Research on Establishing the Model Framework Method for Vehicle Electronic Information System Based on ICD

Ming GAO, Yong LIU and Feng-ying PANG
China North Vehicle Research Institute, Beijing 100072, China

Keywords: Simulink, Model, Interface control document, System function.

Abstract. In the vehicle electronic information system, there is a growing number of data interfaces between devices, and data types are more and more. In the process of device modeling, modifying an interface data involves not only modifying the model associated with it, but also the modification of many communication interfaces. The modification process is complex, resulting in an extension of the modeling cycle. In order to solve this problem, a method was presented in this paper to generate Simulink interface model framework based on ICD. If the ICD needs to be modified, just update the ICD and regenerate interface model framework to complete the modification of all relevant interface models. This method solves the problem of the interface mismatch caused by ICD modification, shortening the modeling period. It is proved that this method is feasible and effective, and can greatly accelerate the establishment of the model.

Introduction

With the development of science and technology, the function of vehicle electronic information system is increasing gradually, and the system composition is becoming more and more complex [1]. This poses a great challenge to the reliability of the system. How to design a highly reliable information system is particularly important. At the beginning of the design, the information system is simulated and verified, and the design defects can be found in time. Modifying and simulating iterative verification, we can repair the design defects in time and improve the design of vehicle electronic information system, improving equipment reliability and quality.

In the simulation process, the establishment of the model is a very important step, which is related to the operability of the test and the credibility of the test data. Vehicle electronic information system involves many logical algorithms, and Simulink provides a number of professional algorithms and model libraries. Therefore, using Simulink as modeling tool can not only meet the demand, but also speed up the establishment of the model.

The interface signal data of each subsystem or electronic equipment of vehicle electronic information system is huge. At present, the management of the interface data is realized by the Interface Control Document (ICD). ICD defines the function and technical characteristics of the interface signals between subsystems or electronic devices of the vehicle electronic information system[2,3].

In the process of modeling, the modification of a single interface data involves not only the modification of its own logical model, but also the modification of several interface models related to it. Therefore, this paper proposes a method to automatically generate Simulink model framework based on ICD. This method maps the ICD data to the Simulink interface model. If the ICD data needs to be modified, the Simulink interface model can be regenerated only according to the updated ICD data. Then, on the basis of the interface model, the internal functional logic of each device is built, and the model is completed. This method can standardize the interface model and avoid the mismatch of the whole model interface data due to the modification of some interface data, which causes that the model cannot run normally and affects the simulation test.

Establishment of Simulink Model Framework

The establishment of Simulink model framework is based on ICD, and the ICD data is expressed in
the form of S-function of Simulink. Each data block of ICD corresponds to the two S-functions, including a packing function and an unpacking function. Then each S-function is encapsulated into a Simulink RTI (Real-Time Interface) module. Each system logic model establishes the communication relation through the RTI module.

The Introduction of S-function

Simulink can build its own custom function module through the S-function and so on. S-function is the abbreviation of system function, which refers to the description of a module in a non-graphical way. A complete S-function architecture contains all the capabilities needed to describe a dynamic system. Users using the S-function can add their own modules to the Simulink model library, and can also freely choose to use MATLAB, C, C++ and other languages to create their own modules [4].

In the S-function, calling different sub-functions has different meanings. In the simulation initialization stage, the mdlInitialize sub-function is called to define the basic S-function module characteristics, including the sampling time, the continuous state and the initial condition of discrete state, size array and so on. At the beginning of the simulation, the mdlGetTimeOfNextVarHit sub-function is called to calculate the next sampling time and provide the sampling time. And the mdlOutputs sub-function is called to calculate the output of the module. In addition, the mdlUpdate sub-function is used to update the discrete state, the sampling time and the maximum time step, and the mdlDerivatives sub-function is used to calculate the derivative of the continuous state. At the end of the simulation, the mdlTerminate sub-function is called to end the simulation [5]. The calling order of each sub-function is shown in Figure 1.

![Figure 1. The calling order of each sub-function.](image)

The ICD Packing and Unpacking Function

ICD data packing and unpacking functions are written based on the S-function format. The packing function is to package the ICD data that the device needs to send according to the ICD data interface definition and send it to the simulation bus. The unpacking function parses the ICD data received by the device according to the ICD data interface definition. Each data block corresponds to a packing function and a unpacking function, the number of data blocks on the whole system on the corresponding number of packing functions and unpacking functions.

The templates of the packing function and the unpacking function are both S-functions. The meaning of the common sub functions in each function is described in detail below.

1. **typedef struct**

   This function defines the content of each data block of ICD as a structure, and each S-function contains a structure.
Static void mdlInitializeSizes (SimStruct *S)
This function initializes the input port data type, the port frame length and the number of input ports corresponding to the ICD data block.

Static void mdlStart (SimStruct *S)
The function is called before the simulation starts, which is used to initialize the state vector of the S-function.

Static void mdlInitializeSampleTimes (SimStruct *S)
This function is used to initialize the sampling time and set the sampling time to 100ms.

Static void mdlOutputs (SimStruct *S, int_T tid)
In the packing function, the sub-function is used to package the result data according to the ICD data and send the data to the simulation bus. In the unpacking function, this sub-function is used to receive data from the simulation bus, and parse the received data according to ICD data definition.

Static void mdlTerminate (SimStruct *S)
The function is placed at the end of the file and executed at the end of the simulation.

The RTI Module Package File
The RTI module package file can be run in the MATLAB environment. It is mainly used to encapsulate the above packing function and unpacking function, and add them to the Simulink module library. The RTI module encapsulation file consists of two files, and their functions are as follows.

install.m
The first function of the document is to compile the packing function files and the unpacking function files into the MEX files that can be used in MATLAB. The second function is to invoke the generated MEX file to generate the interface model.

slblocks.m
The slblocks is MATLAB's own function, and its function is to add custom modules to the Simulink module library. The role here is to name the interface model formed by the install function and add it to the module library.

The Automatic Generation of Package Files
Because of the huge amount of ICD data and data blocks, and the repeatability of S-function writing, a software is designed to reduce the workload. This software uses ICD data as input and automatically generates the packing function and the unpacking function and encapsulation files by analyzing the content of ICD data block. The generated file runs in MATLAB, and the interface model can be generated.

Test Verification
In order to verify the correctness of the above method, this paper uses the CAN bus as an example to verify. The example establishes two CAN bus nodes, and two devices carry out point to point communication. The validation model is illustrated in Figure 2. The detailed steps of verification are as follows.

1. Firstly, the test item "test" is set up, and the system "SYS_XX" is set up under the project. Two devices "HXJ" and "JSY" are set up in the system, and the input and output ICD data of each device are input in sequence. The device "HXJ" has two outputs and one input, and the device "JSY" has one output and two inputs. The two devices are input and output to each other. The device ICD information is shown in Figure 3. Finally, the established ICD information is exported as an ICD file that can be used by the model framework generation software.
Let the ICD file import the model framework generation software. Then the software can automatically generate the corresponding packing and unpacking functions and package file. The exported files list is shown in Table 1. Beginning with "Pack" is the packing function, and beginning with "Unpack" is the unpack function. Named "install" and "slblock" files are the file packages.

### Table 1. The exported files list.

| File name            | File type | File size |
|----------------------|-----------|-----------|
| install.m            | M-file    | 12 KB     |
| slblocks.m           | M-file    | 1 KB      |
| Pack_CAN_Mess01_01.c | C Source  | 6 KB      |
| Pack_CAN_Mess02_01.c | C Source  | 6 KB      |
| Pack_CAN_Mess02_01_0.c | C Source   | 5 KB      |
| Unpack_CAN_Mess01_01.c | C Source   | 6 KB      |
| Unpack_CAN_Mess02_01.c | C Source   | 6 KB      |
| Unpack_CAN_Mess02_01_0.c | C Source   | 6 KB      |

Run the package file in MATLAB and complete the module establishment. The generated module can be displayed in the Simulink module library. Figure 4 is a list display of the module framework in the Simulink module library. Figure 5 is the interface model framework of “HXJ” device, and figure 6 is the interface model framework of “JSY” device.
Through the verification of the above examples, the generated device model framework can fully reflect the ICD information, and can build the internal detailed logic in the Simulink environment. This method accelerates the process of model building, and is more simple and convenient. In the design process, the modification of ICD can be more quickly reflected on the simulation model, which facilitates the development of equipment and the realization of simulation.

**Conclusion**

According to the current situation of vehicle electronic information system simulation needs, through the research on the ICD data and S-function, this paper presents a method of establishing simulation model framework based on ICD, and verifies the method. The built simulation model framework can be used in Simulink, which simplifies the design and simulation of the interface model. This method improves the establishment of interface model, and shortens the modeling time. This paper provides a new idea for the establishment of the model in the subsequent development of complex equipment, which is of great significance for shortening the development cycle of equipment and improving the reliability of equipment.

**Reference**

[1] Liu Yong, Mao Ming, Chen Wang. The design theory and method of information system of tanks and armored vehicles [M]. Beijing: The Publishing House of Ordnance Industry, 2017.

[2] Pang Feng-ying, Liu Yong, Huang Min. Research on Simulation Method of Interface Control Document [J]. Vehicle and Power Technology, 2010, (1): 11-14.

[3] Xia Qing-mei, Xu Ya-jun, Xiong Hua-gang. Database Management of Avionics Interface Control Document [J]. Aeronautical Computing Technique, 2001, 31 (3): 39-42.

[4] Zhang De-feng. Modeling and Simulation of MATLAB/Simulink [M]. Beijing: Publishing House of Electronics industry, 2009.

[5] Li Ying. Modeling and Simulation of Simulink dynamic system [M]. Xi’an: Xidian University Press, 2009.