

基于 DMS 和 EDF 的 CAN 总线混合调度算法

许 万, 扬光友, 唐宗美

(湖北工业大学 机械工程学院, 武汉 430068)

摘要: 针对 CAN 总线调度问题, 分析了 CAN 网络中信息帧发生传输错误的概率化最大响应时间, 融合 DMS 算法的可预测性良好、处理器开销低以及对数分区 EDF 算法的灵活性强、网络利用率高的特点, 提出一种考虑消息关键性的混合调度算法; 在上述研究基础上, 利用 MATLAB/SimEvents 工具箱建立 CAN 网络模型, 并对 3 种调度算法进行了仿真, 当网络利用率达到约 85% 时, 混合调度算法丢包率为 0.8%, 但丢失的数据不包含硬实时消息, 而且成功降低了处理器的额外开销时间, 仿真结果表明混合调度算法既能提高系统的实时性, 又能减小处理器的运算负担。

关键词: WCRT; DMS; 对数分区 EDF; 混合调度; SimEvents 工具箱

CAN Bus Hybrid Scheduling Algorithm Based on DMS and EDF

Xu Wan, Yang Guangyou, Tang Zongmei

(School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China)

Abstract: In this paper we provide a probabilistic analysis of worst case response time of message with faults in the Controller Area network (CAN). By studying and analyzing static scheduling algorithm Deadline Monotonous Scheduling (DMS) and dynamic scheduling algorithm Earliest Deadline First (EDF) in the CAN, and combining the good predictability and cheap processor cost of DMS algorithm with the strong flexibility and high utilization of logarithm partition EDF algorithm, this paper presented a hybrid scheduling algorithm with considering critical level of messages. Based on the work mentioned above, the simulation platform of CANbus is developed with MATLAB/SimEvents tool box. The platform was used to real-time analysis for the scheduling algorithms mentioned before, the simulation results showed that the hybrid scheduling algorithm could not only improve the real-time performance of system, but also could reduce the cost of processor.

Key words: WCRT; DMS; logarithmic partition EDF; hybrid scheduling; SimEvents tool box

0 引言

CAN 总线广泛的应用于汽车、船舶、航空航天、工业自动化等领域。当网络负载加重时, 低优先级报文的响应时间受到严重影响, 因此为保证分布式功能信号准确、实时、稳定地传输, 引入合适的消息调度机制是必要的。目前国内外主流的调度机制可分为静态调度和动态调度两种。静态调度主要有基于表的调度策略^[1]、时间触发 CAN^[2]以及比率单调调度^[3]、截止期单调调度^[4]等, 它们具有可预测性良好、设计简单及处理器开销低等优点, 但也存在着网络利用率低、灵活性差、对非周期性任务调度效率低等缺点; 动态调度主要有最小截止期优先^[5]、最小松弛优先^[6]、二权值晋升^[7]、对数分区 EDF^[8-9]等, 它们对网络资源的利用率较高、灵活性强、可实时适应系统配置的变化, 但也存在实现较为复杂、网络节点额外开销时间大和传输时刻不确定等不足。本文在分析两类算法的基础上, 融合静态调度算法和动态调度算法二者的优点, 提出一种考虑消息关键性的混合调度算法。

1 CAN 消息响应时间分析

在 CAN 网络中, 同一时刻多个消息同时处于发送状态会

给低优先级消息带来严重的阻塞, 从而降低系统的性能, 因此需要对 CAN 消息的响应时间以及在高负载下总线的调度机制做进一步研究。

WCRT (Worst Case Response Time, 最大响应时间) 最早由英国约克大学的 Ken Tindell 和 Alan Burns 提出^[10-12]。该方法是利用上限函数描述 CAN 总线带冲突检测的载波侦听多路访问协议 (CSMA/CD), 计算出最差条件下的响应时间^[13]。考虑发生传输错误的最大响应时间数学表达式为:

$$R_i = J_i + B_i + C_i + I_i(t_i) + E_i(t_i) \quad (1)$$

式中, J_i 为软件抖动; B_i 为阻塞时间; C_i 为总线占用时间; $I_i(t_i)$ 为在时间段 t_i 内所有优先级高于 i 的消息对其造成的干扰; $E_i(t_i)$ 为因发生传输错误而引起的延迟。那么有:

$$B_i = \max_{\forall k \in lp(i)} (C_k) + S \quad (2)$$

$$C_i = (g + 10 + 8b_i + \left\lceil \frac{g + 8b_i - 1}{4} \right\rceil) * \tau_{bit} \quad (3)$$

$$I_i(t_i) = \sum_{\forall j \in hp(i)} \left\lceil \frac{t_i - C_j + J_j + \tau_{bit}}{T_j} \right\rceil (C_j + S) \quad (4)$$

式中, S 为帧间隙, 大小为 $3\tau_{bit}$, τ_{bit} 是位周期; g 根据帧类型 (标准帧或扩展帧) 取 37 或 57; b_i 为数据场字节数; T_j 为消息发送周期。

设产生发送错误的概率服从泊松分布 $F \sim Po(\lambda)$, 则在时间段 t_i 内产生 k 次发送错误的概率可由公式 (5) 得到, 最差情况下每次产生发送错误的最长延时计算式为 (6)^[4], 消息 i 因产生发送错误而引起的延迟可由公式 (7) 算出。

$$P_i(F = k) = \frac{e^{-\lambda t_i} (\lambda t_i)^k}{k!} \quad (5)$$

收稿日期: 2013 - 12 - 26; 修回日期: 2014 - 02 - 24。

基金项目: 国家自然科学基金 (51174084); 武汉市科技青年晨光计划 (201271031386)。

作者简介: 许 万 (1979 -), 男, 湖北武汉人, 工学博士, 副教授, 主要从事网络化控制系统、运动控制等方向的研究。

$$E_i = e_{\max} + \max_{j \in hp(i)} C_j \quad (6)$$

$$E_i(t_i) = kE_i, \text{ 概率为 } P_{i_j} (F = k) \quad (7)$$

式中, e_{\max} 是由于发送最长错误帧而产生的延迟, 长度为 $29\tau_{\text{bit}}$ [14]。 C_j , $E_i(t_i)$ 含义与公式 (1) 中的一样。

为使消息在 CAN 网络中有效、准确地传输, 就必须对网络资源进行合理地分配与调度, 以保证消息在规定的截止期内准确到达目标节点, 一个 CAN 总线系统在时间特性方面的约束要求如式 (8) 和式 (9) 所示。

$$R_i < D_i \leq T_i \quad (8)$$

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \leq 1 \quad (9)$$

式中, D_i 表示消息截止期, 公式 (9) 不等号左边表示总线利用率, C_i 、 T_j 符号含义同前述公式。

2 混合调度算法

通讯过程中的数据分为 3 类: 硬实时数据、软实时数据和非实时数据 [15]。本文提出的混合调度算法的思想是: 依据消息关键性分配优先级和对数分区 EDF 算法对硬实时消息进行调度, 软实时消息和非实时消息采用 DMS 算法调度。

传统 EDF 算法是将最高优先级指派给剩余截止期 d_{rel_i} 最短的消息, 但 d_{rel_i} 最短的消息不一定最关键。如系统中最关键的消息记作 A, 显然其截止期 D_A 很小, 但截止期仍需不小于消息传输时间 C_A , 记作 $D_A \geq C_A$ 。如采用对数分区 EDF 算法, 在某一次优先级分配中, 假设在时间轴上有一个消息 B, 满足 $d_{\text{rel}_B} < d_{\text{rel}_A}$, 且 A 与 B 不属同一分区, 显然此时 B 占用总线, 然而等到 B 传输完毕后, A 可能已超出其截止期。考虑上述问题, 本文提出将例如机器故障和急停等硬实时消息与普通硬实时消息区别开来, 根据消息关键性来分配优先级, 并结合对数分区 EDF 和 DMS 两种算法对系统中消息进行调度。

以 CAN 标准帧为例, 混合调度算法的优先级分配步骤如下:

- (1) 将 0 分配给硬实时消息标识符最高位, 将 1 分配给软实时消息和非实时消息最高位;
- (2) 将 0 分配给紧急硬实时消息标识符次高位, 标识符剩余的 9 位根据系统中消息的关键性来分配, 关键性越高, 优先级越高;
- (3) 将 1 分配给普通硬实时消息标识符次高位, 标识符剩余的 9 位采用分层架构: 上层采用对数分区 EDF 算法, 下层采用 DMS 算法;
- (4) 将 0 分配给软实时消息标识符次高位, 标识符剩余位

采用 DMS 算法分配;

- (5) 将 1 分配给非实时消息标识符次高位, 标识符剩余位采用 DMS 算法分配。

图 1 所示为混合调度算法标准帧 11 位标识符分配结构图。

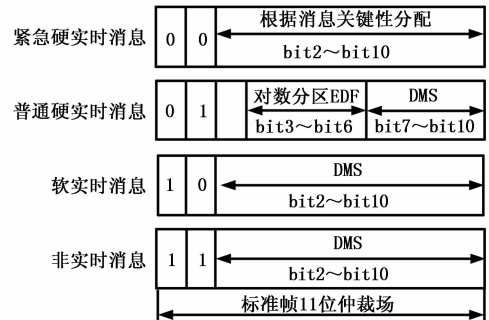


图 1 混合调度算法标准帧 11 位标识符分配结构

如图 1 所示, 普通硬实时消息上层采用 EDF 算法对消息进行分区, 若两个消息的相对截止期通过分区后分属不同区, 则相对截止期越小其优先级越高; 下层采用 DMS 一方面是当两个消息经分区后属同一区, 则利用固定优先级 DMS 再进行调度, 另一方面也起到一个消息滤波的作用, 为每一个普通硬实时消息分配一段独一无二的标识符段 (消息类别段 + DMS 分配段), 以保证目的节点正确接收消息。此外, 如果系统中硬实时消息较多, 可以考虑使用仲裁场为 29 bits 的 CAN 扩展帧。但扩展帧的头开销很大, 在传输时会比标准帧要多占用 20%~30% 的带宽 [16]。

3 仿真测试及结果分析

本文利用 MATLAB/SimEvents 离散事件仿真工具箱在 MATLAB/Simulink 环境下搭建 CAN 网络仿真测试系统, 模型模拟报文在 CAN 网络中所经历的消息产生、排队等待、等待超时、竞争总线、竞争失败重发、占用总线、传输错误等一系列过程。

3.1 CAN 网络仿真测试模型建立

仿真平台如图 2 所示, 主要包含 3 个部分: 网络节点、总线仲裁机制、数据显示。网络中共有 10 个 CAN 节点, 每一个 CAN 节点模型如图 4 所示, 总线仲裁机制模块用于模拟 CAN 网络的 CSMA/CA (载波感测多元存取/具碰撞侦测) 协议; 数据显示部分由两个示波器组成, 两个示波器分别用于显示仿真时间内总线上发送成功的消息个数和网络资源利用率。

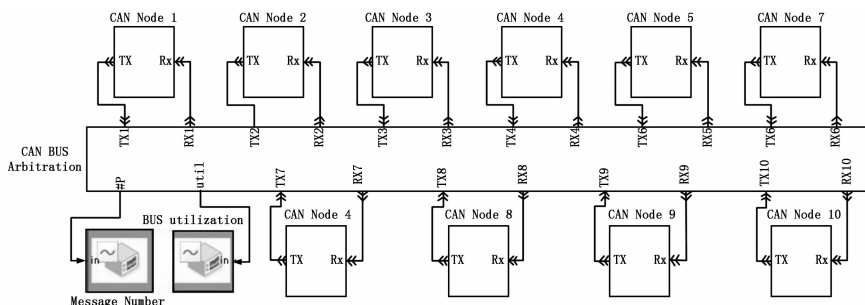


图 2 CAN 网络仿真测试系统模型

表 1 CAN 总线 10 个节点的消息时间特性

节点序号	消息长度(bytes)	周期(ms)	截止期(ms)	属性	节点序号	消息长度(bytes)	周期(ms)	截止期(ms)	属性
1	8	10	4	紧急硬实时	6	8	32	22	普通硬实时
2	8	12	6	紧急硬实时	7	8	38.5	24	软实时
3	8	24	10	普通硬实时	8	8	50	26	软实时
4	8	26.5	12	普通硬实时	9	8	偶发	5000	非实时
5	8	29.5	15	普通硬实时	10	8	偶发	5000	非实时

3.2 实时调度算法仿真及结果分析

利用上述 CAN 网络仿真测试系统模型分别对 DMS 算法、对数分区 EDF 算法以及本文提出的混合调度算法进行仿真，各仿真节点消息时间特性如表 1 所示。

10 个节点分别按照表 1 中消息的时间特性发送/接收消息，总线波特率设定为 50 kbit/s，仿真时间为 20 s。为便于区分多条消息的响应时间曲线，只截取 100 ms 时间范围内消息延迟时间分布，如图 3、图 4、图 5 所示，其中 dt 为消息延迟时间。

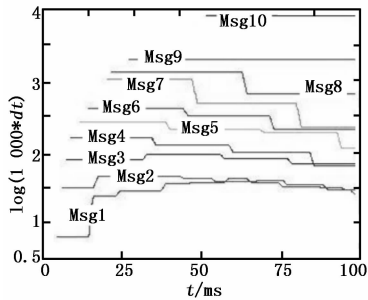


图 3 DMS 算法的消息延迟

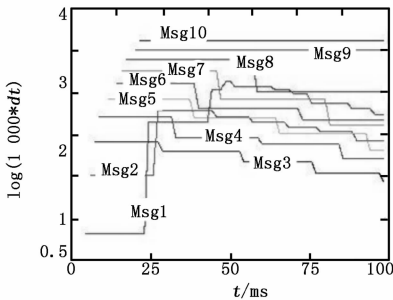


图 4 对数分区 EDF 算法的消息延迟

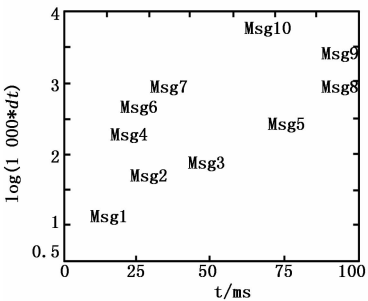


图 5 混合调度算法的消息延迟

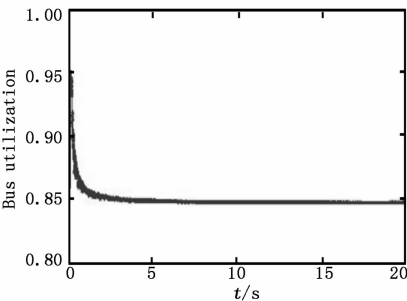


图 6 网络利用率

图 3、图 4、图 5 分别表示 10 个 CAN 网络节点发送的消息在 100 ms 内采用 3 种算法调度时各自的响应时间，Msg1、Msg2...Msg10 分别表示由节点 1、节点 2...节点 10 发送出的消息。图 3 为 DMS 算法的消息延迟特性图，由于 DMS 算法严格按照消息的截止期的大小分配优先级，消息响应延迟也严格按照优先级的由高到低呈递增趋势。图 4 所示为对数分区 EDF 算法的消息延迟特性图，从图中可以看出，消息的优先级呈动态变化，且在实际应用中经常会出现多个消息的周期成倍数关系，所以消息的响应时间分布相对集中。图 5 为混合调度算法的消息延迟特性图，除了普通硬实时消息外全是固定优先级消息，普通硬实时消息的延时分布相对集中，其它消息的延迟按照优先级的由高到低呈递增趋势。

图 6 为网络的利用率。图 7 为分别使用 3 种调度算法调度所有消息，各个节点发送和接收到的消息数目。

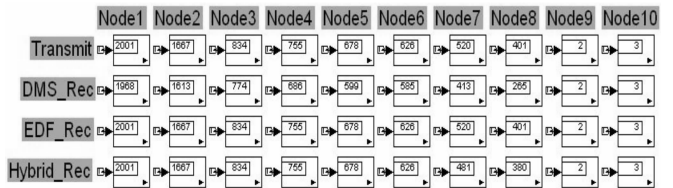


图 7 3 种调度算法的消息接收数目

由图 6、图 7 可知，当网络利用率达到约 85% 时，采用 DMS 算法，网络中丢包率为 7.85%，且包含普通硬实时消息，采用对数分区 EDF 算法时，网络中无丢包现象发生。采用混合调度算法时，网络中丢包率为 0.8%，但丢失的数据不包含硬实时消息，只有软实时消息。网络负载较大时，单纯地运用 DMS 算法会使网络丢包现象严重，采用对数分区 EDF 算法在本系统中虽无丢包现象，但是全部报文都参与动态优先级算法使得处理器额外开销时间增多。混合调度算法虽然牺牲了部分软实时和非实时消息的实时性，但是并未影响硬实时消息的实时性能，也满足系统的实时性要求，比起 DMS 算法，实时性有了较大提升，比起对数分区 EDF 算法，它有效地将报文进行了分类，使得参与动态优先级变换的报文数目减少，成功降低了处理器的额外开销时间。

4 结论

本文结合静态调度和动态调度的优点，并考虑消息的关键性因素，提出了一种混合调度算法，通过网络仿真工具箱 SimEvents 进行仿真测试、比较分析，结果表明该混合算法既提升了系统的实时性，同时也减小了处理器的运算负担。文中网络模型的部分模块是建立在一定的假设基础上的，与现实 CAN 网络还存在一定的差距，其仿真结果还存在一定的误差，还需要进一步提高模型精度。

参考文献:

[1] Hong S H. Scheduling algorithm of data sampling times in the integrated communication and control systems [J]. IEEE Transactions on control systems technology, 1995, 3 (2): 225 - 230.

[2] Leen G, Heffernan D. TTCAN; a new time - triggered controller area network [J]. Microprocessors and Microsystems, 2002, 26 (2): 77 - 94.

[3] Lui Sha, Rajkumar R. Generalized rate - monotonic scheduling

theory: a framework for developing real - time systems [C]. Proceedings of the IEEE, 1994, 82 (1): 68 - 82.

[4] Ian Broster, Alan Burns, Guillermo Rodriguez - Navas. Probabilistic Analysis of CAN with Fault [J]. IEEE, 2002.

[5] Natale M D. Scheduling the CAN bus with earliest deadline techniques [C]. 21st IEEE Real - Time Systems, 2000.

[6] Moke A K. Fundamental design problems of distributed systems for the hard - real - time environment [D]. Laboratory for Computer Science, MIT, Cambridge, Mass, 1983.

[7] 吴 涛. 基于 CAN 总线的工业测控通信系统的研究 [D]. 南昌: 江西师范大学, 2011.

[8] Stankovic J A, Spuri M, Ramam ritham K, et al. Deadline scheduling for real - time system: EDF and related algorithms [M]. Boston: Kluwer Academic, 1998.

[9] 谏介人, 彭 军, 吴 敏, 等. CAN 总线中 EDF 调度算法的研究与改进 [J]. 计算机测量与控制, 2008, 16 (11): 1678 - 1681.

[10] Tindell K, Burns A, Wellings A. Calculating Controller Area Network (CAN) message response time [J]. Control Engineer-

ing Practice, 1995, 3 (8): 1163 - 1169.

[11] Tindell K. An Extendible approach for analysing fixed priority hard real - time systems [J]. Journal of Real - Time System, 1994, 6 (2): 133 - 152.

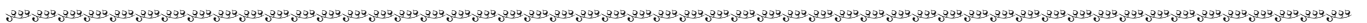
[12] Burns A. Preemptive Priority - Based Scheduling: an appropriate engineering approach [J]. Advances in Real - Time system, 1994: 225 - 248.

[13] 陈 曦, 刘鲁源, 吕伟杰, 等. 基于排队论的 CAN 总线消息响应时间建模与分析 [J]. 天津大学学报, 2012, 45 (3): 228 - 235.

[14] Ian Broster, Alan Burns, Guillermo Rodriguez - Navas. Timing Analysis of Real - Time Communication Under Electromagnetic Interference [J]. Real - Time System, 2005, 30: 11.

[15] 李明波. 基于 CAN 总线的实时通讯研究 [D]. 北京: 北京化工大学, 2006.

[16] 周时莹. CAN 网络控制系统的智能调度研究 [D]. 吉林: 吉林大学, 2009.



(上接第 1468 页)

数据传输量比较大, 将现场数据高速传给远端客户; 也可以利用操作系统的远程桌面等技术, 均可方便实现远程控制天线伺服^[7]。具体使用时, 可根据被测雷达的远场条件, 现场具备的设施, 灵活应用上述远程控制方式。

3 与传统方法应用比较

传统架设一般要两人配合, 由于仪器放置位置一般在天线侧后较低位置, 调整天线的人一般无法直接看清仪器示值变化, 需一人实时判读仪器指挥另一人实时调整天线。而本文设计研制了智能天线伺服系统, 可实现 4 种天线调整方式: 遥控器调整、下位机程序自动调整、上位机程控自动调整以及上位机人机界面根据仪器峰值示值趋势, 人工手动控制伺服调整。

为了对比传统架设方法和伺服系统自动架设效果, 先架设标准信标, 信标的全向有效功率、距离已知, 可计算出测试点的功率约定真值, 在同一个测试点, 分别用两种方案进行标准增益天线的多次架设测试, 架设效率以及测量结果等参数如表 1 所示。

表 1 人工架设与伺服架设天线对比

架设方法	人工架设	伺服架设	备注
人员情况	2 人处于辐射区	1 人在辐射区外	测量同一已知信标
准备时间 (min)	20	8	6 次重复架设从运输装箱状态到架设调整完毕
测量重复性 (dB)	0.6	0.2	6 次重复架设测量计算标准偏差
测量准确度 (dB)	1.2	0.5	6 次重复架设测量平均值与约定真值的差值

由实验数据可见, 智能伺服系统用人少, 避免了人员遭辐射, 架准备时间短, 测量重复性好, 架设准确度高。

4 结论

本文利用模块化设计思路, 在硬件设计上, 利用部件集成技术, 选用各成熟模块完成了硬件系统, 降低了硬件设计成本; 在软件设计上, 避开复杂的传统伺服控制设计思路, 配合仪器判读, 模拟人工调整过程, 设计了简易实用的伺服自动架设控制程序。根据雷达测试专用需要, 还增加了信号切换, 仪器控制等功能, 具有较高的性价比; 通过与传统人工架设天线试验数据相比较, 该伺服系统避免了架设调整人员遭受辐射, 提高了测试效率和测量准确度, 可以推广应用到类似的测试中。

参考文献:

[1] 樊 东, 周国荣. 永磁同步伺服电机智能控制器的研制与开发 [J]. 计算机测量与控制, 2006, 14 (3): 324 - 326.

[2] 马 骏, 黄攀峰. 一种网络化伺服电机运动控制系统设计 [J]. 计算机测量与控制, 2011, 19 (1): 69 - 70.

[3] 马小陆, 陶维青. 基于 430 单片机的远程测控单元 [J]. 仪器仪表用户, 2004, 11 (5): 69 - 70.

[4] 杨井胜. 基于 Labview 的雷达接收自动测试系统设计 [J]. 四川兵工学报, 2009, 30 (7): 24 - 27.

[5] 李培元. LabVIEW 下 GPIB 仪器网络测试研究 [J]. 仪器仪表标准化与计量, 2004, (6): 17 - 19.

[6] 吕继宇, 张华春, 阴和俊. 基于 LabVIEW 的频谱仪控制系统设计 [J]. 测试技术学报, 2005, 19 (4): 426 - 431.

[7] 鲍连升, 赵 峰. 基于 LabVIEW 的微波射频器件自动化测试系统设计 [J]. 电子测试, 2009, (5): 44 - 46.