MVB周期信息的实时调度

朱琴跃[1],谢维达[2],谭喜堂[1],赵亚辉[1]
(1. 同济大学 电子与信息工程学院,上海 200092; 2. 同济大学 铁道与城市轨道交通研究院,上海 200092)
(zqymelisa@mail tongji edu cn)

摘 要:多功能车辆总线 MVB网络对周期信息的通信提出了很高的实时要求,其通信的实时调度主要由 MVB总线管理设备利用实时调度表来完成。在分析一般现场总线周期信息实时调度的基础上,结合 MVB周期信息的通信特点,提出了采用同步 RM调度算法来建立 MVB实时调度表的原理与方法;并进一步提出了采用基于任务响应时间的方法对该调度算法进行可调度性分析,给出了一种有效算法,用以实现对调度表的有效性判断;最后通过 MVB周期信息实例阐述了所提出的实时调度算法及其可调度性分析方法的具体应用,为实际 MVB网络的应用研究提供了理论指导。

关键词:周期通信;实时调度表;MVB;调度算法;可调度性分析
中图分类号: TP301. 6; U285 **文献标志码**: A

Real-time scheduling of periodic messages for MVB

ZHU Qin-yue[1], XIE Wei-da[2], TAN Xi-tang[1], ZHAO Ya-hui[1]
(1. *College of Electronic and Information Engineering, Tongji University, Shanghai* 200092, *China*;
2. *Railway and Urban Mass Transit Research Institute, Tongji University, Shanghai* 200092, *China*)

Abstract: Transmission of periodic messages processing in Multifunction Vehicle Bus(MVB) network should be highly in compliance with the time constraints of the communication. Generally, the bus administrator of MVB utilizes real-time schedule table to manage the scheduling of periodic traffic. On analyzing the scheduling principles of periodic traffic on field bus, the approach how to build the real-time schedule table for MVB with adopting synchronous RM algorithm was introduced. Furthermore, the approach to schedulability analysis based on task response time was proposed, and a new algorithm used to judge the validity of schedule table was also put forward by analyzing the worst case response time. Finally, the application of real-time scheduling algorithm and related way of schedulability analysis were expounded by the example of MVB periodic traffic, which provided some useful guidance for the MVB application.

Key words: periodic communication; real-time schedule table; MVB; scheduling algorithm; schedulability analysis

0 引言

多功能车辆总线 (Multifunction Vehicle Bus, MVB)是列车通信网络标准 IEC 61375-1中明确定义的用于车辆内部各功能设备之间实现互连的网络总线,是应用于车辆控制这一特定场合的现场总线,主要支持周期性过程变量和偶发性消息两类数据的传输[1]。其中周期数据的传输是一个具有确定响应时间的实时通信过程,其通信的实时调度主要由总线管理设备利用实时调度表来完成,因此采用何种实时调度算法建立相应的实时调度表将直接影响着 MVB周期数据通信的实时性,进而决定了 MVB网络控制系统实时可靠的传输[2]。

迄今为止,众多学者对其他多种现场总线实时调度表的构建及相应的调度算法作了大量的研究工作,取得了一定的成果[3,4]。有必要对 MVB实时通信调度理论尤其是对其实时调度表的构建方法及相应的实时调度算法进行研究,以便为 MVB网络的应用研究提供系统的理论指导,满足实际应用的需要。

1 MVB周期信息通信原理

MVB总线上存在周期和非周期两类信息通信,根据它们各自通信的特点,MVB协议采取了基于时间触发和事件触发相结合的介质存取控制方式来有效地管理信息通信。其中,基于时间触发的集中式令牌控制方式被有效地用来管理周期信息通信,以确保其实时性;而对于非周期信息,由于其到达时间不确定,采用集中控制方式无法有效地利用总线的通信带宽,故采用事件触发方式进行管理。本文主要从实时性的角度来分析周期信息的传输及其相应的实时调度机制和工作原理,对非周期信息的通信原理不作进一步的研究。

1.1 MVB周期信息的基本通信过程

MVB通过总线上唯一的总线管理器 (Bus Administrator, BA)对介质存取采用周期性预分配的集中控制方式,按照生产者/仲裁者/消费者 PDC(Producer/Distributor/Consumer)模型来管理信息的实时通信。在通信过程中,由生产者 (MVB源设备)周期性地产生过程数据,消费者 (MVB宿设备)接收相应的过程数据,仲裁者 BA (MVB主设备)负责协调不同 MVB设备间生产者对总线的需求,数据通信模式采用源寻址

广播方式。

1.2 MVB周期信息的报文定时

MVB过程数据报文由主帧 Master_frame和从帧 Slave_frame组成。主帧主要由总线主设备发出,从帧则由从设备为响应此主帧而发出。具体主、从帧结构及过程数据报文定时关系分别见图 1、2所示。

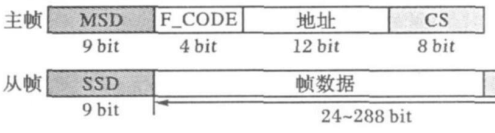

图 1 主帧/从帧结构

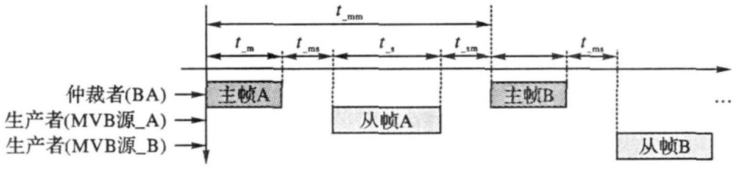

图 2 过程数据报文定时

由图 1可知:主帧的长度固定为 33 bit,从帧的长度为33～297 bit。由此可知,MVB总线完成一次报文传输的时间如图 2所示为:

$$t_{mm} = t_m + t_s + t_{ms} + t_{sm} = \frac{len(\text{Master\_frame})}{V} + \frac{len(\text{Slave\_frame})}{V} + t_{ms} + t_{sm} \quad (1)$$

其中:V为 MVB总线的通信速率,$len()$为数据帧的长度(以 bit为单位),t_{ms}为从主帧到响应该主帧的从帧之间的时间间隔,t_{sm}为从帧到下一主帧之间的时间间隔。

以 EMD介质为例:MVB网段间的最大电气长度 L = 2 km,通过 2个中继器连接(每个中继器的时延 t_{repeat} = 3 μs),则 $t_{ms} = 2 \times (6L + 2t_{repeat}) + 3 = 39\ \mu s$[1];若取 t_{sm} = 3 μs,且 MVB的 V = 1.5 Mbps,则由式(1)可得:

$$t_{mm} = \frac{33}{1.5} + 9 + \frac{(24 \sim 288)}{1.5} + 39 + 3 = (86 \sim 262)\ \mu s \quad (2)$$

2 周期信息实时调度

在实时控制领域大致存在两类应用实时调度方法的系统:单处理器/多处理器硬实时系统和实时通信系统。前者需保证任务的执行在规定的时限内得到正确的结果;后者则要求信息能在一定的时间内完成传输,保证通信的实时性和准确性。但从调度的观点来看,两者存在许多相同之处[5]。因此,现场总线系统中信息的通信调度在某种程度上可看作是实时系统中任务的调度问题,周期信息的调度在某些方面可以借鉴单处理器的实时任务调度方法来实现。

目前,实时任务调度中较常用的是基于优先级驱动的调度算法,一般分为静态优先级调度和动态优先级调度。其中文献[6]提出的单调速率(Rate Monotonic, RM)算法和最小截止期优先(Earliest Deadline First, EDF)算法分别为上述两种优先级调度算法中的典型代表,以后出现的各种优化调度算法也都基于此算法演化而来[7,8]。

2.1 周期信息模型描述

周期任务一般可用一个三元组的实时模型说明其参数[9]。与此相对应,现场总线也可采用类似的模型描述周期信息。假设网络中的周期信息数为 n,则每个周期信息 M_i的模型为:

$$M_i = (C_i, D_i, T_i),\quad i = 1, \cdots, n \quad (3)$$

其中 C_i, D_i和 T_i分别表示周期信息 M_i的执行时间、截止期和周期。由于周期信息一般要求在新数据到达之前,旧的数据必须被取走。因此,周期信息的截止期应该等于其周期,即:$D_i = T_i\ (i = 1, \cdots, n)$。

2.2 RM算法

该算法基于信息的周期为每个周期信息分配固定的优先级:周期越短,优先级越高;信息的释放速率与其周期成反比,而与其优先级成正比,即速率越高,优先级越高。即:

已知周期信息集:$M = \{M_1, M_2, \cdots, M_n\}$, $M_i = (C_i, D_i, T_i)$且 $D_i = T_i\ (i = 1, \cdots, n)$则:

$$T_i \leq T_j \Rightarrow P_i \geq P_j;\quad i = 1, \cdots, n,\ j = 1, \cdots, n \quad (4)$$

其中 P_i表示周期信息 M_i的优先级。

2.3 EDF算法

该算法按照信息的截止期为其分配优先级。周期信息的截止期越小,其优先级越高。信息的优先级在每一个可调度时刻都将动态改变。

定义信息 M_i的调度优先级为:$P_i = D_i(t) - t$, $D_i(t)$为信息在时刻 t的截止期。在每个时刻 t均计算出下个时刻系统中截止期最小的信息,从而优先调度该信息。采用该调度算法系统适应性较强,但工作量较大。

3 MVB实时调度算法

对于具有确定响应时间的 MVB周期信息而言,主要由 BA利用实时调度表来完成其实时调度。根据周期信息的特征,利用相关的实时调度算法,在调度表内建立周期信息传送服务调度。具体运行时,BA通过循环扫描调度表,按照已建立的服务调度次序,依次向各个源设备发送信道使用权。由此可见,MVB周期通信的实时能力主要取决于调度表的内容,而调度表结构的建立方法及采用的调度算法是实时调度的基础。

3.1 实时调度表结构

一般地,实时调度表结构与系统内存等硬件开销紧密相关,其规模主要由周期信息的微周期(microcycle)和宏周期(macrocycle)决定。微周期是调度表的最小调度单位,也是总线仲裁者扫描调度表的最小调度粒度;宏周期是所有周期信息按照一定规律重复出现的最小间隔。文献[10]首先研究了调度表结构,提出了采用 HCF/LCM(最大公约数/最小公倍数)方法建立调度表,即微周期和宏周期分别等于所有信息周期的最大公约数和最小公倍数,调度表就是一个宏周期内要完成调度的全部周期任务的集合。但当信息的周期各不相同且互为质数时,采用 HCF/LCM方法将使调度表的规模变得非常庞大,不仅增加了系统开销,也不便于管理。为此,文献[11]提出了计划调度(planning scheduling)方法,降低了调度表规模,提高了实时性,但同时增加了一定的复杂性。

就 MVB周期信息而言,根据各过程变量的特点及周期应用的时间特性,通过对各变量周期的初步选择,按照下述方法首先采用 HCF/LCM方法确定 MVB实时调度表的规模,接着便可在上述所提周期信息实时模型的基础上,按照一定的算法建立相应的实时调度表。

由于 $D_i = T_i$,则 MVB周期信息模型可描述为:$M_i^* = (C_i, T_i)\ (i = 1, \cdots, n)$,因此,其微周期为:

$$T_{\_bp} = \underset{i=1,\cdots,n}{\text{HCF}}(T_i) = \max\left\{\Omega \;\middle|\; \frac{T_i}{\Omega} = \left\lfloor \frac{T_i}{\Omega} \right\rfloor\right\} \quad (5)$$

且 $\forall i \in [1, n],\ \Omega, i \in N$。

宏周期为:

$$T_{\_Mp} = \underset{i=1,\cdots,n}{\text{LCM}}(T_i) = \min\left\{\phi \;\middle|\; \frac{\phi}{T_i} = \left\lfloor \frac{\phi}{T_i} \right\rfloor\right\} \quad (6)$$

且 $\forall i \in [1, n], \phi, i \in N$。

上述式中 $\lfloor X \rfloor$ 表示取不大于 X 的最大整数。

3.2 实时调度算法的确定

周期信息的实时调度算法根据信息通信方式的不同主要分为异步和同步两种。在异步方式下,总线仲裁者连续处理所有待发送的周期信息,直到所有信息发送完毕。在同步方式下,如果当前微周期能力不够发送某个周期信息,则向后查找新的微周期,直至找到有能力发送此信息的微周期为止,而在每个微周期的剩余时间窗口中可以处理非周期信息传输或插入空闲帧等其他事务。为此,结合经典的 RM 和 EDF 实时调度算法,可产生四种常用的周期信息调度算法:异步 RM、同步 RM、异步 EDF 和同步 EDF。

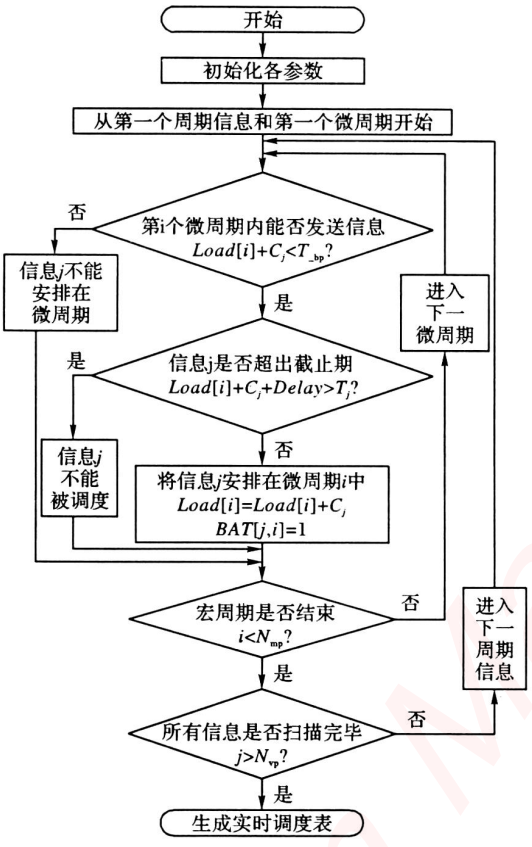

图 3 MVB 实时调度表算法流程

鉴于 MVB 协议中已明确要求在总线传输的每个微周期中除了一定的周期相用以传输周期信息外,尽量保留一定的偶发相以满足非周期信息传输的需求,又考虑到具体应用时,过程数据的循环传输特征周期可根据应用等要求预先初步确定,不必进行优先级的动态调度。为此,假设 MVB 周期信息各自独立,并考虑它们同时到达各自的缓冲区等待发送,即所有周期信息处于临界时刻,此时可采用同步 RM 调度算法对各 MVB 周期信息分配相应的优先级来进行非抢占式的实时调度,建立相应的实时调度表。

该算法具体执行时,首先对实时调度表所有微周期内的负荷初始化为零;然后,从周期最小的信息开始,按照微周期的排列次序,不断地扫描周期信息表,进而找到各信息在宏周期中相应微周期内的位置,从而输出一个代表调度次序的逻辑表。

构建 MVB 实时调度表的算法流程如图 3 所示。其中,$Load[i]$ 表示第 i 个微周期内所有周期信息的负荷,即此微周期内要发送信息的执行时间总和;$BAT[j, i]$ 表示最终生成的实时调度逻辑表:

$$BAT[j, i] = \begin{cases} 1, & \text{周期信息 } j \text{ 能在第 } i \text{ 个微周期内发送} \\ 0, & \text{周期信息 } j \text{ 不能在第 } i \text{ 个微周期内发送} \end{cases}$$

3.3 MVB 实时调度实例

现假设 MVB 总线上有 6 个周期信息需进行通信,其参数如表 1 所示。其中各 MVB 周期信息的执行时间 C_i 按照式(1)、(2)计算所得,优先级高低根据各信息周期从短到长依次排序。

表 1 MVB 周期信息表

周期信息	周期 T_i /ms	信息长度 /Byte	执行时间 C_i /μs	优先级
A	1	4	96	1(最高)
B	2	8	118	2
C	2	32	262	3
D	4	16	166	4
E	8	32	262	5
F	8	8	118	6(最低)

表 2 MVB 周期信息实时调度表

周期信息	微周期							
	1	2	3	4	5	6	7	8
A	1	1	1	1	1	1	1	1
B	1	0	1	0	1	0	1	0
C	1	0	1	0	1	0	1	0
D	1	0	0	0	1	0	0	0
E	1	0	0	0	0	0	0	0
F	0	1	0	0	0	0	0	0

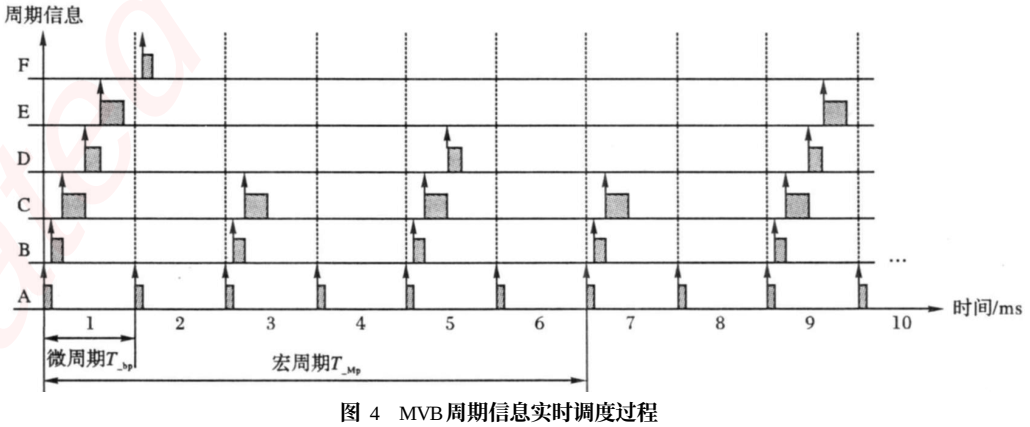

图 4 MVB 周期信息实时调度过程

根据 HCF/LCM 方法,可得出表 1 中各信息的微周期为 $T_{bp} = \underset{i=1,\cdots,6}{\mathrm{HCF}}(T_i) = 1\,\mathrm{ms}$,宏周期 $T_{Mp} = \underset{i=1,\cdots,6}{\mathrm{LCM}}(T_i) = 8\,\mathrm{ms}$,即每个周期性循环调度间隔为 8 个微周期。基于此,采用图 3 所示的同步 RM 调度算法构建的实时调度表如表 2 所示,

由此形成的 MVB 周期信息实时调度过程如图 4所示。图 4中,每个周期信息上的箭头代表了信息的实际发送时刻,而在每个微周期内均假设要发送信息的到达时刻相同。

4 MVB实时算法可调度性分析

无论采用何种算法进行实时调度,在周期信息执行前需预先进行算法可调度性分析,即判定调度算法是否能够满足所有信息的截止期要求。一般地,可采用基于系统利用率和任务响应时间两种方法进行算法可调度性分析[9]。前者虽然计算不够精确,但开销小,可以用于新任务的接入控制;而后者计算开销较大,但可以得到每个任务的响应时间,具有更大的优势。

现采用同步 RM 算法对 MVB 进行实时调度,在每个微周期的剩余时间可能会插入空闲帧以求同步,这样,上述两种目前普遍使用的分析方法都无法直接用于对同步 RM 算法进行算法可调度性分析。近年来 Tovar等学者提出了采用基于任务响应时间的递推式分析方法来判断同步 RM 算法是否满足可调度性[12-15],但由于其计算工作量较大,且结果是在假设各信息执行时间均相等及其他前提条件下得出的,对于 MVB 实际周期信息的调度及应用具有一定的局限性。为此,本文在此基础上,通过求取周期信息的响应时间,再结合 MVB 周期信息实时调度表的建立过程,提出了一种用于分析和判断实时调度有效性的算法,按照 MVB 周期信息从高到低的优先级顺序,循序计算并确定各周期信息的响应时间是否大于其周期 (截止期),从而判断同步 RM 算法是否满足 MVB 周期信息可调度性要求。

不失一般性,仍假设 MVB 周期信息相互独立且处于临界时刻,这样,各信息将可能等待最长响应时间 (Worst-Case Response Time, WCRT)。若采用同步 RM 算法对周期信息进行调度,并根据上述图 3所示算法流程已生成相应的实时调度表,则调度表有效的充分必要条件为:

$$T_{R_i}^{\mathrm{wc}} \leq D_i = T_i \tag{7}$$

其中:$T_{R_i}^{\mathrm{wc}}$ 为 MVB 周期信息 M_i^* 在临界时刻的最长响应时间。

假设 $N_{\mathrm{mp}} = \dfrac{T_{\mathrm{Mp}}}{T_{\mathrm{bp}}}$ 为每个宏周期内所包含的微周期数,考虑到在临界时刻信息 M_i^* 被调度时,有可能存在高优先级信息对 M_i^* 的阻塞。因此,就每个宏周期而言,若 M_i^* 在第 n个微周期内被调度 ($n = 1, \cdots, N_{\mathrm{mp}}$ 为每个宏周期内各微周期的序号),则信息发送及处理的响应时间为:

$$T_{R_i}(n) = [(n-1) \bmod T_i] \cdot T_{\mathrm{bp}} + T_{h_i,n} + C_i \tag{8}$$

$T_{h_i,n}$ 为在第 n个微周期内优先级高于信息 M_i^* 的所有其他信息 $h_i(n)$ 的发送处理时间,其大小为:

$$T_{h_i,n} = \sum_{j \in h_i(n)} C_i \tag{9}$$

因此:

$$T_{R_i}(n) = [(n-1) \bmod T_i] \cdot T_{\mathrm{bp}} + \sum_{j \in h_i(n)} C_j + C_i \tag{10}$$

则在一个宏周期内,周期信息 M_i^* 的最长响应时间为:

$$T_{R_i}^{\mathrm{wc}} = \max_{n \in \{1, \cdots, N_{\mathrm{mp}}\}} \{T_{R_i}(n)\} \tag{11}$$

由式 (8) ～ (11) 可知,判断同步 RM 算法是否可调度的充要条件为:

$$T_{R_i}^{\mathrm{wc}} = \max_{n \in \{1, \cdots, N_{\mathrm{mp}}\}} \{T_{R_i}(n)\} \leq T_i \tag{12}$$

对于表 1所示的 MVB 周期信息,以每个宏周期为单位,采用上述基于响应时间的方法进行同步 RM 算法可调度性判定的过程如下。

对于信息 A,响应时间 $T_{R_A}^{\mathrm{wc}} = C_A = 96\,\mu\mathrm{s} < T_A$;

对于信息 B,响应时间 $T_{R_B}^{\mathrm{wc}} = C_B = 118\,\mu\mathrm{s} < T_B$;

对于信息 C,响应时间 $T_{R_C}^{\mathrm{wc}} = C_C = 262\,\mu\mathrm{s} < T_C$;

对于信息 D,响应时间 $T_{R_D}^{\mathrm{wc}} = C_D = 166\,\mu\mathrm{s} < T_D$;

对于信息 E,响应时间 $T_{R_E}^{\mathrm{wc}} = C_E = 262\,\mu\mathrm{s} < T_E$;

对于信息 F,响应时间 $T_{R_F}^{\mathrm{wc}} = [(2-1) \bmod T_F] \cdot T_{\mathrm{bp}} + \sum_{j \in h_F(2)} C_j + C_F = 1214\,\mu\mathrm{s} < T_F$。

显然,表 1中所有周期信息按照上述方法进行计算后均满足不等式 (12),表明周期信息的最大响应时间均小于各自的截止期,则采用上述实时调度算法生成的实时调度表有效。若部分信息无法满足各自的时限要求,则意味着采用该调度算法不能满足截止期的要求,此时需考虑调整 MVB 信息的到达周期等相关参数以及网络总线结构等其他因素。

5 结语

在综合分析一般现场总线周期信息实时调度算法的基础上,结合 MVB 周期信息的基本通信过程和报文定时特点,研究了采用同步 RM 调度算法来建立 MVB 实时调度表的原理与方法。在此基础上,提出了一种对该实时调度算法进行可调度性分析的具体方法,用以判断构建的实时调度表的有效性,并通过具体的 MVB 周期信息实例阐述了实时调度算法及其可调度性分析方法的具体应用,为实际 MVB 网络的应用研究提供了理论指导。

但本文的研究还有许多方面值得进一步完善和扩展。如相同周期信息的优先级如何确定、各周期信息并非相互独立、各信息的通信存在释放抖动等问题都将是今后继续研究和探讨的内容。

参考文献:

[1] IEC 61375-1, Train Communication Network [S]. 1999.

[2] 朱琴跃,谢维达,谭喜堂,等. 列车通信网性能测试的研究与实现[J]. 测控技术, 2007, 26(2): 60-63.

[3] TOVAR E, VASQUES F. Factory communications: on the configuration of the WorldFIP bus arbitrator table [EB/OL]. [2007-04-03]. http://www.hurray.isep.ipp.pt/asp/show_doc2.asp?id=54.

[4] CHEN J M, WANG Z, SUN Y X. A basic study on algorithm of real-time schedule table for fieldbus [C]// Proceedings of the 4th World Congress on Intelligent Control and Automation. [S.l.]: IEEE Press, 2002, 3: 1760-1763.

[5] CAVALIERI S, STEFANO A D, MIRABELLA O. Pre-run-time scheduling to reduce schedule length in the fieldbus environment [J]. IEEE Transactions on Software Engineering, 1995, 21(11): 865-880.

[6] LIU C L, LAYLAND J W. Scheduling algorithms for multiprogramming in a hard real-time environment [J]. Journal of the ACM, 1973, 20(1): 46-61.

上车辆 2的运动变化。车辆 1行驶 0. 75 s(13. 25 m处)以后,为了不与动态目标位置的后方车辆相撞,增大车头方向与水平方向夹角,以便车辆 1能够较快地变换车道。车辆 1行驶 2 s(32 m处)以后,车辆 1完成车道变换,但尚未赶上车辆 2(已行驶到 35 m处),因此,车辆 1继续前行,车头方向基本保持不变。最终车辆 1与目标方向的差值大约为 0. 015 rad,换算成角度值大约为 0. 859 4 °(小于 1°),在误差允许的范围内。因此,该研究能够较好地模拟驾驶员车道变换的动机,控制方案达到了预期目的。

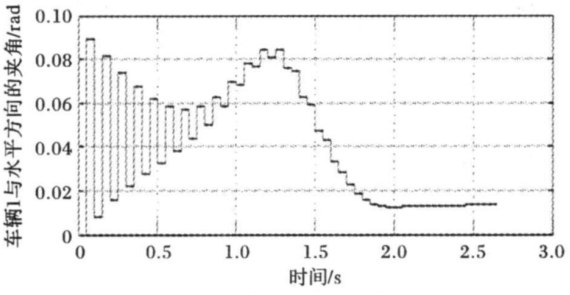

图 6 车辆即时方向变化

5 结语

以智能车辆的横向控制为研究对象,提出了动态目标位置概念作为车辆控制的基础;根据车辆运动特性设计了两层模糊控制器,减少了控制规则的数目,提高了控制器的运行效率;以三次样条曲线作为车道变换的路径拟合曲线,实现车辆的横向智能控制。不仅能够克服基于磁道钉导航方式行驶路线相对不灵活的缺点,而且能够克服设计控制器时采集的数据可能不准确和控制规则固定的缺点。仿真结果表明:被控车辆能够沿着虚拟的路线平稳地跟踪前方车辆,并平滑地变换车道,较为理想地模拟实际交通环境中车辆横向运动的特性和驾驶员的驾驶行为特性。

参考文献:

[1] 周俊,姬长英. 智能车辆横向控制研究[J]. 机器人, 2003, 25(1): 26 - 30.

[2] SHLADOVER S E, DESOVER C A. Automatic vehicle, control developments in the path, program [J]. IEEE Transactions on Vehicular Technology, 1991, 40(1): 114 - 130.

[3] ZHANG J R, XU S J, RACHID A. Sliding mode controller for automatic steering of vehicle [C]// The 27th Annual Conference of the IEEE Industrial Electronics Society. [S. l]: IEEE Press, 2001, 3: 2149 - 2153.

[4] SMITH D E, STARKEY J M, BENTON R E. Nonlinear-gain-optimized controller development and evaluation for automated emergency vehicle steering [C]// Proceedings of the American Control Conference. [S. l]: IEEE Press, 1995, 5: 3586 - 3591.

[5] CHAN C-Y, TAN H-S. Lane tracking in vehicle-following collision situation [C]// Proceedings of the American Control Conference [S. l]: IEEE Press, 1999: 3697 - 3701.

[6] O'BRIEN R T, IGLESIAS P A, URBAN T J. Vehicle lateral control for automated highway systems [J]. Control Systems Technology, 1996, 4(3): 266 - 273.

[7] FENG K-T, TAN H S, TOMIZUKA M. Automatic steering control of vehicle lateral motion with the effect of roll dynamics [C]// Proceedings of the American Control Conference. [S. l]: IEEE Press, 1998: 2248 - 2252.

[8] WANG J Y, TOMIZUKA M. Robust lateral control of heavy - duty vehicles in automated highway system [C]// Proceedings of the American Conference. [S. l]: IEEE Press, 1999: 3671 - 3675.

[9] HESSBURG T, TOMIZUKA M. Fuzzy Logic Control for Lateral Vehicle Guidance[J]. IEEE Control Systems, 1994, 14(4), 55 - 63.

[10] KAMEL A E, DIEULOT J-Y, BORNE P. Fuzzy controller for lateral guidance of busses [C]// Proceedings of the 2002 IEEE International Symposium on Intelligent Control. [S. l]: IEEE Press, 2002: 110 - 115.

[11] 李庆中,顾伟康,等. 移动机器人路径跟踪的智能预瞄控制方法研究[J]. 机器人, 2002. 24(3): 252 - 255.

[12] KEHTARNAVAZ N, CORISWOLD N, MILLER K, *et al* A transportable neural network approach to autonomous vehicle following [J]. IEEE Transaction of Vehicular Technology, 1998, 47(2): 694 - 720.

[13] AMOR M A B, ODA T, WATANABE S. A car-steering model based on an adaptive neuro-fuzzy controller [J]. IEEJ Transaction on Electronics Information and Systems, 2004, 124(11): 2344 - 2351.

[14] RAJU G V S, ZHOU J, ROGER A K. Hierarchical fuzzy control [J]. International Journal of Control, 1991, 54(5): 1201 - 1216.

[15] WI C, WANG L X. A note on universal approximation by hierarchical fuzzy systems [J]. Information Sciences: an International Journal, 2000, 123(3/4): 241 - 248.

(上接第 3111页)

[7] LEHOCZKY J, SHA L, DING Y. The rate monotonic scheduling algorithm: exact characterization and average case behavior [C/OL]// Proceeding of the 11th Real-Time Systems Symposium [2007 - 03 - 02]. http://www.cc.gatech.edu/classes/AY2008/cs4220_fall/5A-RM-Average.pdf.

[8] COTTET F, DELACROIX J. Scheduling in real-time systems [M]. San Francisco: Jossey Bass Academic Publisher, 2002.

[9] LIU J W S. real-time systems[M]. 北京: 清华大学出版社, 2003.

[10] RAJA P. Static and dynamic polling mechanism for fieldbus networks [J]. ACM Operation System Review, 1993, 27(1), 34 - 45.

[11] ALMEIDA L, PASADAS R, FONSECA J. Using a planning scheduler to improve the flexibility of real-time fieldbus networks [J]. Control Engineering Practice, 1999, 7: 101 - 108.

[12] TOVAR E, VASQUES F. A communication support for real-time distributed computer controlled systems [C]// Proceedings of the IEE International Workshop on Discrete Event Systems (WODES'98). Cagliari, Italy: [s n], 1998: 178 - 183.

[13] TOVAR E, VASQUES F. Using WorldFIP networks to support periodic and sporadic real-time traffic [C]// 25th Annual Conference of the IEEE Industrial Electronics Society. [S. l]: IEEE Press, 1999, 3(29): 1216 - 1221.

[14] 王智. 面向现场总线的分布式实时系统的建模与分析方法[D]. 沈阳: 中国科学院沈阳自动化研究所, 2000.

[15] ALMEIDA L, TOVAR E, FONSECA J. Schedulability analysis of real-time traffic in WorldFIP networks: an integrated approach[J]. IEEE Transactions on Industrial Electronics, 2002, 49(5): 1165 - 1174.

嵌入式资源免费下载

总线协议：

1. 基于 PCIe 驱动程序的数据传输卡 DMA 传输
2. 基于 PCIe 总线协议的设备驱动开发
3. CANopen 协议介绍
4. 基于 PXI 总线 RS422 数据通信卡 WDM 驱动程序设计
5. FPGA 实现 PCIe 总线 DMA 设计
6. PCI Express 协议实现与验证
7. VPX 总线技术及其实现
8. 基于 Xilinx FPGA 的 PCIE 接口实现
9. 基于 PCI 总线的 GPS 授时卡设计
10. 基于 CPCI 标准的 6U 信号处理平台的设计
11. USB30 电路保护
12. USB30 协议分析与框架设计
13. USB 30 中的 CRC 校验原理及实现
14. 基于 CPLD 的 UART 设计
15. IPMI 在 VPX 系统中的应用与设计
16. 基于 CPCI 总线的 PMC 载板设计
17. 基于 VPX 总线的工件台运动控制系统研究与开发
18. PCI Express 流控机制的研究与实现
19. UART16C554 的设计
20. 基于 VPX 的高性能计算机设计
21. 基于 CAN 总线技术的嵌入式网关设计
22. Visual C 串行通讯控件使用方法与技巧的研究
23. IEEE1588 精密时钟同步关键技术研究
24. GPS 信号发生器射频模块的一种实现方案
25. 基于 CPCI 接口的视频采集卡的设计
26. 基于 VPX 的 3U 信号处理平台的设计
27. 基于 PCI Express 总线 1394b 网络传输系统 WDM 驱动设计
28. AT89C52 单片机与 ARINC429 航空总线接口设计
29. 基于 CPCI 总线多 DSP 系统的高速主机接口设计
30. 总线协议中的 CRC 及其在 SATA 通信技术中的应用
31. 基于 FPGA 的 SATA 硬盘加解密控制器设计
32. Modbus 协议在串口通讯中的研究及应用
33. 高可用的磁盘阵列 Cache 的设计和实现
34. RAID 阵列中高速 Cache 管理的优化

35. 一种新的基于 RAID 的 CACHE 技术研究与实现
36. 基于 PCIE-104 总线的高速数据接口设计
37. 基于 VPX 标准的 RapidIO 交换和 Flash 存储模块设计
38. 北斗卫星系统在海洋工程中的应用
39. 北斗卫星系统在远洋船舶上应用的研究
40. 基于 CPCI 总线的红外实时信号处理系统
41. 硬件实现 RAID 与软件实现 RAID 的比较
42. 基于 PCI Express 总线系统的热插拔设计
43. 基于 RAID5 的磁盘阵列 Cache 的研究与实现
44. 基于 PCI 总线的 MPEG2 码流播放卡驱动程序开发
45. 基于磁盘异或引擎的 RAID5 小写性能优化
46. 基于 IEEE1588 的时钟同步技术研究
47. 基于 Davinci 平台的 SD 卡读写优化
48. 基于 PCI 总线的图像处理及传输系统的设计
49. 串口和以太网通信技术在油液在线监测系统中的应用
50. USB30 数据传输协议分析及实现
51. IEEE 1588 协议在工业以太网中的实现
52. 基于 USB30 的设备自定义请求实现方法
53. IEEE1588 协议在网络测控系统中的应用
54. USB30 物理层中弹性缓冲的设计与实现
55. USB30 的高速信息传输瓶颈研究
56. 基于 IPv6 的 UDP 通信的实现
57. 一种基于 IPv6 的流媒体传送方案研究与实现
58. 基于 IPv4-IPv6 双栈的 MODBUS-TCP 协议实现
59. RS485CAN 网关设计与实现

VxWorks：

1. 基于 VxWorks 的多任务程序设计
2. 基于 VxWorks 的数据采集存储装置设计
3. Flash 文件系统分析及其在 VxWorks 中的实现
4. VxWorks 多任务编程中的异常研究
5. VxWorks 应用技巧两例
6. 一种基于 VxWorks 的飞行仿真实时管理系统
7. 在 VxWorks 系统中使用 TrueType 字库
8. 基于 FreeType 的 VxWorks 中文显示方案
9. 基于 Tilcon 的 VxWorks 简单动画开发
10. 基于 Tilcon 的某武器显控系统界面设计
11. 基于 Tilcon 的综合导航信息处理装置界面设计

12. VxWorks 的内存配置和管理
13. 基于 VxWorks 系统的 PCI 配置与应用
14. 基于 MPC8270 的 VxWorks BSP 的移植
15. Bootrom 功能改进经验谈
16. 基于 VxWorks 嵌入式系统的中文平台研究与实现
17. VxBus 的 A429 接口驱动
18. 基于 VxBus 和 MPC8569E 千兆网驱动开发和实现
19. 一种基于 vxBus 的 PPC 与 FPGA 高速互联的驱动设计方法
20. 基于 VxBus 的设备驱动开发
21. 基于 VxBus 的驱动程序架构分析
22. 基于 VxBus 的高速数据采集卡驱动程序开发
23. Vxworks 下的冗余 CAN 通讯模块设计
24. WindML 工业平台下开发 S1d13506 驱动及显示功能的实现
25. WindML 中 Mesa 的应用
26. VxWorks 下图形用户界面开发中双缓冲技术应用
27. VxWorks 上的一种 GUI 系统的设计与实现
28. VxWorks 环境下 socket 的实现
29. VxWorks 的 WindML 图形界面程序的框架分析
30. VxWorks 实时操作系统及其在 PC104 下以太网编程的应用
31. 实时操作系统任务调度策略的研究与设计
32. 军事指挥系统中 VxWorks 下汉字显示技术
33. 基于 VxWorks 实时控制系统中文交互界面开发平台
34. 基于 VxWorks 操作系统的 WindML 图形操控界面实现方法
35. 基于 GPU FPGA 芯片原型的 VxWorks 下驱动软件开发
36. VxWorks 下的多串口卡设计
37. VxWorks 内存管理机制的研究
38. T9 输入法在 Tilcon 下的实现
39. 基于 VxWorks 的 WindML 图形界面开发方法
40. 基于 Tilcon 的 IO 控制板可视化测试软件的设计和实现
41. 基于 VxWorks 的通信服务器实时多任务软件设计

Linux：

1. Linux 程序设计第三版及源代码
2. NAND FLASH 文件系统的设计与实现
3. 多通道串行通信设备的 Linux 驱动程序实现
4. Zsh 开发指南-数组
5. 常用 GDB 命令中文速览
6. 嵌入式 C 进阶之道

7. Linux 串口编程实例
8. 基于 Yocto Project 的嵌入式应用设计
9. Android 应用的反编译
10. 基于 Android 行为的加密应用系统研究
11. 嵌入式 Linux 系统移植步步通
12. 嵌入式 CC++语言精华文章集锦
13. 基于 Linux 的高性能服务器端的设计与研究
14. S3C6410 移植 Android 内核
15. Android 开发指南中文版
16. 图解 Linux 操作系统架构设计与实现原理（第二版）
17. 如何在 Ubuntu 和 Linux Mint 下轻松升级 Linux 内核
18. Android 简单 mp3 播放器源码
19. 嵌入式 Linux 系统实时性的研究
20. Android 嵌入式系统架构及内核浅析
21. 基于嵌入式 Linux 操作系统内核实时性的改进方法研究
22. Linux TCP IP 协议详解
23. Linux 桌面环境下内存去重技术的研究与实现
24. 掌握 Android 7.0 新增特性 Quick Settings
25. Android 应用逆向分析方法研究
26. Android 操作系统的课程教学
27. Android 智能手机操作系统的研究
28. Android 英文朗读功能的实现
29. 基于 Yocto 订制嵌入式 Linux 发行版
30. 基于嵌入式 Linux 的网络设备驱动设计与实现
31. 如何高效学习嵌入式
32. 基于 Android 平台的 GPS 定位系统的设计与实现
33. LINUX ARM 下的 USB 驱动开发
34. Linux 下基于 I2C 协议的 RTC 驱动开发
35. 嵌入式下 Linux 系统设备驱动程序的开发
36. 基于嵌入式 Linux 的 SD 卡驱动程序的设计与实现
37. Linux 系统中进程调度策略
38. 嵌入式 Linux 实时性方法
39. 基于实时 Linux 计算机联锁系统实时性分析与改进
40. 基于嵌入式 Linux 下的 USB30 驱动程序开发方法研究
41. Android 手机应用开发之音乐资源播放器
42. Linux 下以太网的 IPv6 隧道技术的实现
43. Research and design of mobile learning platform based on Android
44. 基于 linux 和 Qt 的串口通信调试器调的设计及应用
45. 在 Linux 平台上基于 QT 的动态图像采集系统的设计
46. 基于 Android 平台的医护查房系统的研究与设计
47. 基于 Android 平台的软件自动化监控工具的设计开发
48. 基于 Android 的视频软硬解码及渲染的对比研究与实现

Windows CE:

1. Windows CE.NET 下 YAFFS 文件系统 NAND Flash 驱动程序设计
2. Windows CE 的 CAN 总线驱动程序设计
3. 基于 Windows CE.NET 的 ADC 驱动程序实现与应用的研究
4. 基于 Windows CE.NET 平台的串行通信实现
5. 基于 Windows CE.NET 下的 GPRS 模块的研究与开发
6. win2k 下 NTFS 分区用 ntldr 加载进 dos 源代码
7. Windows 下的 USB 设备驱动程序开发
8. WinCE 的大容量程控数据传输解决方案设计
9. WinCE6.0 安装开发详解
10. DOS 下仿 Windows 的自带计算器程序 C 源码
11. G726 局域网语音通话程序和源代码
12. WinCE 主板加载第三方驱动程序的方法
13. WinCE 下的注册表编辑程序和源代码
14. WinCE 串口通信源代码
15. WINCE 的 SD 卡程序[可实现读写的源码]
16. 基于 WinCE 的 BootLoader 研究
17. Windows CE 环境下无线网卡的自动安装
18. 基于 Windows CE 的可视电话的研究与实现
19. 基于 WinCE 的嵌入式图像采集系统设计
20. 基于 ARM 与 WinCE 的掌纹鉴别系统
21. DCOM 协议在网络冗余环境下的应用
22. Windows XP Embedded 在变电站通信管理机中的应用
23. XPE 在多功能显控台上的开发与应用
24. 基于 Windows XP Embedded 的 LKJ2000 仿真系统设计与实现
25. 虚拟仪器的 Windows XP Embedded 操作系统开发
26. 基于 EVC 的嵌入式导航电子地图设计

PowerPC:

1. Freescale MPC8536 开发板原理图
2. 基于 MPC8548E 的固件设计
3. 基于 MPC8548E 的嵌入式数据处理系统设计
4. 基于 PowerPC 嵌入式网络通信平台的实现

5. PowerPC 在车辆显控系统中的应用
6. 基于 PowerPC 的单板计算机的设计
7. 用 PowerPC860 实现 FPGA 配置
8. 基于 MPC8247 嵌入式电力交换系统的设计与实现
9. 基于设备树的 MPC8247 嵌入式 Linux 系统开发
10. 基于 MPC8313E 嵌入式系统 UBoot 的移植
11. 基于 PowerPC 处理器 SMP 系统的 UBoot 移植
12. 基于 PowerPC 双核处理器嵌入式系统 UBoot 移植
13. 基于 PowerPC 的雷达通用处理机设计
14. PowerPC 平台引导加载程序的移植
15. 基于 PowerPC 嵌入式内核的多串口通信扩展设计
16. 基于 PowerPC 的多网口系统抗干扰设计
17. 基于 MPC860T 与 VxWorks 的图形界面设计
18. 基于 MPC8260 处理器的 PPMC 系统
19. 基于 PowerPC 的控制器研究与设计
20. 基于 PowerPC 的模拟量输入接口扩展
21. 基于 PowerPC 的车载通信系统设计
22. 基于 PowerPC 的嵌入式系统中通用 IO 口的扩展方法
23. 基于 PowerPC440GP 型微控制器的嵌入式系统设计与研究
24. 基于双 PowerPC 7447A 处理器的嵌入式系统硬件设计

ARM：

1. 基于 DiskOnChip 2000 的驱动程序设计及应用
2. 基于 ARM 体系的 PC-104 总线设计
3. 基于 ARM 的嵌入式系统中断处理机制研究
4. 设计 ARM 的中断处理
5. 基于 ARM 的数据采集系统并行总线的驱动设计
6. S3C2410 下的 TFT LCD 驱动源码
7. STM32 SD 卡移植 FATFS 文件系统源码
8. STM32 ADC 多通道源码
9. ARM Linux 在 EP7312 上的移植
10. ARM 经典 300 问
11. 基于 S5PV210 的频谱监测设备嵌入式系统设计与实现
12. Uboot 中 start.S 源码的指令级的详尽解析
13. 基于 ARM9 的嵌入式 Zigbee 网关设计与实现
14. 基于 S3C6410 处理器的嵌入式 Linux 系统移植
15. CortexA8 平台的μC-OSII 及 LwIP 协议栈的移植与实现

16. 基于 ARM 的嵌入式 Linux 无线网卡设备驱动设计
17. ARM S3C2440 Linux ADC 驱动
18. ARM S3C2440 Linux 触摸屏驱动
19. Linux 和 Cortex-A8 的视频处理及数字微波传输系统设计
20. Nand Flash 启动模式下的 Uboot 移植
21. 基于 ARM 处理器的 UART 设计
22. ARM CortexM3 处理器故障的分析与处理
23. ARM 微处理器启动和调试浅析
24. 基于 ARM 系统下映像文件的执行与中断运行机制的实现
25. 中断调用方式的 ARM 二次开发接口设计
26. ARM11 嵌入式系统 Linux 下 LCD 的驱动设计
27. Uboot 在 S3C2440 上的移植
28. 基于 ARM11 的嵌入式无线视频终端的设计
29. 基于 S3C6410 的 Uboot 分析与移植
30. 基于 ARM 嵌入式系统的高保真无损音乐播放器设计
31. UBoot 在 Mini6410 上的移植
32. 基于 ARM11 的嵌入式 Linux NAND FLASH 模拟 U 盘挂载分析与实现

Hardware：

1. DSP 电源的典型设计
2. 高频脉冲电源设计
3. 电源的综合保护设计
4. 任意波形电源的设计
5. 高速 PCB 信号完整性分析及应用
6. DM642 高速图像采集系统的电磁干扰设计
7. 使用 COMExpress Nano 工控板实现 IP 调度设备
8. 基于 COM Express 架构的数据记录仪的设计与实现
9. 基于 COM Express 的信号系统逻辑运算单元设计
10. 基于 COM Express 的回波预处理模块设计
11. 基于 X86 平台的简单多任务内核的分析与实现
12. 基于 UEFI Shell 的 PreOS Application 的开发与研究
13. 基于 UEFI 固件的恶意代码防范技术研究
14. MIPS 架构计算机平台的支持固件研究
15. 基于 UEFI 固件的攻击验证技术研究
16. 基于 UEFI 的 Application 和 Driver 的分析与开发
17. 基于 UEFI 的可信 BIOS 研究与实现
18. 基于 UEFI 的国产计算机平台 BIOS 研究
19. 基于 UEFI 的安全模块设计分析

20. 基于 FPGA NiosⅡ的等精度频率计设计
21. 基于 FPGA 的 SOPC 设计
22. 基于 SOPC 基本信号产生器的设计与实现
23. 基于龙芯平台的 PMON 研究与开发
24. 基于 X86 平台的嵌入式 BIOS 可配置设计
25. 基于龙芯 2F 架构的 PMON 分析与优化
26. CPU 与 GPU 之间接口电路的设计与实现
27. 基于龙芯 1A 平台的 PMON 源码编译和启动分析
28. 基于 PC104 工控机的嵌入式直流监控装置的设计
29. GPGPU 技术研究与发展
30. GPU 实现的高速 FIR 数字滤波算法
31. 一种基于 CPUGPU 异构计算的混合编程模型
32. 面向 OpenCL 模型的 GPU 性能优化

Programming:

1. 计算机软件基础数据结构 - 算法
2. 高级数据结构对算法的优化
3. 零基础学算法
4. Linux 环境下基于 TCP 的 Socket 编程浅析
5. Linux 环境下基于 UDP 的 socket 编程浅析
6. 基于 Socket 的网络编程技术及其实现
7. 数据结构考题 - 第 1 章 绪论
8. 数据结构考题 - 第 2 章 线性表
9. 数据结构考题 - 第 2 章 线性表 - 答案