

Improvement of Mixed Traffic Scheduling of CAN Network and Simulation Research

Shurui Fan, Fei Li, Jie Li, Hexu Sun

Department of Automation
Hebei University of Technology
Tianjin300130, China

* Corresponding author's Email: fansr@hebut.edu.cn

Abstract: Network scheduling plays an important role in Network Control System (NCS). Focused on problems of the present scheduling of CAN bus, an improved Mixed Traffic Scheduling algorithm (MTS) was proposed on the base of the communication principle of network scheduling. The cut-off time was denoted after introducing the definitions about the scheduling. The NCS model for CAN Network was created then. The simulation platform was created based on MATLAB/True Time toolbox, construed the new schedule affected the control system. The simulation test was conducted and the result proved the validity of the method.

Keywords: CAN Bus; NCS; Network scheduling; MTS; True Time;

1. Introduction

Field buses exhibit bounded message latency, reliability of transfer, efficiency of the protocol and, to a certain extent, preventing a node from monopolizing the network. Among field buses, CAN (Controller Area Network) bus provides advanced built-in features, which makes it suitable for complex real-time applications. Some of these features are priority-based, multiparty bus access control using carrier sense/multiple access with collision avoidance (CSMA/CA), bounded message length, efficient implementation of positive/negative acknowledgement, and automatic fail-silence enforcement with different fault levels.

CAN bus development started in the 1980s by Robert Bosch, aiming to design a bus system for the specific needs of automotive applications. The resulting CAN bus is now standardized as ISO 11898. CAN bus as a real-time control system, with its

advantages of the short data frame structure, the multi-master work mode, high transmission speed, long distance, strong anti-interference capability, bit arbitration, and low prices etc, has been widely applied in all walks of life.^[1,2]

Network Control System (NCS) is a distributed feedback control system, which is composed by real-time network. It means a control system which is formed by the control loop through the serial communication network (such as the CAN network).^[4] The most important feature of a NCS is that it connects cyberspace to physical space, thus enabling execution of several tasks from long distance. In addition, NCS eliminates unnecessary wiring, thus reducing the complexity and the overall cost in designing and implementing the control systems. It can also be easily modified or upgraded by adding sensors, actuators and controllers to it with relatively low cost and no major changes in its structure. Moreover, featuring efficient sharing of

data among its controllers, NCS is able to easily fuse global information to make intelligent decisions over large physical spaces. Advent and development of the Internet combined with the advantages provided by NCS attracted the interest of researchers around the globe. Moreover, a most critical and important issue surrounding the design of distributed NCS with the successively increasing complexity is to meet the requirements on system reliability and dependability, while guaranteeing a high system performance over a wide operating range.^{[7][8]}

In this paper, after the CAN bus communication principle and network scheduling analysis, the scheduling algorithm was introduced, and a new Mixed Traffic Schedule (MTS) was proposed and improved. We used the MATLAB/True Time network control toolbox to build NCS simulation platform, analyzed scheduling and algorithms for network control system performance, and achieved relatively good results.

2. Network scheduling of CAN bus

2.1 Communication mechanism of CAN bus

CAN bus uses a differential serial line architecture with dominant and recessive bits where a dominant bit represents a logical 0 and a recessive bit a logical 1. An idle bus has recessive level. Due to the open collector logic, if one station on the bus sends a dominant bit while another controller sends a recessive bit, the dominant bit wins, i.e. the bus is considered as logical 0. This feature allows using a bit-wise arbitration scheme for the medium access, which is often called CSMA/BA (Carrier Sense Multiple Access with Bitwise Arbitration). Using this mechanism, each station listens to the bus while sending data. If a collision occurs where one station tries to send a recessive bit but receives a dominant bit, so that it will notice that another station is sending simultaneously and stops its own transmission immediately. This makes the arbitration non-destructive, since the station sending the dominant bit can continue to send without any negative effects on the bus while the station sending the recessive bit remains silent from the time when

the collision has occurred. A retransmission of the interrupted frame is triggered automatically.

Any node on CAN bus at any time can reach out to send a message to other nodes on the bus, regardless of master-slave. In case to interfere the information being transmitted on the bus, priority takes place by non-destructive arbitration technique in the event of a conflict. It makes sure that the high-priority information can be transmitted without being affected, while the lower priority one has to wait.

2.2 Schedulability Analysis of CAN bus

Equidistant sampling in the NCS can not be guaranteed, because of the network transmission delay and uncertainty. One of the general requirements is the limited delay, that is, the information must be successfully sent in a limited time. Time characteristics of information transmission must meet the system real-time requirement. Network scheduling focuses on how long a node need to transmit a message, as well as the priority area of information transmission. Whether network traffic can meet the key indicators of process and ensure the transmission or not is called schedule ability analysis. To protect the real-time information transmission of CAN bus, a piece of information must be sent before the next coming out. If the information is not transmitted in time, it will be covered by the next piece of information.

Therefore, under normal circumstances, each message needs a deadline D_m . Then the delay time R_m can be calculated, so the message m scheduling shall meet the condition that $R_m \leq D_m$. A given task m has a worst-case response time R_m , which is defined as the maximum time between the time of being in the transmission queue and the time of being received by the target node. Existing analysis for single processors is able to determine this worst-case response time. For a message of arbitrary S_m , its delay time R_m must meet the following condition:^[5]

$$R_m = t_m + C_m \quad (1)$$

Where t_m is the queuing time, which means the time from queuing to winning the arbitration, C_m is the information transmission time. To ensure the reliability of network scheduling, it is necessary to design the right network scheduling algorithm.

2.3 Basic scheduling of CAN bus

CAN bus uses the data frame identifier to express the source of information and the priority.^[3] CAN does not employ explicit senders or receivers addresses, but uses unique message identifiers to describe the content of the respective CAN frame. The frames are broadcasted on the bus and each station can decide if the message content is relevant by examining the message identifier of received frames. These identifiers have to be assigned statically during the design phase of the bus system to avoid ambiguity in the interpretation of the frame content. A global view of the complete system is needed in this process.

The data frame identifier can be presented both statically and dynamically. At present, it can be divided into static priority scheduling algorithm, such as fixed priority scheduling algorithm (FP), the rate monotonic scheduling algorithm (RMS)^[4,6] and the deadline monotonic scheduling (DMS) and dynamic-priority scheduling algorithm such as the earliest deadline first scheduling (EDF), the shortest idle time scheduling priority (LLF).^[4,6]

Rate monotonic scheduling (RMS) algorithm is a typical static scheduling algorithm, based on the different cycles of each task to allocate priority. The shorter the cycle, the higher priority task takes. And also it is a fixed priority assignment algorithm. The task has been set priority before the allocation, so it will not change throughout the course of time.

EDF (Earliest Deadline First) is a dynamic scheduling algorithm that widely used in processors implementing the task. In order to achieve an effective scheduling, the priority is determined by the deadline of the task, so the tasks can be completed within the required deadline. The basic idea of the EDF scheduling algorithm is based on the absolute deadline to allocate priority dynamically. The higher priority is assigned with the shorter cut-off.^[10,11,12]

2.4 A new mixed traffic schedule

The core idea of this paper is to set the relative deadline information into the identifier, using the EDF message scheduling algorithm for high priority information and the RMS message scheduling algorithm for low priority information. The basic idea of MTS scheduling algorithm is: compare with the "To send message" deadline with the most urgent deadline D0. If the "To send message" deadline and the most urgent deadline D0 belong to the same order of magnitude, send the message with real-time high-speed, otherwise send the message with low-speed. In the MTS scheduling algorithm, the priority assigned to real-time information is higher than that assigned to non-real-time information. (the highest bit of high-speed real-time information identifier 0 can be set to achieve this), making the transmission high-speed and real-time. As for the low-speed real-time and non-real-time information, set the highest bit 1, second peak of 0 and 1 to achieve this, if any 5 bits of the identifiers in the standard frame is used for information filtering, one presents priority bit, then the remaining 5 bits are used to encode the deadline. MTS uses the distributed clock synchronization algorithm, and the time is divided into several regions. The encoding of cut-off date is based on the time domain to which it belongs. In the multi-processor control system composed by more than one electrical network, as a result of increased processor and node number, delay will increase. Common MTS at this time are unsuitable to apply with. In order to enhancing the real-time ability of the network control system, the sampling of MTS algorithm should be improved.

Establish the original sampling period T_m . The sampling cycle T_N is calculated as:

$$T_N = T_m / 2^k, \quad k \geq 0 \quad (2)$$

In each 2^k sampling cycle, only one sampling of data is sent through the network. From the beginning time t_0 in a sampling cycle to the end of the cycle,

$$t = t_0 + hT_N \quad (h = 0, 1, 2 \dots 2^k - 1), \text{ where } t \text{ is sampling}$$

time. A sampling time $t_1 = t_0 + h_1 T_N$ ($h_1 = 0, 1, 2 \dots 2^{k-1}$), sampling data was successfully sent, sampling will not happen any more during the following sampling time. This greatly reduces the sampling time. After this improvement, the network delay still exists, but the cut-off time D_m will greatly reduce.^[5]

If the waiting time for information transmission is over the cut-off time, the queuing data will be discarded and re-collected. That will enhance the real-time ability of a NCS.

3. Structures of network control platform based on CAN bus

The MATLAB/Simulink tool TrueTime makes it possible to simulate the temporal behavior of a multitasking real-time kernel executing controller tasks. The tasks are controlling processes that are modeled as ordinary Simulink blocks. TrueTime also facilitates simulation of simple models of communication networks and their influence on networked control loops. Different scheduling policies may be used, e.g. priority based preemptive scheduling and earliest deadline for the first EDF scheduling.

TrueTime can also be used as an experimental platform for research on dynamic real-time control systems. It is for instance possible to study compensation schemes that adjust the control algorithm based on measurements of actual timing variations. It is also easy to do experiments with more flexible approaches to real-time scheduling of controllers, such as feedback scheduling. Therefore, available CPU or network resources are dynamically distributed according to the current situation load, the performance of the different loops, etc. in the systems.^[9]

Dan Henriksson and Anton Cervin and other scholars developed a network based on the MATLAB simulation toolbox TrueTime,^[11] and

Output position signal $y(k)$ follows the changes of a given location signal $u(k)$. We introduced the generally, integral part of PID digital controller is used to eliminate the static error and to improve accuracy.

constructed through the CAN bus control three DC motor, which make up the simulation system, then analyze the effect of the scheduling algorithm to the control performance, NCS test platform structures as shown in Figure 1.

There are three network control systems on the CAN bus, a NCS constituted of node 2, 3, 4 called NCS1, a NCS constituted of node 5, 6, 7 called NCS2, and the NCS3 constitutes of node 8, 9, 10. Network node 4, node 7 and node 10 are proposed as position detection unit. Sample with rate 100Hz, then sent the results to the controller node 3, 6, 9 through the network. After the controller nodes receive the position measurement value, they will operate and generate control signal, and then send it to the actuator node 2, 5, 8 through the network. The control signal works on the DC servo motor directly by actuator node. Interfering node 1 generates useless interference packets, leading to higher network load, and even blocked. CAN bus communication speed is set to 80kbps; the size of each packet is set to 40bit, so the timing can be used to send the packet size.

4. Simulation and analysis

4.1 Simulation designs

To study how the network scheduling impacts on the control system, we established network control system simulation platform. Figure 1 showed the test platform of NCS. The output sample values was sent to a different port through the CAN network. Data was sent to the actuate node via the CAN network from the controller to control the DC motor, an additional interference node enhanced the simulation realism.

Network DC servo motor can be described with the following transfer function:

$$G(s) = \frac{1000}{s^2 + s} \quad (3)$$

However, in practical application, when starting or stopping the motor, the PID calculation would cause the accumulation points and larger system overshoot, even lead to system shocks. Therefore, integration

should not be adopted in this case, and PD algorithm is selected. We used parameters $K = 1.5$, $T_d = 0.035$, $N = 1$ for the proportional plus derivative controller control system, the sampling period was set to 10ms:

$$P(k) = K(r(k) - y(k)) \quad (4)$$

$$D(k) = \frac{NKT_d}{T_m}(y(k-1) - y(k)) \quad (5)$$

$$D(k) = \frac{NKT_d}{T_m}(y(k-1) - y(k)) \quad (6)$$

Where $y(k)$ is the output value of the current cycle, $y(k-1)$ is the output value of the previous cycle.

The platform of NCS was introduced in Figure 1. $r(k)$ is the given value and $u(k)$ is the controller output value. set $\varepsilon = 0.1$, the delay of data calculate, sensor data acquisition and actuator action are set to 0.1ms.

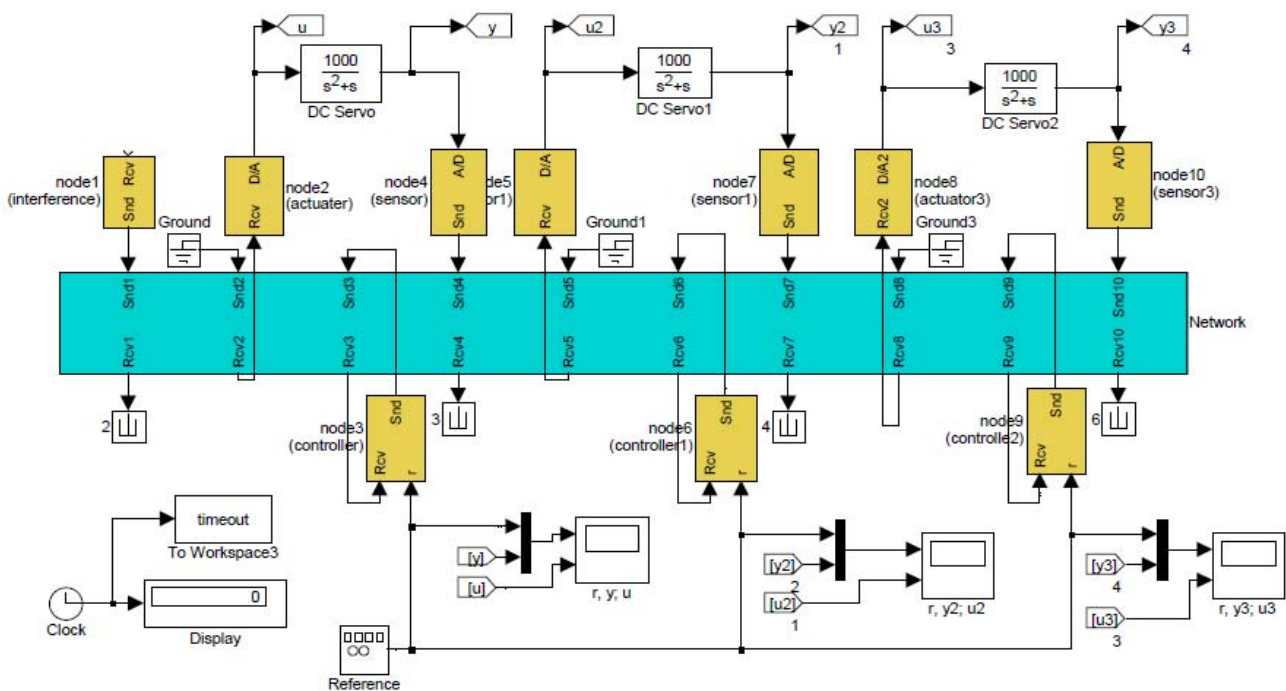


Figure 1 The test platform of Network control system

4.2 Simulation results

Let us set the baud rate of CAN bus to 80Kbps, set packet loss rate to 0, and use the fixed-priority scheduling algorithm (FP) to simulate. Its simulation results: NCS1 square-wave response; NCS2

square-wave response; NCS3 square-wave response and network control state can be shown in Figure 2 below.

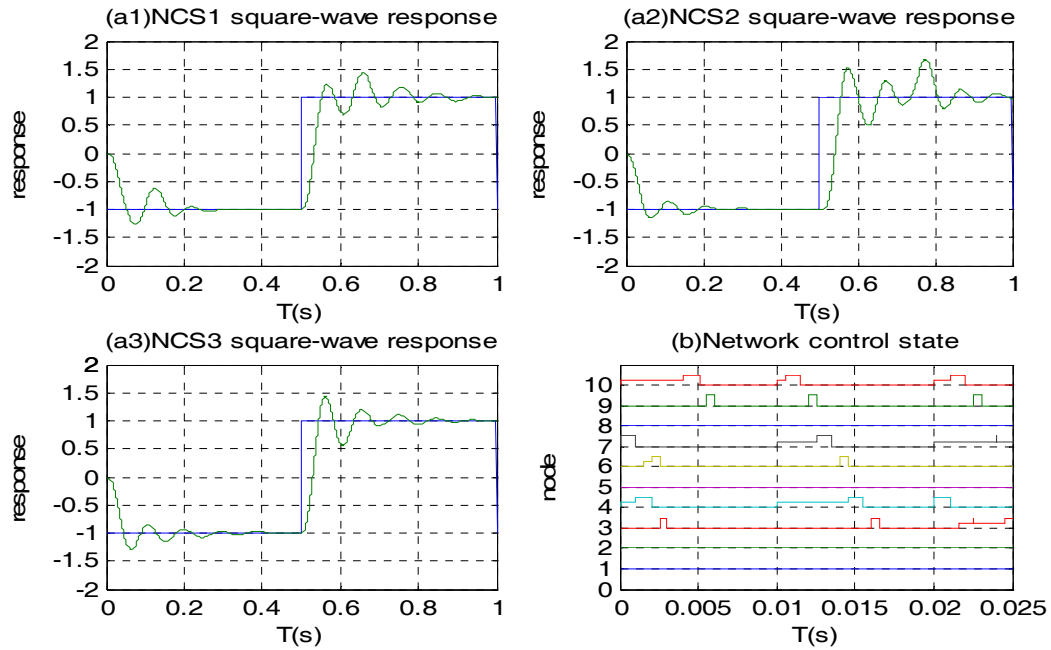


Figure 2 The simulation results Fixed-priority scheduling

- (a1) NCS1 square-wave response;
- (a2) NCS2 square-wave response;
- (a3) NCS3 square-wave response;
- (b) Network control state;

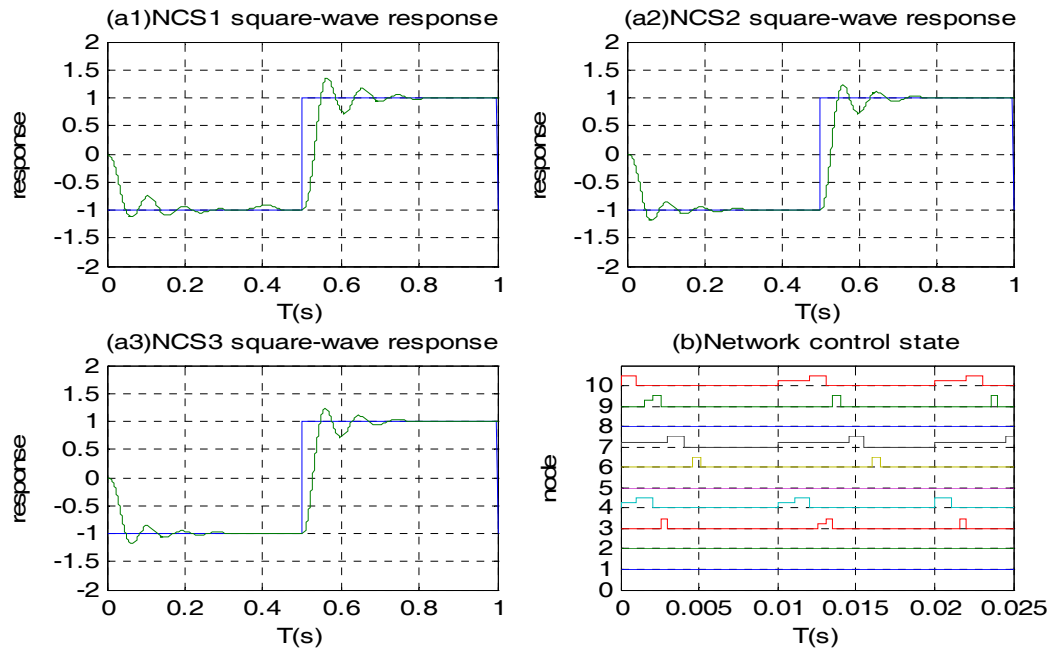


Figure 3 The simulation results of mixed traffic scheduling

- (a1) NCS1 square-wave response;
- (a2) NCS2 square-wave response;
- (a3) NCS3 square-wave response;
- (b) Network control state

In Figure 2 (a1), (a2), (a3), due to a large network delay, the dynamic response of two control subsystems is poor, adjustment time of both NCS1 and NCS2 subsystem are more than 0.3s. Adjustment time is so long that overshoot of subsystem NCS2 is more than 20%, which leads to less stability. Only NCS3 subsystem performs better response. Figure 2 (b) is the network scheduling status map. According to CAN protocol, a node with smaller number has higher priority, that means, node 1 has the highest priority, the sensor node 10 has the lowest priority. Node 3, node 9 and node 10 have larger network latency delays. The delay of both node 10 and node 9 are more than half of the sampling period, which result in these two dynamic control subsystems poor performances.

With the number of network nodes increases, the performance of control subsystem with lower priority node will get worse.

We simulated with the proposed mixed traffic scheduling algorithm, and just modified the scheduling algorithm, and chose square wave input, $k = 3$, $T_N = T_m/8$. Other conditions remained unchanged. The responses and network scheduling of the three control subsystems can be shown in Figure 3 above.

The results show that, when simulating with the mixed traffic scheduling algorithm of NCS, the system overshoots do not exceed 10%, adjustment time is less than 0.15s. There is a remarkable improvement compared with the original simulation. The network scheduling state diagram 3(b) of the network scheduling windows shows that network conflict will occur, but the network delay will become smaller. The time delay on Node 3, node 9 and node 10 has significantly reduced. As is seen, the overall performance of the system was highly improved.

5. Conclusion

In order to improve real-time performance of CAN bus network, in this paper, a new mixed traffic scheduling is presented and improved. Based on CAN bus, NCS simulation platform is constituted.

The simulation results show that the algorithm is feasible in the scheduling of information. The improved scheduling algorithm enhances the stability of the system and improves the overall performance of the system. In this paper, we made the design of network scheduling algorithm only under certain conditions, and for scheduling information generic research needs to continue in-depth.

Acknowledgments

This work was supported by the National Nature Science Foundation of Tianjin under grant No. 07JCYBJC05600 and by the Natural Science Foundation of Hebei Province under grant No. F2007000108.

References

- [1] Martion Andersson, Dan Henriksson, Anton Cervin. *TrueTime1.5-Reference Manual*, 2007.
- [2] K. Tindell, A. Burns, A. J. Willings. *Analysis of Hard Real-time Communication*. Real-Time Systems, Vol.9, No.2, pp.147-171, 1995.
- [3] N. Navet, *Controller area network CANs use within automobiles*. Potentials, IEEE. Vol.17, No.4, pp.12-14, 1998.
- [4] GC. Walsh, H. YE, L G. Bushnell, "Stability analysis of networked control systems", In: *Proc of the American Control Conf.* San Diego, California, pp.2876-2880, 1999:.
- [5] Marco Di Natale, Antonio Meschi. *Scheduling Message with Earliest Deadline Techniques*, Real-time Systems, Vol.20, No.3, pp.155-285, 2001.
- [6] Y. Tipsuwan , Mo-Yuen Chow. *Control methodologies in networked control systems* , Control Engineering Practice, Vol.11, Issue 10, pp.1099-1111, 2003,
- [7] Y. Q. Wang, S. X. Ding, H. Ye and G. Z. Wang. *A New Fault Detection Scheme for Networked Control Systems Subject to Uncertain Time-Varying Delay*. IEEE Transactions on Signal Processing, Vol.56, No.10, pp.5258-5268, 2008.
- [8] Y. Q. Wang, H. Ye and G. Z. Wang. *Fault detection of NCS based on eigendecomposition, adaptive evaluation and adaptive threshold*. International

Journal of Control, Vol.80, No.12, pp.1903-1911, 2007.

- [9] A. Cervin, D. Henriksson, B. Lincoln, Karl Erik Årzén, “Jitterbug and TrueTime: Analysis Tools for Real-time Control Systems”, In: *Proc of 2nd Workshop on Real-Time Tools*, Copenhagen, Denmark, 2002.
- [10] H. Shokry, M. Shedeed, S. Hammad, M. Shalan, A. Wahdan, “Hardware EDF scheduler implementation on controller area network controller”, In: *Proc. of 4th International Design and Test Workshop (IDT)*, pp.1-6, 15-17, 2009.
- [11] P. Pedreiras, L. Almeida, *EDF message scheduling on controller area network*, Computing & Control Engineering Journal, Vol.13, No.4, pp.163-170, 2002.
- [12] S. Fuster, F. Rodriguez, A. Bonastre, “Software-based EDF message scheduling on CAN network”, In: *Proc. of ICESS 2005 - Second International Conference on Embedded Software and Systems*, pp.400-455, 2005