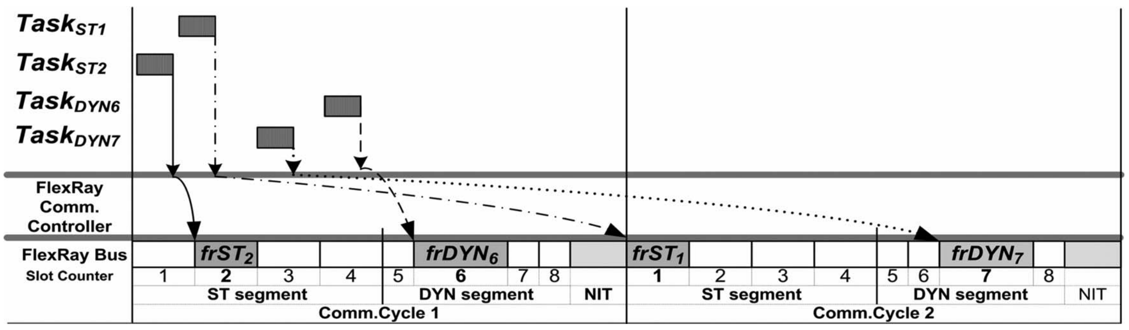


Fig. 4. Example of FlexRay's MAC.

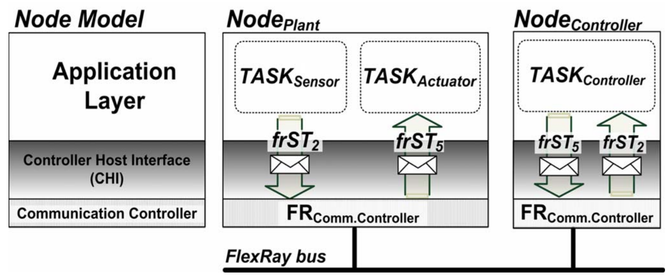


Fig. 5. System model example.

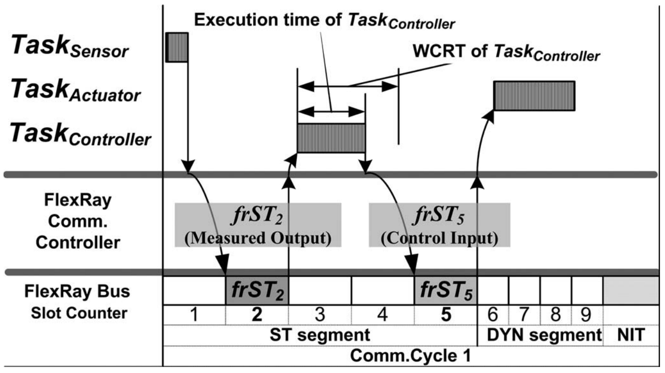


Fig. 6. SYNC system example.

## B. Assumptions

Our method was based on several assumptions to simplify the optimal configuration method. Some assumptions, such as the duration of MT, minislot, action-point offsets, and DYN slot idle phase were picked up from examples of commercial FlexRay network configuration tool [22]. The assumptions made regarding the low-level application-independent parameters and the message-request conditions were as follows:

1) In an ST segment, only periodic messages are requested, and their periods are longer than the communication cycle.
2) In order to simplify the design problem, multiple generations of nonperiodic messages are not allowable in a same communication cycle. This is because only the messages for diagnosis and firmware update are typically allocated to DYN frames in automotive applications.
3) Nonperiodic messages are assigned to DYN frames with regard to their priority levels. In the case of high priority, nonperiodic message has a low numbered DYN frame ID.
4) Baud rate is 10 Mb/s with a bit-time of 0.1 $\mu$s/b.
5) Duration of an MT is 1 $\mu$s.
6) Duration of a minislot is two MT.
7) Duration of an SV slot action-point offset is one MT.
8) Duration of a minislot action-point offset is one MT.
9) Duration of a DYN slot idle phase is one MT.
10) Duration of an NIT segment is 200 MT.

## C. Network Configuration With the Parameter Optimization Method

In this paper, we propose a two-step NPO method for designing two core parameters: ST slot length and communication cycle.

*1) Optimal ST Slot Length:* In order to transmit ST messages using FlexRay, frame construction should first be performed. Due to the uniform ST slot (or frame) length, diverse frame-construction results are possible. This causes a design problem in determining the ST slot length. In this paper, we define bandwidth limitation caused by frame construction as an optimal criterion.

There are two types of bandwidth limitations: protocol overhead and unused network resource. The protocol overhead, which consists of all communication elements except for payload, is listed in Table I. As introduced in Section II-C, the protocol overhead is an inevitable part for data communication. Another bandwidth limitation is unused network resource induced by uniform ST slot and various message lengths.

Fig. 8 shows an example for protocol overhead and unused network resources. There are three ST messages: $msgST_1$ (10 B), $msgST_2$ (8 B), and $msgST_3$ (4 B). If the payload length of the ST frame is determined by the longest message (i.e., 10 B), $msgST_1$, $msgST_2$, and $msgST_3$ are allocated, respectively, to $frST_1$, $frST_2$, and $frST_3$, as shown in Fig. 8(a).

In this figure, the protocol overhead is represented by the horizontal-lined region, and the unused network resource is shown by the shaded region. As shown in Fig. 8(a), each ST frame has the same amount of protocol overhead.

However, the unused network resource depends on the payload length. In the case of the 5-B payload length, there are different frame-construction results. As shown in Fig. 8(b), $msgST_1$ is allocated in $frST_1$ and $frST_2$, and $msgST_2$ is divided into $frST_3$ and $frST_4$. Although the amount of protocol overhead increased, the unused network resource is decreased due to the short payload length.

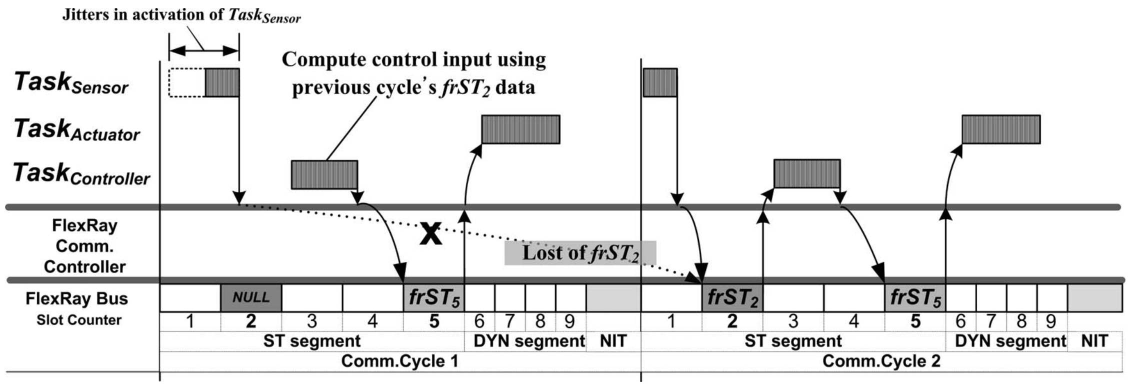


Fig. 7. ASYNC system example.

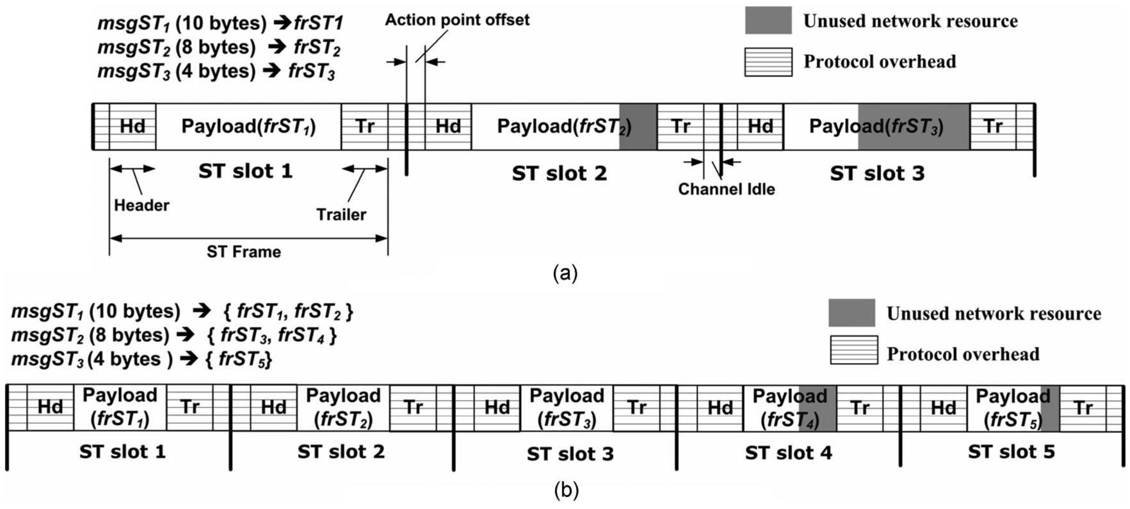


Fig. 8. ST slot configuration example. (a) Case 1: payload (10 B). (b) Case 2: payload (5 B).

As shown in Fig. 8, the payload length determines the frame-construction result and bandwidth-limitation behaviors. In order to maximize the efficiency in the FlexRay bus, an optimal ST slot length should be designed by minimizing the bandwidth limitation.

When a set of ST messages is given, the number of ST frames can be determined by a variable $x$, as in

$$\sum_{k=1}^{n} \left\lceil \frac{msgST_k}{x} \right\rceil \tag{1}$$

where $msgST_k$ is the byte length of each ST messages, $n$ is the number of ST messages, and $x$ is the payload length for ST frames. Using (1), a cost function for the protocol overhead can be calculated by

$$J_O = (C_o + x \cdot \mathrm{BSS}) \cdot \sum_{k=1}^{n} \left\lceil \frac{msgST_k}{x} \right\rceil \tag{2}$$

where $J_O$ is the cost function of the protocol overhead, $C_O$ is the summation of the protocol-overhead length, including the header, trailer, TSS, FSS, FES, idle delimiter, action-point offset, and safety margin. BSS is the additional coding element for each byte of the frame. As described in (2), the cost for the protocol overhead is closely related to the total number of ST frames. If we choose a long payload length $x$, the ST messages are divided into a smaller number of ST frames, so we can reduce the protocol overhead cost. However, if the payload length $x$ is long, a large amount of unused network resource will result. Therefore, the protocol overhead and unused network resources must be considered concurrently.

In this paper, the cost for unused network resource is described by the amount of unused time in a communication cycle in the FlexRay bus. Using (1), the cost function of unused network resource $J_U$ is defined as follows:

$$J_U = \sum_{k=1}^{n} \left( x \cdot \left\lceil \frac{msgST_k}{x} \right\rceil - msgST_k \right) \cdot (8 + \mathrm{BSS}). \tag{3}$$

Based on the definitions of the two bandwidth limitations, the cost function for optimal ST slot is defined as the summation of the cost function of the protocol overhead $J_O$ and unused network resource $J_U$

$$J_{\mathrm{STslot}} = J_O + J_U \tag{4}$$

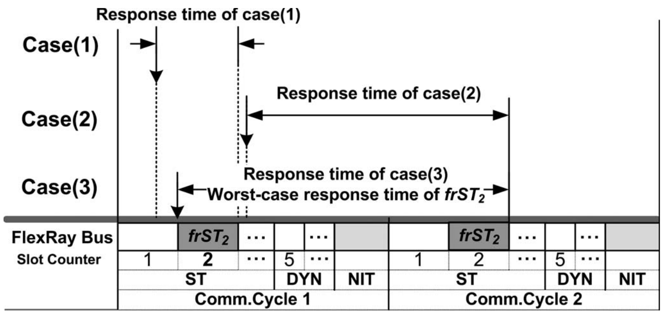


Fig. 9. Response-time examples for ST frames.

where $J_{\mathrm{STslot}}$ is the cost function for designing an optimal ST slot length. Consequently, the ST slot-length optimization problem is formulated as follows:

$$\begin{aligned} \min \quad & J_{\mathrm{STslot}} = J_O + J_U \\ \text{subject to} \quad & X = \left\{ x | x = 2 \cdot i, \;\; for\; i = 1, 2, \ldots, \max\left( \left\lceil \frac{msgST_k}{2} \right\rceil \right) \right\} \\ & k = 1, 2, \ldots, n \end{aligned} \tag{5}$$

where $n$ is the number of ST messages and $X$ is the design space for the payload length $x$.

*2) Optimal Communication-Cycle Length:* One of the most important parameters in FlexRay is the communication-cycle length. It determines the communication period and affects the frame's response time. The response time has some variations caused by the asynchronization between the applications and the FlexRay network. Therefore, the request timings for frame transmission in the application layer drifted from the FlexRay communication schedule. This ASYNC behavior can cause problems such as significant response-time jitter. Jitter is one of the reasons that degrades the control performance in distributed control systems [23], [24]. In Fig. 9, a response-time jitter example is shown. There are three cases which have different transmission request times for ST frames. These perturbed timings induce response-time jitter, as shown in Fig. 9. Therefore, the minimum response time must be considered when the communication-cycle length is being determined. Pop *et al.* proposed a general response-time model for the FlexRay frame [10]. It consists of the communication time and two delays as described in

$$R_i = \sigma_i + w_i + C_i \tag{6}$$

where $C_i$ is the communication time of frame $i$, $\sigma_i$ is the longest delay suffered from request timing to the end of the first communication cycle, and $w_i$ is the worst case delay induced by frames with lower ID.

In this paper, the WCRT of the ST and DYN frames are considered to find the optimal communication-cycle length. In the case of ST segments, the WCRT of an ST frame depends on the communication-cycle length. Fig. 7 shows the response times of ST frames for three different cases. The response time of the ST frame has the longest delay, as shown in case 3 of Fig. 9, when the transmission request time is identical to the beginning of the ST slot. Using (6), the WCRT of the ST frame is modeled by (7). The summation of $\sigma_i$ and $w_i$ is the communication-cycle length. Therefore, the WCRT of the ST frame is derived by the summation of a single communication cycle and ST slot length

$$\begin{aligned} R_{ST_i} &= (\sigma_i + w_i) + C_i \\ &= (C_{\mathrm{ST}} + C_{\mathrm{DYN}} + C_{\mathrm{NIT}}) + C_i \\ &= C_{\mathrm{Comm.Cycle}} + C_i \end{aligned} \tag{7}$$

where $R_{STi}$ is the WCRT of ST frame $i$. $C_{\mathrm{ST}}$, $C_{\mathrm{DYN}}$, and $C_{\mathrm{NIT}}$ are the durations of the ST segment, DYN segment, and NIT, respectively, and $C_{\mathrm{Comm.Cycle}}$ is the communication-cycle length. The summation of the WCRT of the ST frames is considered as a cost function for optimal communication-cycle length.

As denoted in (7), the $R_{STi}$ is proportional to the length of the communication cycle. In this paper, we assume that the ST segment length is determined by the optimal ST slot length and the number of ST frames. Therefore, the $C_{\mathrm{ST}}$ is calculated by

$$C_{\mathrm{ST}} = ST_{\mathrm{slot}} \cdot n_s \tag{8}$$

where $ST_{\mathrm{slot}}$ is the duration of the ST slot and $n_S$ is the number of ST slots. Using the result of the ST slot optimization, the $ST_{\mathrm{slot}}$ and $n_S$ are described by

$$ST_{\mathrm{slot}} = \lceil \{C_o + x \cdot (8 + \mathrm{BSS})\} / MT \rceil \cdot MT \tag{9}$$

$$n_s = \sum_{k=1}^{n} \lceil msgST_k / x \rceil \tag{10}$$

where $MT$ is the number of bits per MT.

The $C_{\mathrm{DYN}}$ is defined by the number of minislot, and the duration of a minislot is defined in

$$C_{\mathrm{DYN}} = d \cdot MS \tag{11}$$

where $d$ is the number of minislots and $MS$ is the number of bits in a minislot. In this paper, we assumed that the duration of NIT is 200 $MT$

$$C_{\mathrm{NIT}} = 200 \cdot MT. \tag{12}$$

Therefore, the $C_{\mathrm{Comm.Cycle}}$ is formulated by $C_{\mathrm{ST}}$, $C_{\mathrm{DYN}}$, and $C_{\mathrm{NIT}}$ (assumed to be 200 $MT$) as follows:

$$C_{\mathrm{Comm.Cycle}} = ST_{\mathrm{slot}} \cdot n_s + d \cdot MS + C_{\mathrm{NIT}}. \tag{13}$$

As described in (13), the $C_{\mathrm{Comm.Cycle}}$ is a function of the number of DYN slots. Therefore, (7) is reformulated as follows:

$$R_{ST_i} = C_{\mathrm{Comm.Cycle}} + C_i. \tag{14}$$

The communication-cycle length is determined by the variation using the following constraint:

$$d_{\min} \le d \le d_{\max} \tag{15}$$

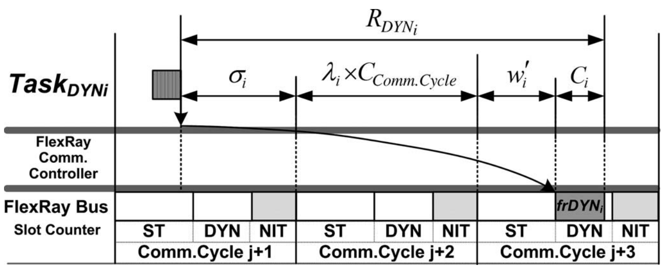


Fig. 10. WCRT model of a DYN frame.

where $d_{\min}$ and $d_{\max}$ are the minimum and maximum number of minislots, respectively. The value of $d_{\max}$ is declared in the FlexRay specification as in [15]

$$d_{\max} = 7980. \tag{16}$$

$d_{\min}$ is derived by the upper bound of the minimum requirement for each DYN frame transmission. It is given by

$$d_{\min} = \max\left(w_{DYN_i} + C_{DYN_i}\right) \tag{17}$$

where $w_{DYNi}$ is the number of waiting minislots and $C_{DYNi}$ is the required minislots to transmit DYN frame $i$. The number of waiting minislots is

$$w_{DYN_i} = ID_{DYN_i} - ID_{ST_{nS}} - 1 \tag{18}$$

where $ID_{DYNi}$ is the ID of DYN frame $i$, and $ID_{STns}$ is the last ST slot ID. We describe the number of minislots for transmitting DYN frame $i$ as

$$C_{DYN_i} = \left( \left\lceil \frac{(Fr_{DYN_i} \cdot (8 + \text{BSS}) + MS_{\text{APO}} + C_o + \text{DTS}}{MS} \right\rceil + 1 \right) \tag{19}$$

where $Fr_{DYNi}$ is the payload size of DYN frame $i$, $MS_{\text{APO}}$ means the action point offset of the minislot, and DTS is the DYN trailing sequence.

Another criterion for designing the optimal communication-cycle length is the WCRT of the DYN frame. It is defined by the critical instant [25], [26]. When all the DYN frames are requested simultaneously, each DYN frame has the longest response time. In this paper, the summation of the WCRT of the DYN frames is considered as a cost function for the optimal communication-cycle length. The WCRT of the DYN frame is described by Fig. 10 and

$$R_{DYN_i} = \sigma_i + \lambda_i \cdot C_{\text{Comm.Cycle}} + w_i' + C_i \tag{20}$$

where $R_{DYNi}$ is the WCRT of the DYN frame $i$, $\lambda_i$ is the delayed communication-cycle number, and $w_i'$ is the delay suffered from the beginning of the last cycle to the transmission start of frame $i$.

In order to analyze the WCRT of the DYN frames, we developed a simulation algorithm that implemented FTDMA principles for communication in a DYN segment. Using this algorithm, we can find out the WCRT of DYN frames and their

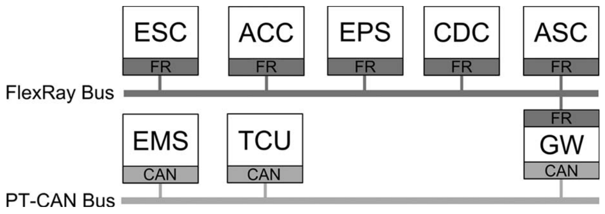


Fig. 11. Case study architecture.

schedulability. The following pseudocode describes the details of the algorithm.

```
01 WHILE Transmission of all $frDYN$s is finished
02    $CNT_{MiniSlot} = 1$
03    $CNT_{DynSlot} = n_s + 1$
04    WHILE $CNT_{MiniSlot} \leq d$
05       IF $fr_{DYN}[CNT_{Dynslot}]$ exists THEN
06          INCREMENT $CNT_{DynSlot}$
07          COMPUTE $CNT_{MiniSlot}$
08          COMPUTE $\sigma$, $\lambda$ and w' of
                $frDYN[CNT_{DynSlot}]$
09       ELSE
10          INCREMENT $CNT_{MiniSlot}$
11          INCREMENT $CNT_{DynSlot}$
12       ENDIF
13    ENDWHILE
14    INCREMENT $CNT_{Comm.Cycle}$
15 ENDWHILE
```

| | |
|---|---|
| $^*frDYN[i]$ | *ID of the ith DYN frame* |
| $^*CNT_{MiniSlot}$ | *minislot counter* |
| $^*CNT_{DynSlot}$ | *DYN slot counter* |
| $^*CNT_{Comm.Cycle}$ | *communication cycle counter* |

Based on the definition of WCRTs for ST and DYN frames, we propose a cost function for finding the optimal communication cycle. At first, the cost function of ST frames is calculated as

$$J_{\text{ST}} = \sum_{i=1}^{n_s} R_{ST_i} \tag{21}$$

where $J_{\text{ST}}$ is the cost function of the ST frames and $n_s$ is the number of ST frames. Similarly, the cost of DYN frames $(J_{\text{DYN}})$ is described by

$$J_{\text{DYN}} = \sum_{i=1}^{n_d} R_{DYN_i} \tag{22}$$

where $n_d$ is the number of DYN frames.

As a result, the cost function for the optimal communication cycle is denoted by the weighted summation of $J_{\text{ST}}$ and $J_{\text{DYN}}$ as follows:

$$\begin{aligned} \min \quad & J_{\text{Comm.Cycle}} = \omega_1 \cdot J_{\text{ST}} + \omega_2 \cdot J_{\text{DYN}} \\ \text{subject to} \quad & \omega_1 + \omega_2 = 1 \\ & n_{ST_{fail}} + n_{DYN_{fail}} = 0 \end{aligned} \tag{23}$$

where $\omega_1$ and $\omega_2$ are the weighting factors for $J_{\text{ST}}$ and $J_{\text{DYN}}$.

TABLE II
EXAMPLE MESSAGE SET

| Node | msgST | Deadline [ms] | Length [byte] | msgDYN | Deadline [ms] | Length [byte] |
|---|---|---|---|---|---|---|
| ESC[a] | *ESC_Status* | 1.25 | 4 | *ESC_Diag* | 1.5 | 16 |
| | *ESC_WheelRot* | 1.5 | 8 | | | |
| | *ESC_WheelSpeed* | 1.5 | 8 | | | |
| | *ESC_DampReq* | 2 | 3 | | | |
| ACC[b] | *ACC_Status* | 2 | 6 | *ACC_Diag* | 2 | 16 |
| | *ACC_EMSReq* | 2 | 8 | | | |
| GW[c] | *GW_Status* | 4 | 16 | *GW_Diag* | 2 | 16 |
| | *EMS_AirConStatus* | 7 | 3 | *EMS_Diag* | 2 | 32 |
| | *EMS_Status* | 2 | 8 | | | |
| | *EMS_IdleSpeed* | 2 | 6 | | | |
| | *EMS_AccPedalPos* | 1.5 | 7 | | | |
| | *EMS_Temp* | 10 | 3 | | | |
| | *EMS_Torq* | 10 | 1 | | | |
| | *TCU_Status* | 10 | 8 | *TCU_Diag* | 3 | 32 |
| | *TCU_Temp* | 5 | 5 | | | |
| | *TCU_TurSpeed* | 5 | 2 | | | |
| | *TCU_TorqReq* | 5 | 11 | | | |
| EPS[d] | *EPS_Status* | 20 | 1 | *EPS_Diag* | 3 | 16 |
| | *EPS_StrAng* | 10 | 5 | | | |
| CDC[e] | *CDC_Status* | 2.5 | 8 | *CDC_Diag* | 3 | 16 |
| ASC[f] | *ASC_Status* | 10 | 1 | *ASC_Diag* | 4 | 16 |

[a]ESC(Electronic Stability Control), [b]ACC(Adaptive Cruise Control), [c]GW(Gateway), [d]EPS = Electric Power Steering, [e]CDC (Continuous Damping Control), [f]ASC(= Air-Suspension Control)

$n_{STfail}$ and $n_{DYNfail}$ are the number of unschedulable ST and DYN frames, respectively. As described in (23), designing an optimal communication-cycle length is a multiobjective optimal problem. We can find the optimal communication-cycle length from the exploration of $d$ in the constraints such as $\omega_1$, $\omega_2$, $d_{\min}$, $d_{\max}$, $n_{STfail}$, and $n_{DYNfail}$.

## IV. Case Study: FlexRay Network-Based Chassis Control System

We now present a case study of a chassis control system for a vehicle. This example system is connected to the powertrain CAN network through a gateway node. It plays a role in exchanging information between the FlexRay-based chassis control system and the CAN-based powertrain system. Fig. 11 shows the system architecture of this example.

In this paper, we focused on the FlexRay-based chassis control system. It consists of six FlexRay nodes, 21 periodic ST messages, and eight diagnostic DYN messages, as noted in Table II. Using this example system, we evaluated the NPO method discussed in Section III.

The first step of the NPO method is to make the optimal ST slot-length decision. Using (2) and (3), the cost functions of the protocol overhead and the unused network resource are expressed as follows:

$$J_O = (113 + 2 \cdot x) \cdot \sum_{k=1}^{21} \left\lceil \frac{msgST_k}{x} \right\rceil \tag{24}$$

$$J_U = 10 \cdot \sum_{k=1}^{21} \left( x \cdot \left\lceil \frac{msgST_k}{x} \right\rceil - msgST_k \right). \tag{25}$$

We formulate the optimal ST slot-length problem with the set of $x$, which is determined by the longest ST message as follows:

$$\begin{aligned} &\min && J_{\text{STslot}} = J_O + J_U \\ &\text{subject to} && X = \{x | x = 2, 4, 6, 8, 10, 12, 14, 16\}. \end{aligned} \tag{26}$$

With the exploration of $J_{\text{STslot}}$ in the design space $X$, the results of (24) and (25) are shown in Fig. 12(a). This denotes the influence of the payload length on the protocol overhead and the unused network resource. As discussed in (2), the protocol-overhead cost is proportional to the number of frames. The longer the payload length, the shorter the protocol overhead. In contrast, a long payload length produces a significant unused network resource. Fig. 12(b) shows the summation of $J_O$ and $J_U$ and the optimal payload length where the minimum cost function $J_{\text{STslot}}$ is 8 B. As shown in Fig. 12(b), the minimum cost of payload length induce a minimum length of ST segment. This means that the result of the optimal ST slot length make the bandwidth maximum. From this process, the 21 ST messages are divided into 23 ST frames. Specifically, two ST messages longer than 8 B are divided into four ST frames. The rest of the ST messages are assigned to the 19 ST frames.

In this paper, we assumed that the ST segment consists of the minimum required number of ST slots. Therefore, the ST segment length is derived from the optimal payload length and the number of ST frames. Consequently, the ST segment length is 621 MT, and the $C_{\text{Comm.cycle}}$ can be modeled by

$$C_{\text{Comm.Cycle}} = 621 + 2 \cdot d + 200. \tag{27}$$

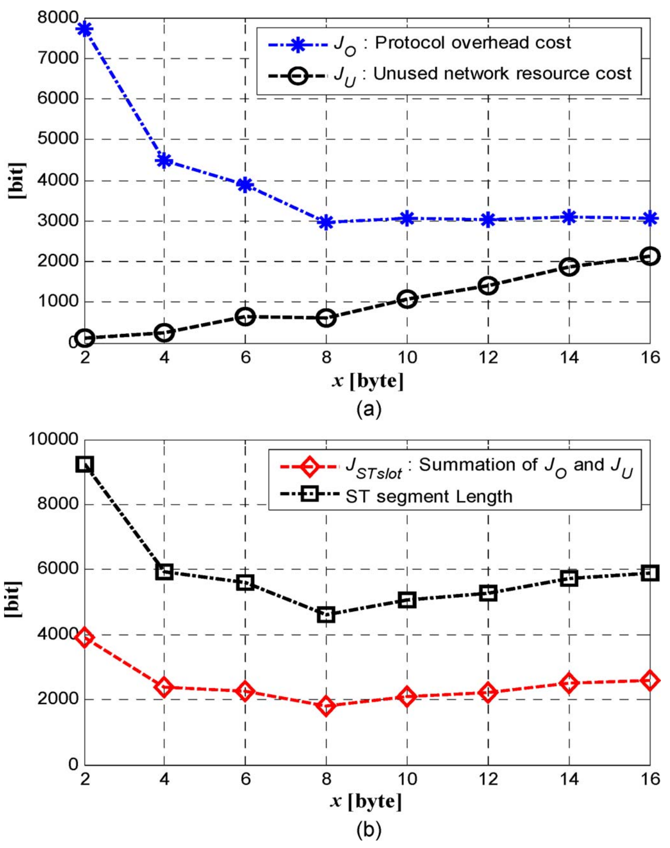


Fig. 12. Case Study: Results of optimal ST slot length. (a) Analysis results of $J_O$ and $J_U$. (b) Analysis results of $J_{\mathrm{STslot}}$ and ST segment length.

Based on these results, the optimal communication-cycle length can be computed. Using (17)–(19), the range of the minislot number is

$$28 \leq d \leq 7980. \tag{28}$$

This defines the design space for the communication-cycle length which is derived using $d$. The WCRT of the ST and DYN frames are analyzed using (14) and (20) with exploration of $d$ from 28 to 7980. In this example, we assumed that the weightings of $J_{\mathrm{ST}}$ and $J_{\mathrm{DYN}}$ are equal (i.e., $\omega_1$ and $\omega_2$ are both 0.5). In the case of different weighting factors, this method produces different results. The costs for the WCRT of ST and DYN frames are shown in Fig. 13(a), and their weighted summation and schedulability evaluation results are shown in Fig. 13(b). Consequently, the optimal communication-cycle length is 989 MT with the optimal value of $d(= 84)$. Fig. 14 shows the designed network parameters.

As shown in Fig. 13(a), the cost for WCRTs of the ST frames is linearly proportional to the communication-cycle length. Therefore, a long cycle length has a damaging effect on WCRT of the ST frames. Otherwise, the WCRT of the DYN frames shows different results. The cost for DYN frame's WCRT is inversely proportional to the cycle length until the critical point, as shown in Fig. 13(b). After that point, the cost remains constant. This property results from the FTDMA principles of the DYN segment.

In order to clarify this feature for the WCRT of the DYN frame, we show an example in Fig. 15. In this example, three different communication-cycle lengths are evaluated with three DYN frames. As shown in Fig. 15, the shortest communication-cycle length results in the longest response time for the DYN frames because the DYN segment is not sufficiently long to transmit all DYN frames in the first communication cycle. In the second case of Fig. 15, the DYN segment is long enough to transfer all the DYN frames in a DYN segment. Therefore, this case shows the shortest possible response times for the DYN frames. However, the long communication-cycle length is not always good for the WCRT of the DYN frames. In the third case of Fig. 15, there is a longer communication cycle than the second case. Although the WCRT of DYN frames have the same results, the WCRT of the ST frames get worse.

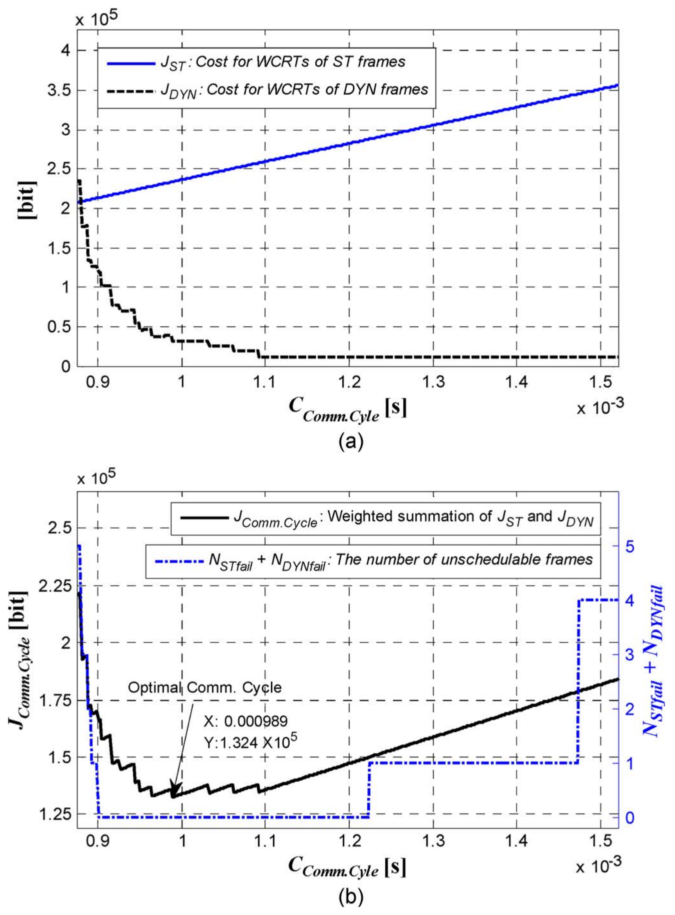


Fig. 13. Case Study: Results of optimal communication-cycle length. (a) Analysis results of $J_{\mathrm{ST}}$ and $J_{\mathrm{DYN}}$. (b) Analysis results of $J_{\mathrm{Comm.Cycle}}$ and $(N_{STfail} + N_{DYNfail})$.

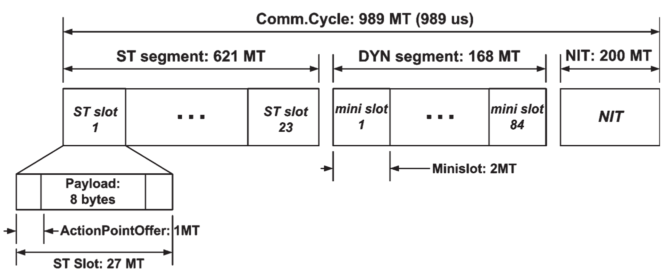


Fig. 14. NPO method results for the example system.

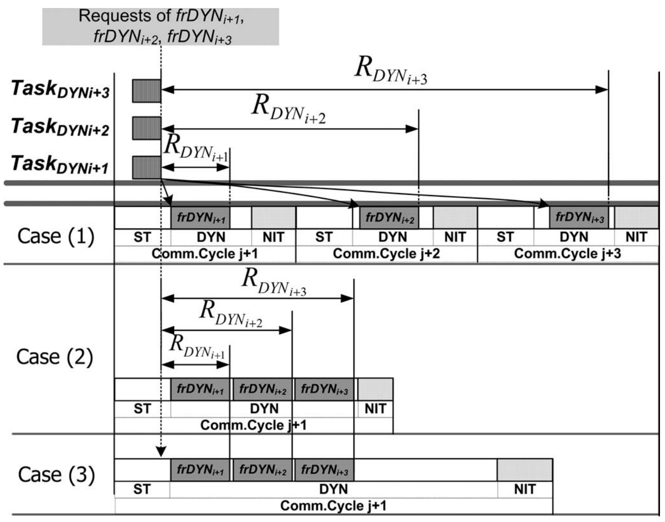


Fig. 15. Properties of DYN frame's response time.

## V. Conclusion

In this paper, we have presented a methodology for designing FlexRay network parameters. Our NPO method is predicated on the principles for optimizing the bandwidth utilization and WCRT of the frames. The method consists of two steps which determine the optimal ST slot length and communication-cycle length. This methodology makes it easy to design FlexRay network. In the first step of the NPO method, the optimal ST slot length can be determined with optimal criteria such as protocol overhead and unused network resource. Using the results from the first step, the second step finds the optimal communication-cycle length with WCRT analysis of ST and DYN frames.

The process of the NPO method is clearly defined in a straightforward manner. Therefore, it can easily be applied to network design applications for automotive, avionic, and industrial fieldbus systems as well.

Ultimately, designing a FlexRay configuration is an emerging field. There exists a wide array of possibilities for future work. Some important issues include supporting signal-based frame construction and multirate communication with cycle-filtering and in-cycle repetition functionalities.

## References

[1] P. Marino, F. Poza, M. A. Dominguez, and S. Otero, "Electronics in automotive engineering: A top–down approach for implementing industrial fieldbus technologies in city buses and coaches," *IEEE Trans. Ind. Electron.*, vol. 56, no. 2, pp. 589–600, Feb. 2009.

[2] R. Junghwan, Y. Maru, and S. Myoungho, "Development of a network-based traction control system, validation of its traction control algorithm and evaluation of its performance using net-hils," *Int. J. Autom. Technol.*, vol. 7, no. 6, pp. 687–695, 2006.

[3] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert, "Trends in automotive communication systems," *Proc. IEEE*, vol. 93, no. 6, pp. 1204–1223, Jun. 2005.

[4] G. Cena and A. Valenzano, "FastCAN: A high-performance enhanced CAN-like network," *IEEE Trans. Ind. Electron.*, vol. 47, no. 4, pp. 951–963, Aug. 2000.

[5] G. Cena and A. Valenzano, "An improved CAN fieldbus for industrial applications," *IEEE Trans. Ind. Electron.*, vol. 44, no. 4, pp. 553–564, Aug. 1997.

[6] G. Cena and A. Valenzano, "Achieving round-robin access in controller area networks," *IEEE Trans. Ind. Electron.*, vol. 49, no. 6, pp. 1202–1213, Dec. 2002.

[7] D. Paret, *Multiplexed Networks for Embedded Systems: CAN, LIN, Flexray, Safe-by-Wire*. New York: Wiley, 2007.

[8] P. Mishra, "Distributed control system development for FlexRay-based systems," presented at the SAE World Congr., Detroit, MI, 2005, Paper 2005-05AE-329.

[9] E. Armengaud, A. Steininger, and M. Horauer, "Automatic parameter identification in FlexRay based automotive communication networks," in *Proc. 11th IEEE Int. Conf. Emerging Technol. Factory Autom.*, Prague, Czech Republic, 2006, pp. 897–904.

[10] T. Pop, P. Pop, P. Eles, Z. Peng, and A. Andrei, "Timing analysis of the FlexRay communication protocol," *Real-Time Syst.*, vol. 39, no. 1–3, pp. 205–235, Aug. 2008.

[11] OSEK/VDX, Time-triggered operating system Ver. 1.0. [Online]. Available: http://portal.osek-vdx.org/files/pdf/specs/ttos10.pdf

[12] OSEK/VDX, OSEK-OS specification Ver. 2.2.3. [Online]. Available: http://portal.osek-vdx.org/files/pdf/specs/os223.pdf

[13] A. Albert, "Comparison of event-triggered and time-triggered concepts with regard to distributed control systems," in *Proc. Embedded World*, 2004, pp. 235–252.

[14] A. Hagiescu, U. Bordoloi, S. Chakraborty, P. Sampath, P. V. V. Ganesan, and S. Ramesh, "Performance analysis of FlexRay-based ECU networks," in *Proc. 44th Annu. Conf. Des. Autom.*, 2007, pp. 284–289.

[15] FlexRayConsortium, FlexRay Protocol Specification 2.1 Rev. A. [Online]. Available: http://www.flexray.com

[16] R. Obermaisser, "Reuse of CAN-based legacy applications in time-triggered architectures," *IEEE Trans. Ind. Informat.*, vol. 2, no. 4, pp. 255–268, Nov. 2006.

[17] A. Ray and Y. Halevi, "Integrated communication and control systems: Part II—Design considerations," *Trans. ASME, J. Dyn. Syst. Meas. Control*, vol. 110, no. 4, pp. 374–381, Dec. 1988.

[18] H. Chih-Lyang, C. Li-Jui, and Y. Yuan-Sheng, "Network-based fuzzy decentralized sliding-mode control for car-like mobile robots," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 574–585, Feb. 2007.

[19] L. Hongbo, C. Mo-Yuen, and S. Zengqi, "EDA-based speed control of a networked DC motor system with time delays and packet losses," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp. 1727–1735, May 2009.

[20] S. Minsuk and S. Myoungho, "Optimal period and priority assignment for a networked control system scheduled by a fixed priority scheduling system," *Int. J. Autom. Technol.*, vol. 8, no. 1, pp. 39–48, 2007.

[21] B. Tavassoli, P. Jabehdar-Maralani, and N. Rezaee, "Tuning of control systems over CSMA networks," *IEEE Trans. Ind. Electron.*, vol. 56, no. 4, pp. 1282–1291, Apr. 2009.

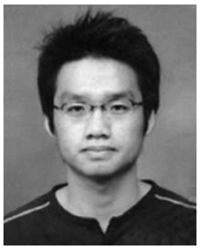
[22] Vector, Davinci Network Designer FlexRay 2.0. [Online]. Available: http://www.vector.com/vi_networkdesigner_flexray_en,,223.html

[23] B. Wittenmark, J. Nilsson, and M. Torngren, "Timing problems in real-time control systems," in *Proc. Amer. Control Conf.*, 1995, pp. 2000–2004.

[24] K. Arzen, A. Cervin, J. Eker, and L. Sha, "An introduction to control and scheduling co-design," in *Proc. 39th IEEE Conf. Decis. Control*, 2000, pp. 4865–4870.

[25] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.

[26] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Berlin, Germany: Springer-Verlag, 1997.



**Inseok Park** (S'09) received the B.S. degree in electronics and computer engineering and the M.S. degree in automotive engineering from Hanyang University, Seoul, Korea, in 2005 and 2007, respectively, where he is currently working toward the Ph.D. degree in the Automotive Control and Electronics Laboratory.

His research activities include design of hard real-time system, distributed control system, and in-vehicle network. He has also done some work on model-based embedded software development for chassis and powertrain control systems.

**Myoungho Sunwoo** (M'81) received B.S. degree in electrical engineering from Hanyang University, Seoul, Korea, in 1979, the M.S. degree in electrical engineering from the University of Texas at Austin, Austin, in 1983, and the Ph.D. degree in system engineering from Oakland University, Rochester Hills, MI, in 1990.

He was a Researcher with GM Research Laboratories, Warren, MI, for about nine years. He has been a Professor with the Department of Automotive Engineering, Hanyang University, since 1993. His research interests are design and development of adaptive control algorithms and distributed real-time control systems for various automotive systems.

Dr. Sunwoo was the recipient of the Best Scientist/Engineer Award of the month from the Korean Ministry of Science and Technology in 2007, the fifty remarkable research accomplishments among 400 National Research Laboratories in 2008, and the Grand Award of Academic–Industrial Foundation in 2009. He is currently a member of the National Academy Engineering of Korea. He served as President of the Korea Society of Automotive Engineers in 2009.