Control Strategy of Knob Operated Shift for Electric Vehicles

Xi Chen[1, a], Hong Fu[1, b], Kun Wang[2, c], Ling Su[2, d] and Jin Wen[2, e]

[1]Chongqing Changan New Energy Automobile Co., Ltd, Chongqing 401120, China;

[2]Chongqing Changan Automobile Co., Ltd, Chongqing 400023, China.

[a]jakechen700@163.com, [b]fhwiwa@163.com, [c]wangkun1@changan.com.cn, [d]suling@changan.com.cn, [e]wenjin@changan.com.cn

Keywords: Electric vehicle; Knob operated shift; Embedded Coder; Automatic code generation.

Abstract: We developed knob operated shift control strategy used on pure electric vehicles by use of *Matlab 2015b* and its Embedded Coder (EC) toolbox based on the V-model development process. Firstly, the functional requirements of knob operated shift strategy were analyzed, a simulation system based on finite-state machine was established in *Simulink/Stateflow*, the input and output signals, and the parameters of the model were scaled. Secondly, C code were automatic generated by use of EC toolbox, test cases were compiled with *Excel*, model in loop (MIL) and software in loop (SIL) auto test tool were used and the model was tested and modified. Test results indicate that, under different conditions, the dynamic response of knob operated shift function meet the design requirements.

1. Introduction

At present, in the development of electronic control system for automobile and its parts, V mode is the mainstream method of electronic control system development because of its short development cycle, high reliability [1, 2]. V model mainly includes requirement analysis and off-line simulation, rapid control prototype design, code generation and MIL/SIL tests, hardware in loop (HIL) tests and on-line calibration, these five stages [3]. Compared with the traditional design method, each stage based on V model is linked together, any errors can be resolved on the same platform, the code can be automatically generated by code generation tool such as RTI and EC, manpower and time can be saved, developers can focus on Algorithms and modeling efforts [4].

In this paper, the V model is adopted, and the knob shift control strategy of pure electric vehicle is studied based on *Matlab 2015b* and its EC toolbox.

2. Knob shift function requirements

Different with the traditional internal combustion engine, motor can output larger torque in low or zero speed, which can cover a sufficiently wide speed range with a fixed speed ratio. So most pure electric vehicles do not need transmissions, and their running needs can be met only by matching the motor with single speed reducers[5]. A new electric vehicle developed by the company,and the knob operated shift function is adopted. The gear shift knob, from left to right in the order of R, N, D, E four gears, The shifter outputs physical gear signals through four hard lines, the output logic of physical gear signals are as shown in Fig. 1.

In the knob operated shift function, there are three kinds of gear signals, that is, physical gear signals (hard line signals produced by the rotating knob of the driver), system gear signals (according to the current physical gear signals, vehicle speed, brake pedal position and other signals, VCU determines the system gear output signals and sends them to the torque interpretation module), the instrument gear signals (VCU sends the signals to the central console of the vehicle, which serve as a

DOI: 10.35532/JCES.V1.015

display and warning function). There are three kinds of gear faults, the physical gears are not detected, multiple physical gears are detected, the physical gear signals are inconsistent with the system gear signals.

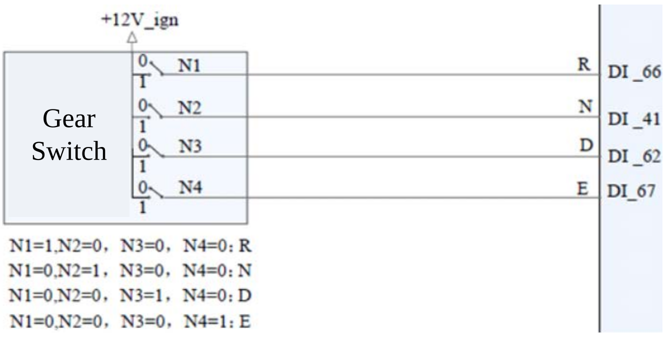

Fig. 1 Logic of the physical gear output signal

According to the function demand of the knob shift, it is necessary to judge whether the vehicle condition is in the running mode before the gear shift is decided. According to the requirements of the power system, if three conditions are met at the same time meet as follows, the vehicle is in the running mode:(1) KeyStart signal is set 1, (2) power system high-voltage power on, (3) no detection of charging line connection signal. If any of the following conditions are met, the vehicle will exit the running mode: (1) KeyOn signal is set 0, (2) detection of the charging cable connecting signal, (3) the vehicle appears faults that needs to exit running mode.

In non charging mode, effective processing of shift signals are as follows:(1) Before the vehicle enters the running mode, the system gear default for N, if the physical gear is in N, then the instrument display N, if the physical gear is in R/D/E, the system gear will output N, the instrument display N flash. (2) When the power system of the vehicle is powered on, no matter what the physical gear is, system gear will output N, if any gear faults occur, the diagnosis scheme shall be carried out, if there are no gear faults and the vehicle has entered the running mode, perform shift operation in the following ways: a) the physical gear is in N, system gear determines output N and instrument screen displays N, b) when the physical gear switches from N/D/E to R, the vehicle speed must be less than 5 km/h and the brake pedal must be pressed, otherwise these shift operations are invalid, the system gear determines output N and the instrument screen displays N flash, c) when the physical gear switches from N/R to D/E, the forward speed must be greater than or equal to 5 km/h, or the absolute value of vehicle speed is less than 5 km/h and the brake pedal is pressed, otherwise these shift operations are invalid, the system gear determines output N and the instrument screen displays N flash, d) physical gear D and E can be switched at will, without any conditions, the system gear and instrument gear are also switched. (3) When the system gear is D/E/R, if the vehicle exit the running mode, system gear output N and the instrument screen display N flash. (4) When gear faults occur, the diagnosis scheme shall be carried out [6].

In charging mode, effective processing of shift signals are as follows:(1) If there is no physical gear signal being detected, system gear will output N and the instrument does not show any gear signal. (2) If a physical gear signal is detected, the display and output strategies of the instrument gear and system gear are consistent with those of the vehicle not enter the running mode. (3) If multiple physical gear signals are detected, system gear output N and the instrument screen display N flash.

In other abnormal circumstances, effective processing of shift signals are as follows: if there are no charging cable connecting signal, no physical gear signals, no gear faults, the instrument screen will show the last stop information.

3. Knob shift *Simulink* model implementation

3.1 Build *Simulink* model

According to the function demand of the knob operated shift, the flow chart and judging condition of the system gear are plotted as shown in Fig. 2.

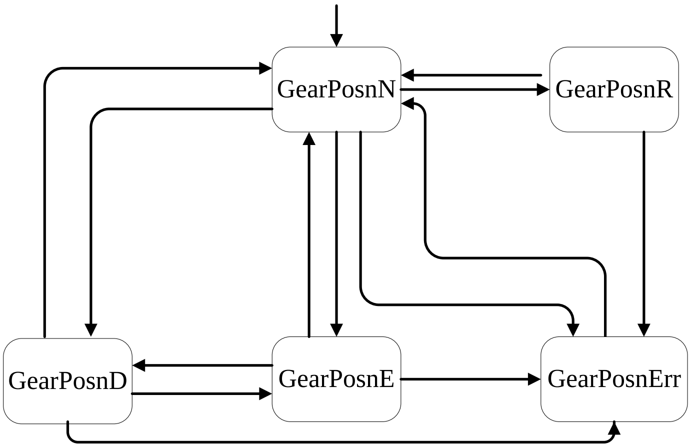

Fig. 2 Schematic diagram of system gear shift

System gear switching conditions are as shown in Table 1, the vehicle speed referred to as VehV, and the physical Gear R/N/D/E are referred to as ShiftLvrPosnR/N/D/E, respectively.

Table 1 System gear shift conditions

State shift	Conditions
GearPosnN to GearPosnD	ShiftLvrPosnR = 0, ShiftLvrPosnN = 0, ShiftLvrPosnD = 1, ShiftLvrPosnE = 0, no gear faults and the vehicle is in the running mode, VehV > 5, or -5 ⩽ VehV ⩽ 5 and the brake pedal is pressed.
GearPosnD to GearPosnN	(1) ShiftLvrPosnN = 1 or ShiftLvrPosnR = 1, ShiftLvrPosnD = 0, ShiftLvrPosnE = 0, no gear faults and the vehicle is in the running mode. (2) The vehicle is out of the running mode.
GearPosnD to GearPosnE	ShiftLvrPosnN = 0, ShiftLvrPosnD = 0, ShiftLvrPosnR = 0, ShiftLvrPosnE = 1, no gear faults.
GearPosnE to GearPosnD	ShiftLvrPosnN = 0, ShiftLvrPosnD = 1, ShiftLvrPosnR = 0, ShiftLvrPosnE = 0, no gear faults.
GearPosnE to GearPosnN	(1) ShiftLvrPosnN = 1 or ShiftLvrPosnR = 1, ShiftLvrPosnE = 0, no gear faults and the vehicle is in the running mode. (2) The vehicle is out of the running mode.
GearPosnN to GearPosnE	VehV > 5, or -5 ⩽ VehV ⩽ 5 and the brake pedal is pressed, ShiftLvrPosnN = 0, ShiftLvrPosnD = 0, ShiftLvrPosnR = 0, ShiftLvrPosnE = 1, no gear faults and the vehicle is in the running mode.
GearPosn(E/D/N/R) to GearPosnErr	Gear faults.
GearPosnErr to GearPosnN	No gear faults, ShiftLvrPosnN = 1, ShiftLvrPosnD = 0, ShiftLvrPosnR = 0, ShiftLvrPosnE = 0.
GearPosnN to GearPosnR	VehV ⩽ 5, the brake pedal is pressed, ShiftLvrPosnN = 0, ShiftLvrPosnD = 0, ShiftLvrPosnR = 1, ShiftLvrPosnE = 0, no gear faults and the vehicle is in the running mode .
GearPosnR to GearPosnN	(1) No gear faults, ShiftLvrPosnR = 0 and (ShiftLvrPosnD = 1 or ShiftLvrPosnN = 1 or ShiftLvrPosnE = 1)， the vehicle is in the running mode. (2) The vehicle is out of the running mode.

In Fig. 2, system gear judgment module (SysGear) is built by use of *Stateflow*/*Simulink*. The ShiftLvrPosnR/N/D/E 4 signals, brake pedal signal (BrkPedlFlgPsd), gear fault signs, VehV, running mode sign (PtFlgDrvAllwd), these 10 signals/signs served as the inputs of the *Stateflow*, the system gear output signals (TrsmFlgGearR/N/D/E) are the outputs of the *Stateflow*. There are 5 states in the *Stateflow*, GearPosnN, GearPosnD, GearPosnE, GearPosnR and GearPosnErr. In GearPosnN and

GearPosnErr these two states, TrsmFlgGearN is set to 1, in GearPosnD/E/R states, TrsmFlgGearD/E/R is set to 1, respectively.

In fault diagnosis module (GearFltDiag), there are ShiftLvrPosnR/N/D/E,TrsmFlgGearR/N/D/E, KeyOn and the charging mode signal (BCUChrgMod) which act as the inputs. When the ShiftLvrPosnR/N/D/E are set to 0 and lasts more than 1 s, the fault diagnosis module judges the fault of the non physical gear signals (GearDfctErr = 1); when more than two physical gear signals are set at 1 and lasts more than 1s, then fault diagnosis module judges the fault of the multiple physical gear signals (GearCoftErr = 1); when the physical gear signal and the system gear signal are not consistent and lasts more than 0.1s, then fault diagnosis module judges the fault of the gear signals do not match (GearDifErr = 1).

ShiftLvrPosnR/N/D/E, TrsmFlgGearR/N/D/E, GearDfctErr, GearCoftErr, GearDifErr, KeyOn and BCUChrgMod are used as the inputs of the instrument gear module (IPGear), and the logic judgment is carried out as follows:(1) when there is no gear fault, the instrument gear shows the current system gear. (2) when gear fault occurs, the instrument shows the N flash or the system gear at the previous moment. The output signal of the instrument gear module (VCUGearPosn), which set to 0, the instrument screen does not show anything; set to 2/3/4/5, the instrument screen correspond to display R/N/D/E, respectively; set to 0 and 3, periodically, the instrument screen shows N flash. These three modules are shown in Fig. 3.

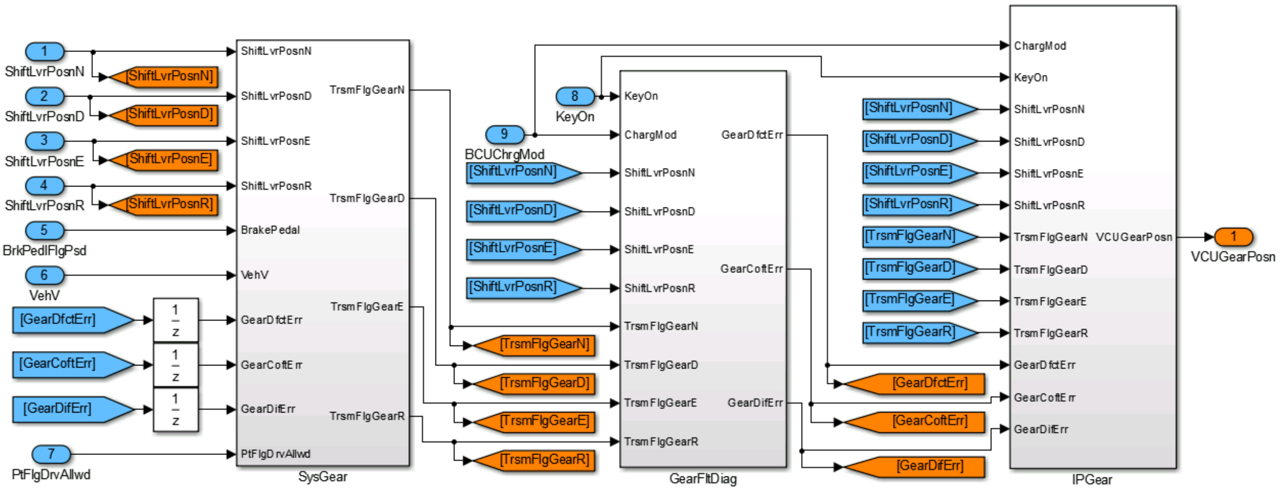

Fig. 3 Knob operated shift model

After the above three modules have been built up, they are encapsulated and built into a whole vehicle control strategy model with other modules.

3.2 Calibration of signals and parameters

After the completion of the model, it is necessary to scale the input and output signals, display signals, intermediate signals, constants and parameters. The scaling of signals and parameters is accomplished by using *Excel* tools. The names, storage categories, maximum and minimum values, data types and units of signals and parameters are filled into the form. The inputs are calibrated as "*ImportedExtern*", outputs and display signals are calibrated as "*ExportedGlobal*", intermediate signals and the constants are calibrated as "*Auto*", and the parameters are calibrated as "*Volatile*". The data types of the input and output signals refer to the CAN protocol, and select the appropriate range and precision to reduce the memory occupied by the hardware .

Since the scaling is completed, the *Excel* which contains signals and parameters is converted into M files using the script file, run the M file, the signals and parameters are loaded into the *Matlab* Workspace, call for MIL/SIL and code generation.

4. Code automatic generation and the MIL/SIL test

4.1 Automatic generation of code based on EC

Before the MIL/SIL test, code generation using the EC toolkit is needed. EC supports more *Simulink* modules and can easily implement complex modeling work. EC model is based on the *Simulink* development environment, which its complete tool chain system can be supported, such as automation test, standard automation verification and automatic report generation tools.

Parameters of the model must be set up, in the *Matlab* "*Model Configuration parameters*" interface, Slover type is set to discrete, fixed step is 0.01 s; select "*ert.tlc-Embedded Coder*" in the Code Generation tab, which is use EC to generate code, select the C language in the target language, in the "*Hardware Implementation*", if the code is used for SIL test, then the hardware platform select " *inter* " , and if it is used for the VCU, select the chip that matches the VCU[7].

4.2 MIL/SIL test process

When the model is built and the signals and parameters are scaled, MIL and SIL tests are needed to detect logic errors, calibration errors, overflows and code generation problems in time [8]. Check the allocation and use of VCU hardware resources to avoid development budget overruns and periodic timeouts in late stage of software development due to the changes in requirements [9].

The MIL tests processes are as follows:(1) Define the test object and establish the test environment. Put the model being test into Model Reference. Install the self-developed automated test tool "*tctool*" and click the "*AutoCreateFromWorkspace*" on the *Matlab* to generate the peripheral interfaces of the inputs and the Scope of the outputs (2) Write test cases, the timing of all inputs should be determined, and all possible working conditions are to be covered by the test cases. Test cases mainly include input signal names, sampling time, signal values, output signals, expected output values . (3) Load the model and test cases with *tctool* and check the results after run. (4) If the MIL tests are passed, then carry out the SIL tests and if not, analyze and locate the causes, modify the model or test cases or test environment, through until all MIL tests are passed [10].

The SIL test processes are as follows:(1) Switch the *tctool* to the SIL test mode, load the test cases and run. (2) Check the results, if all SIL tests are passed, the next step is to carry out HIL tests and if not, analyze and locate the causes, modify the model or the compiling mode of the code.

5. Test result analysis of knob shift

According to the running conditions of the vehicle, four kinds of test cases are designed, each of the first three corresponds to one kind of gear fault, and the fourth one corresponds to no gear fault condition, respectively. The time in the test cases is strictly increasing, and the time point is strictly corresponding to the input and output values. For example, in the 5.8 s of the fourth test case, the VehV is less than 5 km/h and the brake pedal is pressed, the physical gear is shifted from N to D, as it is shown in Table 2.

Table 2 Gear display test case

The sampling points of the input signals are(/s): [0 2 5.0 5.8 6.0 6.5 7.0 7.5]	
Input signals	values
BrkPedlFlgPsd	[0 0 0 1 0 1 1 1]
ShiftLvrPosnR	[0 0 0 0 0 1 0 0]
ShiftLvrPosnN	[1 0 1 0 0 0 0 0]
ShiftLvrPosnD	[0 0 0 1 0 0 1 0]
ShiftLvrPosnE	[0 1 0 0 1 0 0 1]
VehV	[0 4 4 4 4 4 4 4]
PtFlgDrvAllwd	[1 1 1 1 1 1 1 1]
KeyOn	[1 1 1 1 1 1 1 1]
The output signal sampling points are(/s): [0 2 2.09 2.58 3.08 3.58 4.08 4.58 5.0 5.8 6.0 6.5 7.0 7.5]	
Simulation time：9 s	
Output signals	Desired output values
VCUGearPosn	[3 3 0 3 0 3 0 3 3 5 4 2 5 4]

After the *tctool* runs, compare the expectations and results of the MIL test with the Fig. 4. System gear signals, shown in Fig. 5.

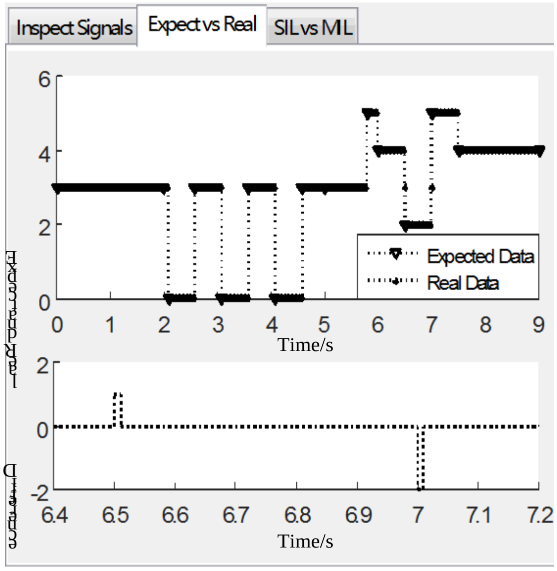

Fig. 4 MIL test expectations and results

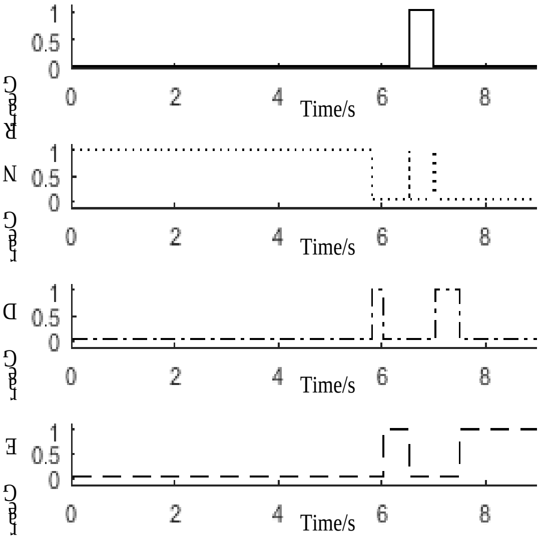

Fig. 5 System gear signal

In Fig. 4, the expectation of the MIL test is inconsistent with the result, with the difference of 1 and -2 at the 6.50-6.51 s and 7.0-7.01 s of VCUGearPosn, respectively. This is because in the 6.5 s, physical gear switches from E to R, simulates the driver rotate the knob too fast, and the knob selector has no time to set the ShiftLvrPosnD and ShiftLvrPosnN at 1 in turn, so that, the ShiftLvrPosnR is set at 1, directly, at the same time, in *Stateflow*, GearPosnE skips GearPosnD and switches directly to GearPosnN, and then switches from GearPosnN to GearPosnR, the system gear set TrsmFlgGearE/N/R to 1 in turn. Meanwhile, VCUGearPosn is first set to 4, then set to 3 and lasts for 10 ms and finally set to 2. For the same reason, when the physical gear switches from R to D, TrsmFlgGearN is set to to 1 for 10 ms in the 7.00-7.01 s. This kind of difference does not affect the knob shift function, it is no needs to modify the test case or the model.

Switch the auto test tool to the SIL test mode, and run the test, the outputs are shown in Fig. 6. As it is shown in Fig. 6, the MIL and SIL test results are the same.

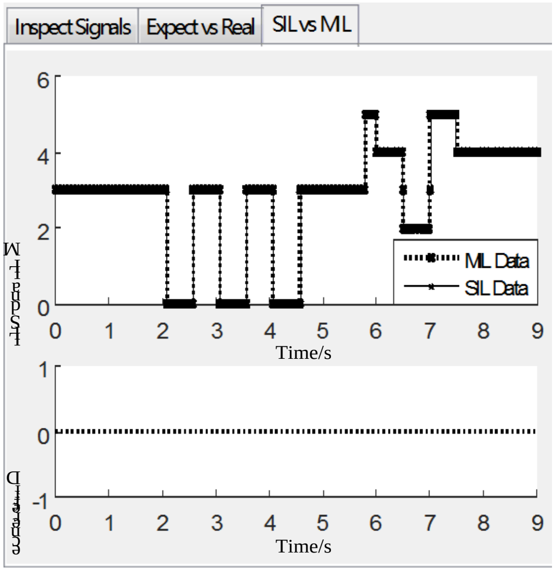

Fig. 6 Comparison of MIL and SIL test results

Fig. 5 and Fig. 6 show that the physical gear switches from N to E does not meet the conditions (-5 ⩽ VehV ⩽ 5, but the brake pedal is not pressed), TrsmFlgGearN is set to 1, VCUGearPosn is alternately set to 3 and 0 (N flash). When the gear shift conditions are satisfied, the knob shifter is rotated to D again, the ShiftLvrPosnD is set to 1, the TrsmFlgGearD is set to 1, and the VCUGearPosn is set to 5, rotate the knob shifter to E R D E in turn, the response of the physical gear, system gear and instrument gear modules are normal. Load the remaining test cases in *tctool* and run in turn, all the tests are passed.

6. Conclusions

In this paper, a model of knob operated shift is built by *Matlab*/*Simulink* for the pure electric vehicle. The MIL/SIL tests of the model are completed by using the automatic test tool. The test results show that the function of the knob operated shift can be correctly corresponding to each working condition, and the design requirements are satisfied. In the future, HIL tests are needed, and since which are passed, generate the code for the VCU, and continue testing in the vehicles.

Acknowledgments

The research work was supported by National Science and Technology support program under Grant No. 2015BAG10B00 (Research and demonstration application of time shared rental mode and support technology for electric vehicles in Mountain Cities).

References

[1]B. Liu, H. Zhang and S. Zhu, "An Incremental V-Model Process for Automotive Development", 2016 23rd Asia-Pacific Software Engineering Conference (APSEC), Hamilton, 2016, pp. 225-232.

[2]Wei Xue-zhe, Dai Hai-feng. Principle, Design and implementation of automotive embedded system [M]. Beijing:Electronics industry Press, 2010:162-179.

[3]Zou Hong-ming, Ding Neng-gen, Wang Neng-da, et al. Research on the RCP and HILS for ABS V-cycle Development [J]. Automotive engineering, 2009, 31(4):358-361.

[4]Yang Xiang-zhong, An Jin-wen, Cui Wen-ge. The research on Application of Embedded Auto Code Generation [J]. Journal of Projectiles, Rockets, Missiles and Guidance, 2008, 28 (3):250-257.

[5] Luo Guang-tao. Research on the Automobile Transmission Integral Performance Test Analysis System [J].Drive system technique, 2008, 22(2):46-48.

[6]Chen Yan-ping, Wan Lei, Lu Jian, et al. Knob operated shift control mechanism and its safe shift method thereof [P].Beijing: CN106369145A, 2017-02-01.

[7]Zhu Jing, Zhang Xing-long. Developing Software of LNG Ship-to-ship Bunkering Control with Simulink/Stateflow [J]. Journal of Shanghai ship and shipping research institute, 2016, 38(1):74-82.

[8]Sun Li-ning, Mu Chun-yang, Du Zhi-jiang, et al. VCU development for 2-axes parallel hybrid bus with V+v development model [J]. Journal of Jilin University (Engineering and Technology Edition), 2009.39(4):1013-1017.

[9]Zhao Yan-bin, Zhong Zai-min. Development on automotive electronic real time control software based on auto-code generation technology [J] Computer Aided Engineering, 2008, 17(3):37-39.

[10]Wang Wei, Fu Hong, Li Yuan-fang. The Application of Charge Lock in Electric Vehicles Based On MIL/SIL Test [J]. AUTO SCI-TECH, 2016(5):75-78.