

基于ARM的汽车ACCS硬件回路仿真平台设计

于海昕, 曾文华

(厦门大学软件学院智能信息技术福建省重点实验室, 厦门 361005)

摘 要: 汽车自适应巡航控制系统(ACCS)是一个完全的分布式嵌入式控制系统。该文采用硬件回路仿真的方法, 设计了一个基于 ARM 的汽车 ACCS 仿真平台, 介绍了平台的硬件结构和软件设计。

关键词: 自适应巡航控制系统; ARM; 硬件回路仿真; CAN

Design of HIL Emulation in ACCS of Car Based on ARM

YU Haixin, ZENG Wenhua

(Fujian Provincial Key Lab of Intelligent Information Technology, Software School, Xiamen University, Xiamen 361005)

【Abstract】 Adaptive cruise-control system of car is a completely distribute embedded control system. Using HIL control system means, the paper designs an emulation in ACCS of car based on ARM, and introduces the design of the hardware structure and software structure.

【Key words】 Adaptive cruise-control system (ACCS); ARM; Hardware in the loop (HIL); CAN

随着汽车工业的发展, 汽车电子技术也得到了迅速的发展。如今以人为本的观念深入人心, 人的生命财产提到了重要的高度。每年世界各地因交通事故而伤亡的人数逐年增多, 给人类造成了巨大的损失。因此, 人们越来越多地关注汽车安全问题。除此, 欧洲、北美洲等发达国家和地区有笔直的高速公路网络, 使用汽车出行是大众通常的选择, 司机经常因过度驾驶得不到休息, 存在着大量的交通隐患, 同时也损害身体健康。为了解决安全驾驶和疲劳驾驶问题, 越来越多的人开始研究汽车自适应巡航控制系统(Adaptive Cruise-control System, ACCS)。这种系统在笔直的高速公路上, 以不经常改变的恒定速度运行的情况下效果较好。通过 ACCS 的智能控制, 可以在相当的程度上减小驾驶员的疲劳度, 提高汽车的安全可靠性。

汽车 ACCS 是一种智能化的自动控制系统, 主要由雷达传感器、轮速传感器、发动机控制器和扭矩控制器等组成。雷达传感器可探测到汽车前方的距离, 在前后轮毂上均装有轮速传感器, 可测出车辆的行驶速度。发动机控制器和扭矩控制器用以探测与调整发动机接通和输出扭矩, 以提高发动机的动力性, 并适时调整车辆的运行速度。各种控制器和传感器均由车内计算机控制。装有自适应巡航控制系统的智能汽车, 采用主动安全技术, 系统通过各种传感器不断收集汽车、道路和周围环境等方面的信息, 通过计算机来调整汽车的运行状态。它能够准确地判断汽车四周的安全情况, 自动采取措施回避危险或者选择安全的行车路线和工作状态。

1 ACCS 设计的基本思想

汽车 ACCS 系统的模拟大多数是通过 Matlab 仿真采用单纯的软件方法实现的, 本文提出的硬件回路仿真平台(Hardware in the Loop, HIL)是在嵌入式系统的环境下, 模拟真实汽车的环境, 这样, 更加接近现实汽车的客观情况并提高了实验环境的真实性和可靠性。图 1 是汽车 ACCS 硬件回路仿真平台硬件逻辑图, PC 仿真端和嵌入式系统通过必要的 I/O 接口进行信息交换, 如数字信号、模拟信号和串口等。

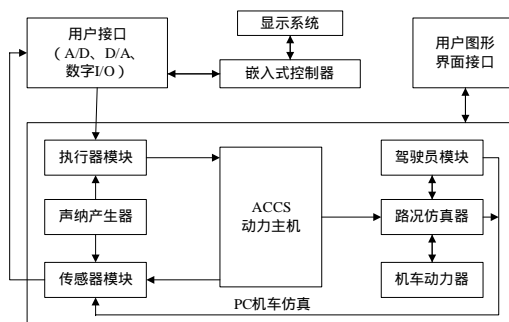


图 1 汽车 ACCS 的 HIL 平台硬件逻辑

除此, 在平台的设计中还要注意以下几个方面的内容:

- (1) 模拟系统不能完全替代原有汽车的物理系统, 因此, 应该具有使能控制信号启动或者停止 ACCS 系统。
- (2) 在 ACCS 系统正常运行的过程中, 驾驶员有权利改变目前的汽车运行速度。
- (3) 对于加速控制和刹车控制模块设置开关控制, 可以不使用 ACCS, 而让驾驶员单独行使控制权。

系统的控制器采用闭环反馈控制系统, 首先将预估输出值输入给控制器, 由控制器产生控制信号并送给被控系统, 由被控系统产生实际的输出速度值, 同时将输出速度值返回给控制器, 控制器将实际输出值和预估输出值进行比较, 调整它们之间的误差, 经过算法控制, 有效控制实际的输出值。其具体算法可用模糊 PID 控制算法。

2 汽车 ACCS 系统 HIL 平台的硬件设计

汽车 ACCS 的硬件部分除 PC 外主要使用的是 10 块 EasyARM2100 总线协议使用的是 CAN 总线结构。其由 CAN 总线收发器、JTAG 接口、4 个并口、A/D 转换器和相应的连

基金项目: “985 工程”智能化国防信息安全技术科技创新平台项目

作者简介: 于海昕(1980 -), 男, 硕士生, 主研方向: 嵌入式系统; 曾文华, 教授、博导

收稿日期: 2006-09-06 **E-mail:** yuyugo_fobje@sina.com

接线构成。硬件逻辑结构如图 2 所示。

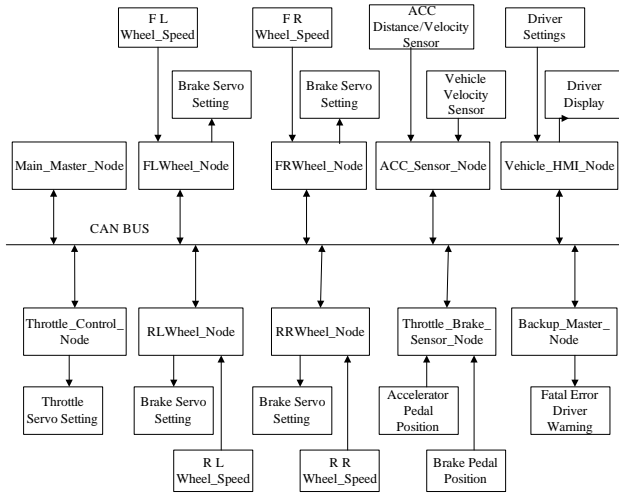


图 2 ACCS 的硬件逻辑

每个节点通过各自的接口与 CAN 总线相连。相对各个节点的输出模块是数字量输出到并口，而相对各个节点的输入模块是传感器的模拟量输入。在图 2 中，CANBUS 上面的 5 个节点依次是主控制节点、左前轮节点、右前轮节点、ACCs 传感器节点、人机界面节点；CANBUS 下面分别是节气门控制节点、左后轮节点、右后轮节点、节气门刹车节点、系统备用节点。主控节点接收各传感器节点的信息，经过控制运算再通过网络送给各执行节点；节气门开度控制节点接收从主节点发回的控制量，通过并口返回给 PC 端的虚拟执行器；4 个车轮(前后左右)节点功能基本相同，接收 PC 端返回的 4 个车速信息，通过网络传给主控节点，然后接收主控节点的刹车信息返回给 PC。每个车轮都由各自的传感器把各自的速度值输入到各自节点中，同时将各自的数字量输出到并口；雷达接口节点(传感器节点)作为一个专门的传感器节点接收前车车距、相对速度、车体速度等信号；人机界面接口主要是设置输入和显示输出，接收驾驶员控制信息，比如 CCS 定速信息等，并显示车速等信息，类似车辆电子仪表盘；节气门刹车传感器节点用于加速控制和刹车控制；备用节点报告是否有致命错误，并启动备用主节点，该节点为主控节点的热机备用节点。

基于硬件回路仿真平台的汽车 ACCS 是通过 4 个并口(LPT)和一个数模转换控制器(DAC)使各个节点间的信息与 PC 端进行交互。

表 1 LPT1 的定义

Pin	引脚定义	输入/输出
1	Cruise Enable	Output
2	Throttle Setting LSB	Input
3	Throttle Setting	Input
4	Throttle Setting	Input
5	Throttle Setting	Input
6	Throttle Setting	Input
7	Throttle Setting	Input
8	Throttle Setting	Input
9	Throttle Setting	Input
10	Throttle Setting	Input
11	Throttle Setting MSB	Input
12	CCS Running	Input
13	CCS Error	Input
14	Speed -5	Output
31	Speed +5	Output
32	CCS Warning	Input
36	Cruise Disable	Output

表 2 LPT2 的定义

Pin	引脚定义	输入/输出
1	Cruise Enable	Output
2	Throttle Setting LSB	Input
3	Throttle Setting	Input
4	Throttle Setting	Input
5	Throttle Setting	Input
6	Throttle Setting	Input
7	Throttle Setting	Input
8	Throttle Setting	Input
9	Throttle Setting	Input
10	Throttle Setting	Input
11	Throttle Setting MSB	Input
12	CCS Running	Input
13	CCS Error	Input
14	Speed -5	Output
31	Speed +5	Output
32	CCS Warning	Input
36	Cruise Disable	Output

(1)LPT1 中主要有节气门的 10 位数字量输入、CCS Running 输入、CCS Error 输入、CCS Warning 输入、Cruise Enable 输出、Speed +5 和-5 输出、Cruise Disable 输出。其中 Cruise Enable 信号是 PC 仿真使用的使能控制，Cruise Disable 是用来禁止 ACCS，Speed +5 和-5 输出用来调节速度，每次调节值为 5 个单位，CCS Running 是运行状态标志，CCS Warning 是非致命错误标志，CCS Error 是致命错误标志。如表 1 所示。

(2)LPT2 主要有 8 位左前轮数字量输入、8 位右前轮数字量输入。如表 2 所示。

(3)LPT3 主要有 8 位左后轮数字量输入、8 位右后轮数字量输入。管脚接线与表 2 类似。

(4)LPT4 主要预设置和模拟接口管脚，它与 A/D 转换器芯片相连。有 RST、CLK、SDI、LDA、LDB、CS1、CS2、CS3、CS4、CS5。其中 RST 用作复位，SDI 和 CLK 使能位触发移位寄存器，同时 LDA 或 LDB 执行相应 D/A 转换，当芯片的 CS 置为低电平时 LDA 和 LDB 激活。本系统共使用 9 路传感器信号(模拟量)，来全面反映车辆模型的运行状态，分别是左后轮车速(Rear Left Wheel Speed)、右后轮车速(Rear Right Wheel Speed)、右前轮车速(Front Right Wheel Speed)、左前轮车速(Front Left Wheel Speed)、驾驶员节气门开度设置(Driver Throttle Setting)、驾驶员刹车设置(Driver Brake Setting)、ACC 前车相对速度(ACC Target Distance)、线性车体速(Linear Vehicle Speed)、汽车设置(Driver Setting)。如表 3 所示。

表 3 LPT4 的定义

Pin	名称	作用
1	!STR	RST
2	D0	CLK
3	D1	SDI
4	D2	LDA
5	D3	LDB
6	D4	CS1
7	D5	CS2
8	D6	CS3
9	D7	CS4
14	!ALF	CS5

(5)9 路模拟量是 PC 输出的数字信号，经过一块专门的 D/A 转换板后输出。由于信号类型不同，因此需要 DAC 进行转换，完成嵌入式系统和 PC 的正常通信。其基本思想是将并口 4 的部分引脚改为串行输入脚、时钟脚和一些片选信号脚，数模转换接口与各个信号和节点的接线如表 4 所示。

表 4 DAC 接线

功能	顺序	节点	引脚
SDI	Vout9	R R Wheel_Node	P0.27
RST	Vout8	R L Wheel_Node	P0.27
LDA	Vout7	F R Wheel_Node	P0.27
LDB	Vout6	F L Wheel_Node	P0.27
CS1	Vout5	Throttle_Brake_Sensor_Node	P0.27
CS2	Vout4	Throttle_Brake_Sensor_Node	P0.28
CS3	Vout3	ACC_Sensor_Node	P0.27
CS4	Vout2	ACC_Sensor_Node	P0.28
CS5	Vout1	Vehicle_HMI_Node	P0.27

3 汽车 ACCS 系统 HIL 平台的软件设计

图 3 给出了汽车 ACCS 的软件后台结构，其程序是在硬件设计的基础上用 C 语言实现的。每层的每个模块都把函数和数据输出给其上一层，最低 3 层存在于基本系统中，在

应用过程中并不产生。

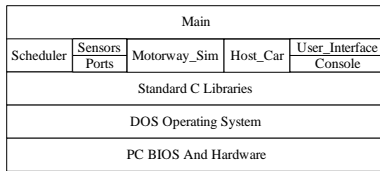


图 3 ACCS 的软件层次结构

其它的基本函数解释如下：

Main：主应用模块，包含主要的应用代码入口、程序控制流与用户交互功能。

Schedule：把函数和数据输出给 DOS 调度器。

Ports：给仿真并口提供定义和函数。

Sensors：传感器和实现器的接口函数。

Motorway_Sim：路况仿真数据和函数需求。

Host_Car：在仿真路况的条件下执行非线性的动态模块。

Console：一些简单的控制函数。

User_Interface：执行启动菜单，运行时间信息菜单和数据接口。

此外，一些有用的信息用某个按键代表，其含义如下：

‘q’：退出并返回主菜单。

‘1’：路况正常。

‘2’：路况较湿有雨情况。

‘3’：路况有雪。

‘4’：路况有冰。

‘a’：增加路的坡度。

‘z’：减少路的坡度

‘s’：增加风阻力。

‘x’：减小风阻力。

‘e’：开启 ACC。

‘d’：禁止 ACC。

‘r’：ACC 复位。

‘t’：ACC 速度加 5 个单位。

‘g’：ACC 速度减 5 个单位。

主节点的主要功能函数(函数意义如英文名称)如下：

SCC_A_MASTER_Init_T2_CAN();

Debug_SerialComm_167_Init();

Vehicle_Status_Init();

Vehicle_Control_Init();

Vehicle_Speed_Init();

SCH_Add_Task();

SCC_A_MASTER_Start();

SCH_Dispatch_Tasks();

车轮节点(以左前轮为例)的主要功能函数如下：

SCC_A_SLAVE_Init_CAN();

Debug_SerialComm_167_Init();

FLWheel_Speed_Init();

FLWheel_Brake_Init();

SCH_Add_Task();

SCC_A_SLAVE_Start();

SCH_Dispatch_Tasks();

传感器节点主要功能函数如下：

SCC_A_SLAVE_Init_CAN();

Debug_SerialComm_167_Init();

ACC_Measurement_Init();

SCH_Add_Task();

SCC_A_SLAVE_Start();

SCH_Dispatch_Tasks();

人机界面节点主要功能函数如下：

SCC_A_SLAVE_Init_CAN();

Debug_SerialComm_167_Init();

Vehicle_LCD_Init();

Vehicle_Action_Detection_Init();

SCH_Add_Task();

SCC_A_SLAVE_Start();

SCH_Dispatch_Tasks();

节气门控制节点主要功能函数如下：

SCC_A_SLAVE_Init_CAN();

Debug_SerialComm_167_Init();

Vehicle_Throttle_Control_Init();

SCH_Add_Task();

SCC_A_SLAVE_Start();

SCH_Dispatch_Tasks();

节气门刹车节点主要功能函数如下：

SCC_A_SLAVE_Init_CAN();

Debug_SerialComm_167_Init();

Vehicle_Pedal_Detect_Init();

SCH_Add_Task();

SCC_A_SLAVE_Start();

Jitter_Init();

SCH_Dispatch_Tasks();

备用节点主要功能函数如下：

Backup_Master_Init();

SCC_A_SLAVE_Init_CAN();

Debug_SerialComm_167_Init();

Limp_Home_Init();

SCH_Add_Task();

SCC_A_SLAVE_Start();

SCH_Dispatch_Tasks();

4 结论

在当今社会中，汽车电子的应用领域非常广泛，产生了较大的社会效益，也拉动了汽车工业和汽车控制的发展。本文结合社会需求，打破传统的单纯软件模拟仿真的思想，从嵌入式平台入手，介绍了 ACCS 的整体设计过程，开拓了 ACCS 的设计思路，也为今后相关方向的研究做了一个铺垫，具有一定的启示作用。下一步将对控制算法和通信协议作进一步的研究，进而全面提高系统的控制效果和安全可靠性。

参考文献

- 1 Ayavoo D, Pont M J. Two Novel Shared-clock Scheduling Algorithms for Use with CAN-based Distributed Systems[C]//Proc. of the 2nd UK Embedded Forum, Birmingham. 2005-10.
- 2 Ayavoo D, Pont M J, Fang Jianzhong, et al. A “Hardware-in-the Loop” Testbed Representing the Operation of A Cruise-control System in A Passenger Car[C]//Proceedings of the 2nd UK Embedded Forum, Birmingham. 2005-10.
- 3 Short M, Pont M J, Huang Q. Safety and Reliability of Distributed Embedded Systems: Simulation of Vehicle Longitudinal Dynamics[R]. Embedded Systems Laboratory, Technical Report: ESL04/01, 2004-10-07.
- 4 Short M, Pont M J, Huang Q. Safety and Reliability of Distributed Embedded Systems: Simulation of Motorway Traffic Flows[R]. Embedded Systems Laboratory, Technical Report: ESL04/02, 2004-10-07.
- 5 Short M, Pont M J, Huang Q. Safety and Reliability of Distributed Embedded Systems: Development of a Hardware-in-the-Loop Test Facility for Automotive ACC Implementations[R]. Embedded Systems Laboratory, Technical Report: ESL04/03, 2004-10-07.