



Research and Realization of Vehicle Controller Algorithm Based on MotoHawk

Liu Xuejing¹ and Li Yuanyuan^{1*}

¹*School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China.*

Authors' contributions

This work was carried out in collaboration between both authors. Author LX designed the study, performed the statistical analysis, wrote the control algorithm, and wrote the first draft of the manuscript and managed literature searches. Author LY managed the analyses of the study and literature searches. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2015/19365

Editor(s):

(1) José M. Garrido-Balsells, Department of Communications Engineering, University of Malaga, Spain.

Reviewers:

(1) Anonymous, Boumerdes University, Algeria.
(2) Rajinder Tiwari, Electrical & Electronics Engg, Amity University, India.
Complete Peer review History: <http://sciencedomain.org/review-history/10044>

Method Article

Received 5th June 2015
Accepted 23rd June 2015
Published 6th July 2015

ABSTRACT

The development of automotive embedded systems used the "V" font development process and model based development mode, in the development process, CAN bus development and test platform usually was used to develop and test vehicle controller algorithm. According to the above design requirements, the underlying MotoHawk control system is configured using Matlab/Simulink module, the overall structure of vehicle controller algorithm and specific control module is designed and the program is compiled and tested. Through the MotoTune tool for calibration of each output, the control algorithm program is more accurate. The test results show that the control algorithm has good portability, flexibility, control and high accuracy to provide theoretical guidance and strategies for tracking for independent research and development of vehicle controller algorithm.

Keywords: MotoHawk; vehicle controller; calibration; control algorithm.

1. INTRODUCTION

Currently the domestic ECU development process is mainly based on C language, but because of the rapid prototyping based MotoHawk technology has matured and become an international mainstream and itself has a clearly structured design, graphic language, and good code portability advantages better simulation environment [1], which is increasingly being accepted domestic engineer. Automatic code generation tools continue to mature, which makes the design of the prototype and the product software can be well combined. Ugly as it was, the development of automotive control algorithm is still a need for repeated testing and verification work.

In this paper, the development and testing platform of CAN bus is the object of study. MotoHawk rapid prototyping platform was adopted through building simulation and control logic to achieve the design and test the vehicle control system [2,3]. The whole process based on Woodward's D2P MotoHawk software, via Matlab/Simulink to complete [4]. Using rapid prototyping methods can quickly and easily implement control algorithms in real-time hardware platform, which makes early mistakes and deficiencies in the design of the control algorithm will be able to be found and solved in time. Meanwhile, Woodward's MotoHawk development platform based on product-level ECU, its code maturity and applications have access to the product of the requirements, so greatly reducing the cost from development to production cycle and reduce the development process.

2. OVERALL SYSTEM FRAME CHART

Through analysis of the design requirements, the system diagram of the vehicle controller is listed, as shown in Fig. 1. Automotive electronic control system generally adopts the control module, engine control system controls the spark plug, electronic solar term door, fuel injector and accelerator pedal as the execution component [5]. Transmission control system is responsible for the car stalls and the vehicle speed control. Vehicle control system control instrument display, mode switching, fuel, and battery voltage input and output quantity, also doubles as a gateway function. This paper established between EMS, VCU, TCU three control unit communication through the CAN2 interface, VCU controller is responsible for receiving and transmitting data,

real-time acquisition of key switch signal, mode switch signal, the accelerator pedal signal sensor signal, according to the instrument, electronic solar term valve and the spark plug. The information returned by the CAN network, control and management in accordance with the established control strategy. In the framework of the system, the maturity and stability of the software and hardware of the controller and control algorithms of rationality is to ensure the vehicle safety, power and economic premise.

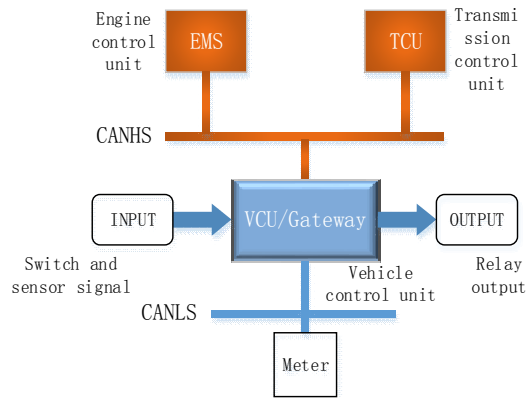


Fig. 1. System diagram of vehicle controller

3. INTRODUCTION OF SYSTEM HARDWARE AND SOFTWARE

The general control module of the GCM-0563-048 series is the new product of the MotoHawk Woodward control solution. The GCM-0563-048 series controller is based on Freescale MPC555 series micro processor, can support the complex control strategy. Using the CAN2.0B protocol, it has two different frame formats, 11 frame identifier containing called standard frame, which contain 29 frame identifier called an extended frame. The control core operating system has the function of generating MotoHawk code, and the MotoHawk development kit can realize the rapid development of complex control system. Flash version of the controller is commonly used for the product, the calibration version of the controller is only for prototyping/development, this paper uses the calibrated version of the controller.

MotoHawk is the external expansion module installed in the MATLAB/Simulink module. The module is the same as the module in Simulink, which will directly drag the module to the working window to build the model [6,7]. MotoHawk enables the Simulink model to run in the MotoTron module. When the task scheduling,

creating a quantitative, the most important is the single step process, you can access the input and output module through the MotoHawk module. MotoHawk extends the Simulink module and the embedded real-time working space, easy to generate the interface between the program code and the module. The goal of the MotoHawk is to focus on solving the problem of control algorithms rather than solving programming problems. The embedded programming is a complicated problem, and the MotoHawk platform makes the problem simpler, but the algorithm control of the engine or the vehicle is still complex and requires in-depth study of the part.

4. ESTABLISHMENT OF VEHICLE CONTROL ALGORITHM

The controller system is composed of the bottom layer and application layer. Bottom is mainly responsible for the configuration of the system memory resources available, compiler parameters setting, configuration, communication protocol, the application layer is mainly control logic and arithmetic, the underlying operating system configuration as shown in Fig. 2.

Specifically, it consists of the following composition: controller and types, memory allocation, relay main control, time trigger, CAN communication baud rate and compiler definition. The module Simulink library MotoHawk drag to work under the window, while the module

parameters are set, you can build the underlying operating system.

This paper set the CAN1 and CAN2 port baud rate is 250 Kbps, each time to send and receive data frame format for 16. Can1 channels are defined as data receiving and sending the channel and frame format for 29 of the extended frame. The can port is responsible for program of brush writing and calibration test of the controller; definition of can2 bus system, responsible for communication between TCU and the VCU and EMS controller to establish a connection.

4.1 Overall Structure of Control Algorithm

MotoHawk function trigger module, its function is to trigger a Simulink Function-Call subsystem, with the built-in framework task event. This block is the root of each MotoHawk task. More than one trigger may be placed in a model hooking off of the same event. In this case, the 'Priority Order' can be used to order these independent triggers. If the priorities are the same, then the order is implementation dependent. Control strategy of vehicle controller is established according to the function of this module [8], as shown in Fig. 3. The control algorithm and the overall structure is divided into the can data transmission, can data receiving, sensor, controller and actuator module unit.

CAN_Rev and CAN_Trans module are responsible for CAN bus data transmitting and

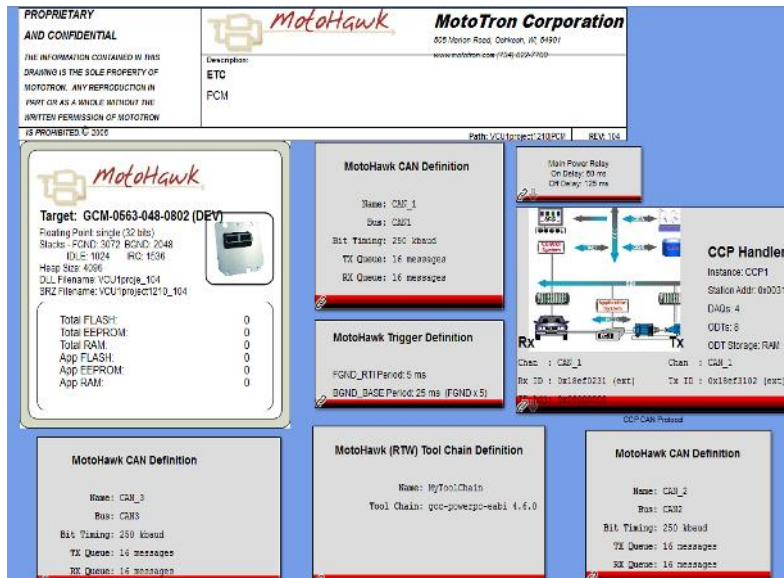


Fig. 2. The underlying operating system configuration

receiving. Due to the vehicle controller acquisition of two other controller of the data frame, so in CAN_Rev module of engine and gearbox controller data were processing and packaging, from the CAN_IN output port and access to the CAN bus. The CAN_Trans module transmits the received data from the CAN bus to the corresponding controller, and then transmits the digital data and analog data to the instrument via the CAN3 channel of the vehicle controller. Sensor module and Actuator module are mainly for the input and output data conversion, the main function of the Controller module is to control the preparation of the control strategy.

4.2 Write Control Algorithm

The main part of the vehicle controller algorithms including Sensor Module, Controller Module and Actuator Module three parts. Here are focus on these three subsystems.

For Sensor module, which mainly contains the analog I/O and digital I/O module [9], it needs to be packaged data processing, analog input is defined here for example to introduce tire pressure, as shown in Fig. 4. The MotoHawk module in the library of the analog input module, convert conversion module, MotoHawk lookup table module and gain module dragged into the window, the connection after the access to the bus and output from ANA_IN port. Add MotoHawk Probe module Ain1 value of tire pressure monitoring, and follow-up based on the value of the calibration parameters MotoHawk Lookup Table module and gain blocks were set up to achieve the precise numerical conversion and control purposes.

For analog input battery voltage programming processing, Fig. 5 and Fig. 4 different point is need in behind the gain module and a subsystem module, the main switch function, from the MotoHawk data definition of "Auto_Test" module to read data values, if the value is greater than zero, the value of the input port 1 output from an output port 1, otherwise the output from an output port 1 port 2 of the input values. When the controller is in the mode of automatic test, the analog quantity control function is in the shield state.

Similar to other analog for the same process, package them into port ANA_IN. For digital input, without conversion, directly packaged Dig_IN port, where no introduction.

For Controller module, the algorithm function as shown in Fig. 6. A left portion of the graph of the function of the program is mainly read "automatic test mode under the state of each quantity, if the state of the controller is in "automatic test" mode, mask changes "Manual mode" analog values. The right part of Fig. 6, for analog data from ANA_IN port for processing, if the detected battery voltage is less than 28V (this voltage value can be modified according to design requirements), the output "battery charging" state. Similar if it detects the amount of fuel is less than 5, then define the output "low fuel quantity" of the state. The processed data output from the analog port 1 to the corresponding actuator. For port 2, the "Auto Test" on the bus digital switching signal conversion and writes it to MotoHawk of Auto_Test module, control strategies and automated test set accordingly.

For Actuator module, shown in Fig. 7, in the digital port 1, for the "battery low" state defined alarm indicator, the Pin feet LSD2. For the "low fuel quantity" of state definition LED Pin feet LSD3, which Subsystems1 module for warning lights flashing cycle and frequency settings, D2out to MotoHawk Absolute Override module, use MotoTune software, the output value can be forced to the new value, ignore the input value, use the module for the convenience of online debugging. Fig. 7 digital port 2, the switches of Auto_Test, brake_state, Danger, AirCondition are accessed to Pin feet LSD1, LSD4, LSD5, LSD7, by LED indicator displays the status of each volume to achieve the corresponding control functions.

When the system model is completed, the amount of input and output pins are connected corresponding to the actual hardware. On the basis of the correctness and reliability of the control system in the physical connection relationship, the ECU code is automatically compiled into the SRZ file by using the GCC compiler. The compiled SRZ file is written in the ECU by the brush write calibration monitoring software MotoTune, to make a preparation for the next controller testing and online debugging.

4.3 The Calibration Procedure of the Control Algorithm on the Equipment

In order to realize the optimization and debug of the parameters, the calibration system should be built on the basis of the platform [10,11]. Calibration system consists of three parts: (1) calibration software MotoTune. (2) interface hardware, due to communication protocol CCP

protocol, so the selection of USB-CAN hardware.(3) Vehicle Control VCU system includes a controller, sliding rheostat and instrumentation, calibration of the system shown in Fig. 8. Calibration is divided into offline and online calibration. Online calibration refers to modify the control parameters in engine running process, the offline calibration is according to the actual device display parameters to the program to optimize the parameters settings, this paper adopts the offline calibration.

Take sliding rheostat as an example, introduces the calibration process. Through the USB-CAN hardware connection between PC and the controller is built, in the MotoTune software build a display window, to show the input quantity sensor. By reading the initial value of the sliding rheostat, conversion value in the program can make a more precise control. First of all, a rheostat sliding to a minimum, this time to observe to each quantity value, after rheostat slip to the highest, then observed the value of each quantity as shown in Table 1.

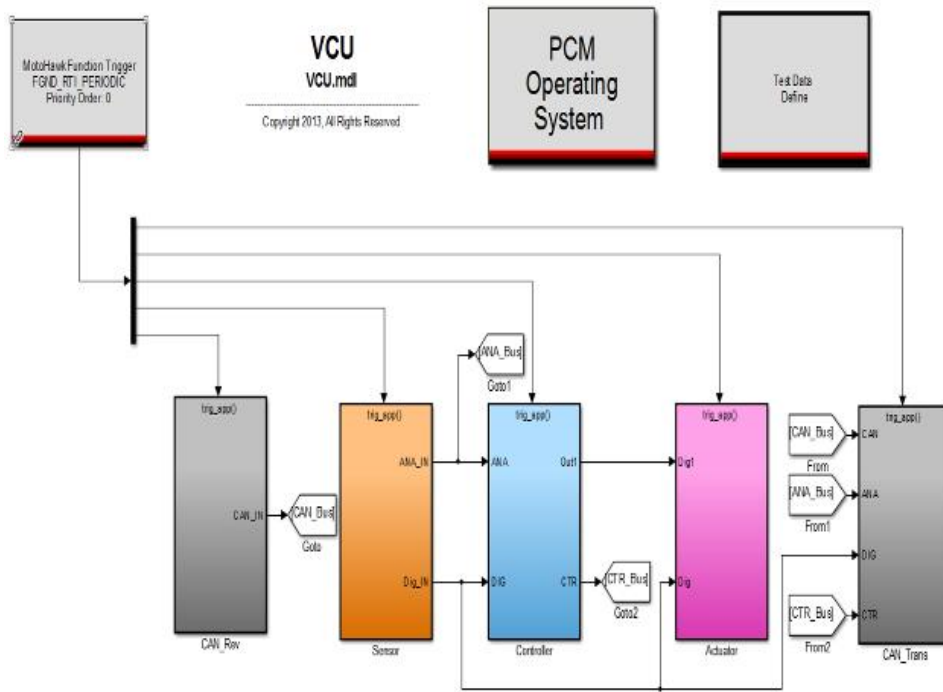


Fig. 3. Control algorithm overall structure diagram

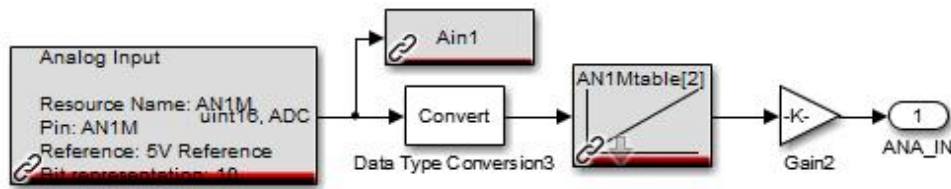


Fig. 4. Analog output process

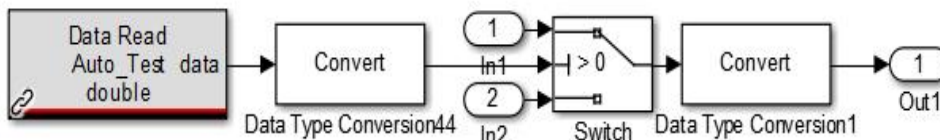


Fig. 5. Input volume switching process

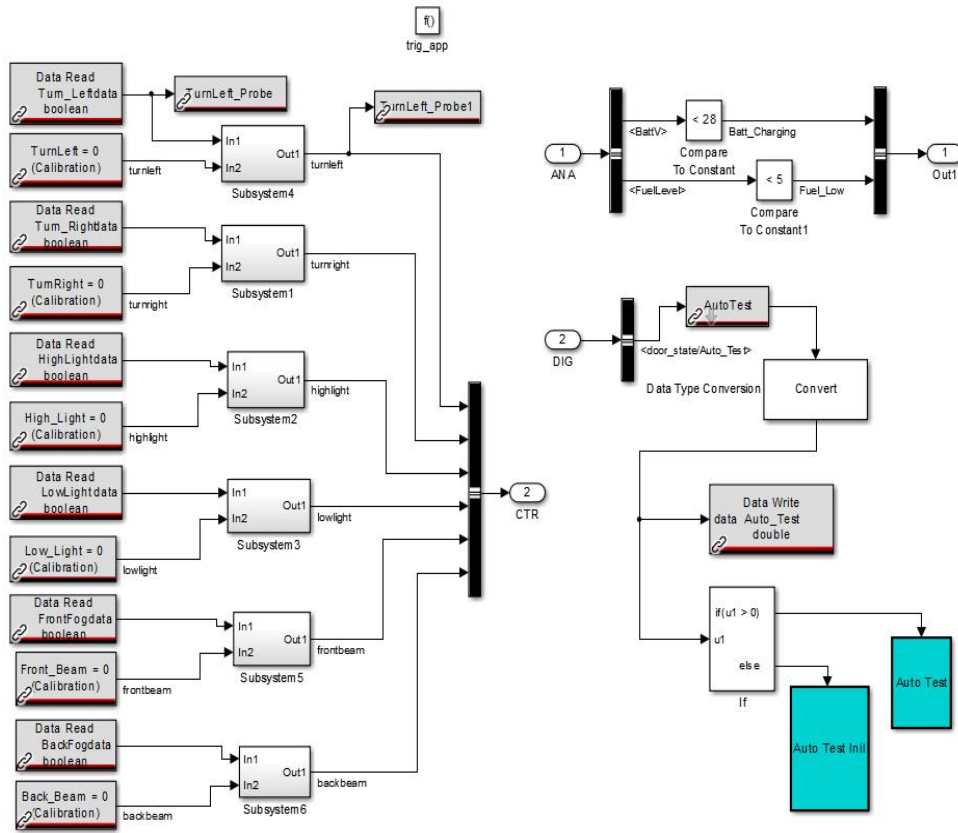


Fig. 6. Algorithm function of controller module

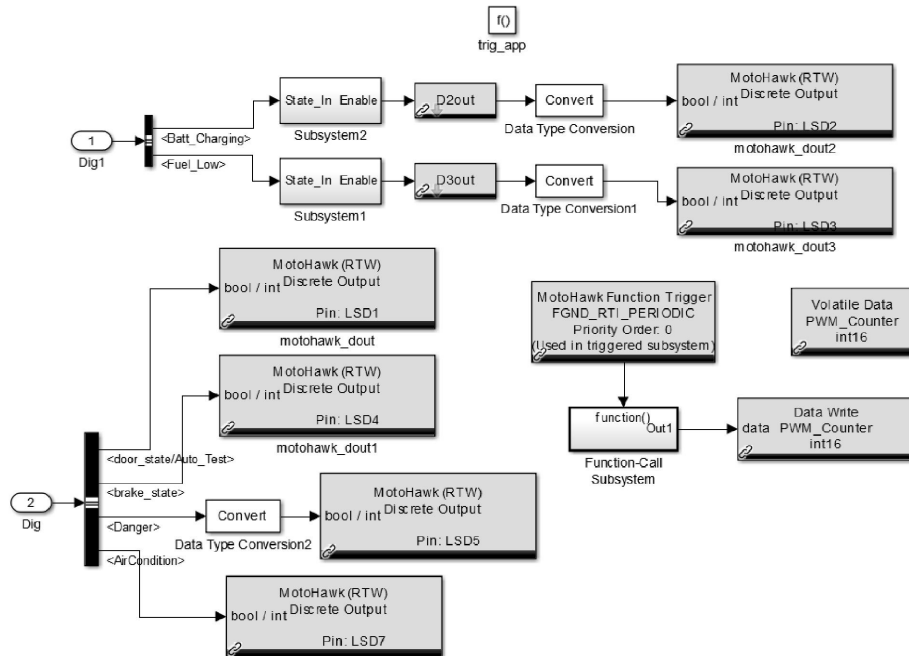


Fig. 7. Algorithm function of actuator module

Table 1. The lower and the upper limit of rheostat

Value of monitoring quantity	The lower limit rheostat	The upper limit rheostat
AN1	95	1023
AN2	22	1023
AN3	0	1023

Analog input, such as sliding rheostat, requires data conversion in order to ensure the accuracy of control [12]. All analog defined here will be converted to between 0-1023. The three analog need conversion values for: in the [95 1023], [22 1023], [0 1023]. Will the values were corrected for [0 1023]. The conversion can be performed in the Lookup Table MotoHawk module in the 3.2 section, and then download the program to the VCU controller for testing.

4.4 Establishment of Automatic Test Control Model

Using the MotoHawk platform, the automatic test model is built to realize the testing process of the system , Can be more flexible to build the corresponding testing process. Before building the test model, a time count function is needed to

define the time node of the whole test procedure.

As is shown in Fig. 9, the main function of this function is: when the controller automatic test switch is turned on, control the output ports of the value starting from a sequence plus 1, when the output value reached 6899, the output value from 1 to output. Due to the periodic system is set to 5 ms, so a single test time as the output of the 34.5 second. The output values are within the defined 1-6899 range. At the same time, the output value is corresponding to the measured variable, as shown in Fig. 10, for example, when the value of u1 is equal to 400, the value of variable "turn left" is set to 1, you can test the left steering lamp, other test items are defined similarly to this. In these ways, the test process can be realized in accordance with the established testing strategy.

5. VERIFICATION OF VEHICLE CONTROLLER ALGORITHM

Vehicle controller algorithm in the controller test defines the automatic test switch, with the combination of manual and automatic test pattern of the control algorithm for verification of the overall, but also to ensure that control the diversity and flexibility.

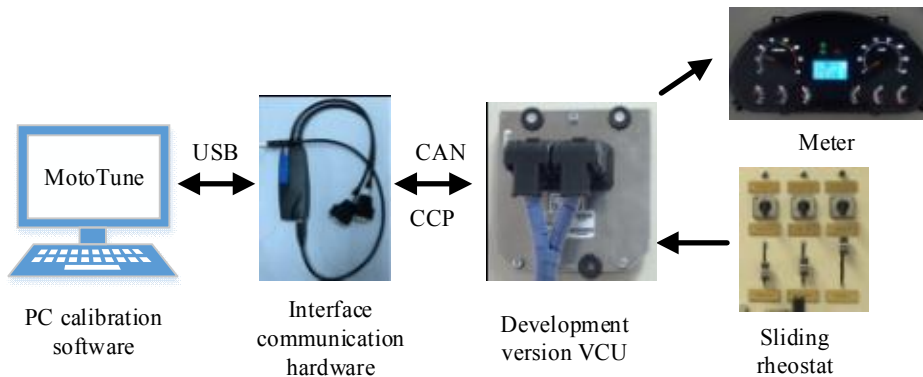


Fig. 8. Composition of calibration system

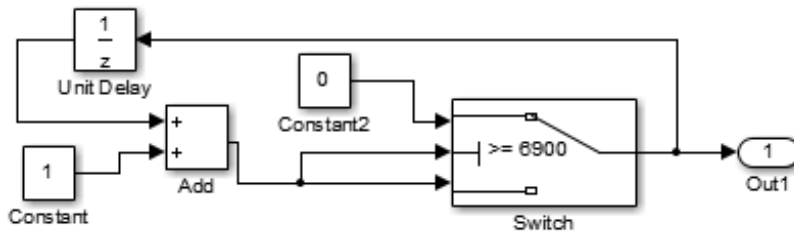


Fig. 9. Time counting function in automatic test

5.1 Manual Mode Test Procedure

The default of the controller is the manual test mode, taking the slip line of the battery voltage and fuel quantity as an example, and introduces the process of the test. When the simulated cell voltage of slide wire rheostat slippery key position of the slide to the middle position (below setting 28 V) and simulating the fuel amount of slide wire rheostat sliding key location sliding to close to the lower limit value (less than 5), the definition of alarm indicator light starts flashing, as shown in Fig. 11, the process better verify the algorithm in Fig. 6 controller module function.

Shift the file to the D file and press the accelerator pedal with the hand to keep the accelerator pedal and hold it. As shown in Fig. 12, we can observe that throttle value on the meter corresponding reached the maximum value, the spark plug ignition frequency to set the maximum value, the electronic throttle opening to maximum. At the same time, the speed is reached the maximum value 140 km/s with desired control strategy.

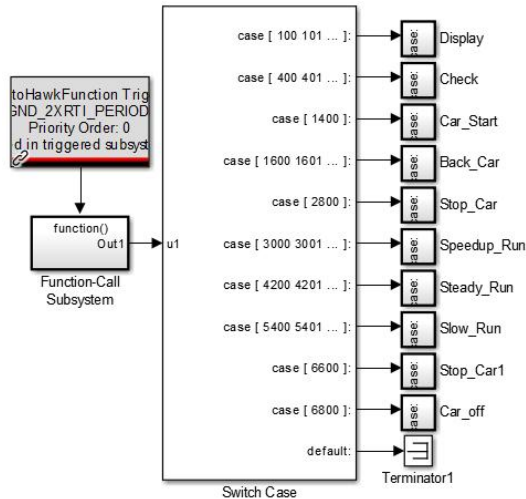


Fig. 10. Definition of test items for each time period

5.2 Automatic Mode Test Procedure

When the automatic test switch is open, we can observe the meter lamp is lit, then turn the test in accordance with the control strategy is the left turn signal, right turn signal, high beam, low beam lights, after the start of the test gear, when the shift position is the R range, the throttle gradually started to increase, the speed limit is set to rise by 40 km/h, then the gear in D range,

with the throttle, the speed increased to the maximum, as shown in Fig. 13. It can well verify the algorithm control model in Fig. 10.



Fig. 11. Verification process of analog quantity control



Fig. 12. Acceleration process under manual mode



Fig. 13. Automatic test process of vehicle controller

6. CONCLUSION

Through the actual hardware test, the control algorithm is verified. Switch and analog quantity control function, accelerator pedal input and electronic throttle gate output function and the instrument parameters display function are verified concretely.

The whole development process of automotive electronic control system mainly concentrated in the application of writing and control algorithm of the test, this paper based on CAN bus development and test platform, the hardware used Woodward company controller, vehicle control system model was constructed using MotoHawk rapid prototype development platform, through calibration to realize the control function, the realization of the vehicle controller algorithm development. Through CAN bus development and test platform can quickly control algorithm to download to the ECU debugging, greatly shorten the development cycle of the system controller and the whole development process for subsequent system further development to provide a reference value, is of great significance.

7. PROJECT SUPPORT

This paper is supported by Shanghai University of Engineering Science connotation construction project "Study on the bus technology of automobile electric control hybrid protocol based on CAN/LIN" (project number: nhky-2013-10).

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Gao Haiyu, Lu Wenchang, Shang Zhe, et al. Rapid prototyping of engine ECU based on MotoTron [J]. Design and Manufacture of Diesel Engine. 2008;15(3):16-20.
2. Yu Panxia, Zhang Xueyong. D2P platform application in automatic transmissions TCU in rapid prototyping [C]. Southwest Automotive Information. 2013;9:139-143.
3. Kong Feng. Research on the development of EMS and the development platform of the rapid prototyping of the electronic control unit of the engine -- UECU's autonomous development [D]. Liaoning: Dalian University of Technology; 2007.
4. Hu Mi, Sun Lei, Ye Erhu, et al. Application of Woodward controller for automatic clutch application [J]. Journal of Chongqing University of Technology: Natural Science. 2014;10:30-34.
5. Zeng Zhiwei. Research on electronic throttle control technology of automotive engine based on CAN bus [D]. Hunan: Hunan University; 2006.
6. Feng Chao. Design and application of electric vehicle simulation model based on Matlab/Simulink [D]. Beijing: University of Chinese Academy of Sciences; 2013.
7. Li Yin, Xue Haibing. Simulink dynamic modeling and simulation (Second Edition) [M]. Xi'an Electronic Science and Technology Press; 2009.
8. Hu Zhonglu, Li Yuelin, et al. The development of transient throttle controller based on mototron development Platform[C]. Digital Manufacturing and Automation (ICDMA). 2013;6:1456-1460.
9. Wang Lianxu. Design and development of pure electric vehicle controller based on Mototron [D]. Jilin: Jilin University; 2013.
10. Richert J, Rolfsmeier A, Leinfellner R. A new calibration system for ECU development[C].SAE Paper. 2003;01-0131.
11. Lu Wenchang, Gao Haiyu, Meng xin, et al. Design and experiment of ETC control system based on MotoTron [J]. Internal combustion engine engineering. 2010; 31(6):59-64.
12. Hosagrahara A, Smith P. Measuring productivity and quality in model-based design [J]. SAE. 2008;4:592-593.

© 2015 Xuejing and Yuanyuan; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:
The peer review history for this paper can be accessed here:
<http://sciencedomain.org/review-history/10044>