

Formal Design Process for FlexRay-Based Control Systems with Network Parameter Optimization

Inseok Park, Jeamyoun Young, Jooyoung Ma and Myoungho Sunwoo
Hanyang University

Copyright © 2008 SAE International

ABSTRACT

FlexRay is a deterministic and fault-tolerant in-vehicle network (IVN) protocol. It is expected to become a practical standard for automotive communication systems. According to the FlexRay protocol specifications, there are about 60 configurable parameters which should be determined in the design phases. The parameters increase the complexities of FlexRay-based control system development. In this study, we are suggesting a formal design process for FlexRay-based control systems, which is focused on network parameter optimization. We introduce design phases from functional system models to implementations. These phases present formal ways for task allocation, node assignment, network configuration, and implementations. In the network configuration phase, two FlexRay core parameters are selected to optimize network design. Optimal methods of the core parameters provide concise guide lines for optimal communication cycle length and optimal static slot length. For the purpose of feasibility test, we have applied the proposed design process to an electronic stability control (ESC) system.

1. INTRODUCTION

In the last few decades, as increased the number of Electronic Control Units (ECU) in vehicles, point-to-point links have been replaced with bus-based network systems such as Controller Area Network (CAN)[1] and Local Interconnect Network (LIN)[2,3]. At present, CAN is the dominant in-vehicle network protocol. However, CAN suffers from limitations that rules out this protocol for the use of advanced automotive control systems such as x-by-wire systems. More fault-tolerant and faster communication protocols have been required [4]. As a result, FlexRay protocol was presented by the FlexRay consortium. It is expected that FlexRay will be a practical standard for IVN.

FlexRay is a heterogeneous protocol which provides time division multiple access (TDMA) and flexible time division multiple access (FTDMA) schemes [5] and combines advantages of TDMA and FTDMA. Furthermore, fast communication speed and redundant channels are possible. Therefore, FlexRay can provide features such as determinism, high bandwidth, and fault-tolerance to FlexRay-based control systems.

Unfortunately, it is more complex to develop the FlexRay-based control systems than CAN-based. A FlexRay bus has more configurable parameters which affect the performance of control systems, such as communication cycle length, static slot length, static segment length, and dynamic segment length. These parameters should be configured with considerations of the system performance in the design phase.

In this study, we suggest a formal design process which can provide design rules for optimized network configuration of FlexRay-based control systems. This process presents procedures from functional system model to implementation step. The process includes design rules for dividing a system model into tasks, assigning the tasks to nodes, FlexRay network configuration, and implementation steps. In the network configuration phase, we focused on two core parameters (communication cycle length, static slot length) which are dominantly influential with temporal behaviors of FlexRay bus. Optimal methods based on criteria such as protocol overhead, network utilization and worst-case response time are proposed for determining the core parameters.

This paper is organized as follows. At first, FlexRay protocol basics are described. Then a formal design process for FlexRay-based control systems is suggested. For the purpose of feasibility test, the proposed design process is applied to an Electronic Stability Control (ESC) system [8]. At last, conclusions and possible directions for future work are considered.

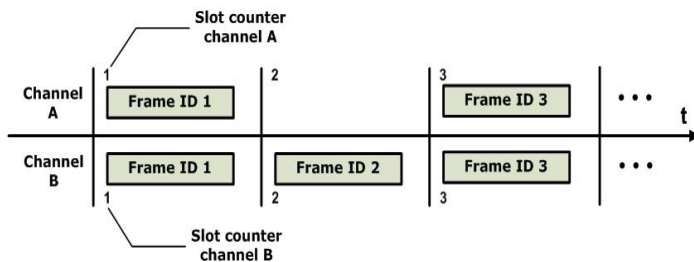


Figure 1. An example of static segment

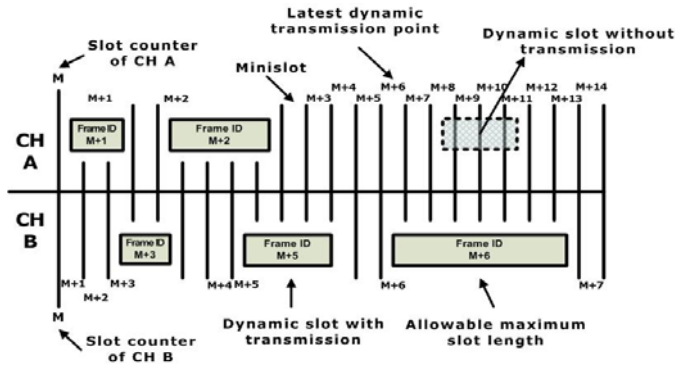


Figure 2. An example of dynamic segment

2. FLEXRAY

2.1 FRAME FORMAT

The frame format of FlexRay is composed of three segments: header, payload, and trailer. The header segment consists of frame indicators, frame ID, payload length, header cyclic redundancy check code (CRC), and cycle counter. In the payload segment, a maximum 254 bytes of data can be transmitted. The trailer segment includes a 24-bit CRC for the frame. This field has the cyclic redundancy check code which is computed over the header segment and the payload segment of the frame.

2.2 TIMING HIERARCHY

A communication cycle consists of a static segment, dynamic segment, symbol window, and network idle time (NIT). The static segment and NIT are necessary elements of a communication cycle and the dynamic segment and symbol window are optional. The static segment consists of a number of static slots and the dynamic segment is composed of minislots. The static slots and minislots are composed of macroticks (MT), representing a unit of global time.

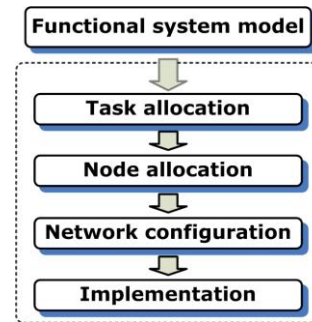


Figure 3. Overall design process

2.3 MEDIA ACCESS CONTROL (MAC)

The MAC of FlexRay protocol is based on the slot counter and frame ID [5]. When a frame ID is identical to a slot counter, the frame can be transmitted to the FlexRay bus.

2.3.1 Static segment

A static segment consists of a number of static slots which have the same length. Within the static segment, a TDMA scheme is used to arbitrate transmissions as shown in Fig. 1.

2.3.2 Dynamic segment

A dynamic segment is an optional element for non-periodic communications. Within the dynamic segment, a dynamic mini-slotting based scheme with flexible time divisions is used for arbitrate transmissions as shown in Fig. 2. The length of a dynamic slot is dependent on a transmitting frame. The dynamic slot consists of multiple minislots if the communication takes place on the channel. If no communication takes place on the channel, the dynamic slot has only one minislot.

3. DESIGN PROCESS FOR FLEXRAY-BASED CONTROL SYSTEMS

In this section, we introduce a design process for FlexRay-based control systems. The process consists of four procedures as shown in Fig. 3. This process starts from a functional system model. The system model is redesigned for a network-based control system during task allocation and node allocation phases [7]. In network configuration phase, configurable parameters of FlexRay are determined by optimization rules. Based on the results of previous phases, FlexRay-based control systems can be implemented.

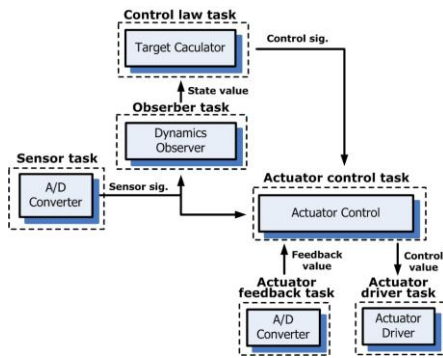


Figure 4. Example of task allocation

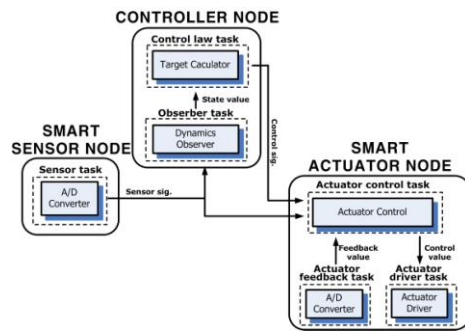


Figure 5. Example of node allocation

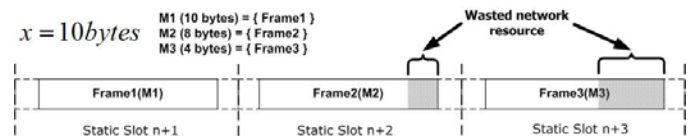
3.1 TASK ALLOCATION

Task allocation is a first phase of the design process. The functional system model should be divided into tasks with regard to the functionality, period, deadline, and dependent relationships. In this phase, the task properties such as priority and worst-case execution time should be defined. Fig. 4 shows an example of the task allocation.

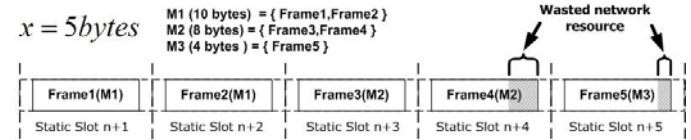
3.2 NODE ALLOCATION

Upon considerations of computing power, available Input/Output(I/O), system memories, dependent relationships of tasks, and physical constraints, the allocated tasks have to be assigned to network nodes.

The tasks of Fig. 4 are allocated to the three nodes as shown in Fig. 5. In this phase, the signals between tasks can be classified into the network signals and node internal signals. In a FlexRay network, the network signals refer to FlexRay frames which have properties such as payload length and period.



(a) Payload length of static frame (10bytes)



(b) Payload length of static frame (5bytes)

Figure 6 Example of optimal static slot length

3.3 NETWORK CONFIGURATION WITH PARAMETER OPTIMIZATION METHOD

In this section, methods for optimizing network configuration parameters, such as static slot length and communication cycle length, are proposed.

3.3.1 Assumptions

In order to simplify the complexity of this problem, some assumptions are defined as follows.

- An application layer is not synchronized to FlexRay network. (i.e. FlexRay network is isolated from the application layer)
- In a static segment, transmission request periods of static frames should be longer than communication cycle.
- In a dynamic segment, only aperiodic tasks [9] request message's transmission. The minimum message request intervals are bigger than communication cycle sufficiently.
- FlexRay network supports single channel.
- A dynamic frame ID should be assigned to a single dynamic message.
- Baudrate: 10Mbps
- Bit-time: 0.1us/bit
- The duration of a MT: 1us
- The duration of a minislot : 2 MT
- The duration of a static slot action point offset: 1 MT
- The duration of a minislot action point offset: 1 MT
- The duration of a NIT segment: 60 MT

3.3.2 Optimal static slot length

In spite of different lengths of static messages, all static slots have a common length. Generally, the static slot length is determined with respect to the longest messages. If there are lots of messages which are smaller than the longest length, most of network resources can be wasted. In order to reduce the waste of network resource, the adequate payload length should be determined by optimization method. Fig. 6 shows an example about this approach. There are three static