

基于 AUTOSAR 的 FlexRay 动态段调度

龚志鹏^{1,2}, 陈特放¹, 邹复民³, 陈意军², 李延平²

(1. 中南大学信息科学与工程学院, 湖南长沙 410083; 2. 湖南工程学院电气与信息工程学院, 湖南湘潭 411104;
3. 福建工程学院福建省汽车电子与电驱动技术重点实验室, 福建福州 350108)

摘要: 针对 FlexRay 的动态段调度难题, 提出了一种基于 AUTOSAR 规范的与 FID 分配无关的消息传输周期及其长度确定方法, 建立了基于并行处理器负载均衡调度的 BIP 模型, 设计了一种适合大规模消息调度的周期递增 LPT 算法 PILPT 和一种防止长周期消息占用短周期消息位置的相位保留 LPT 算法 PRLPT, 最后给出了在实际应用环境中 PILPT 算法和 PRLPT 算法的一般性选择依据. 仿真实验结果表明, 不论是对 J1939 消息集还是对消息数量更多、消息长度分布更广的可变消息集, PILPT 算法和 PRLPT 算法在优化比率、负载均衡以及算法速度上均达到了较好的调度性能.

关键词: FlexRay; AUTOSAR; 消息调度; 动态段; 车载通信网络

中图分类号: TP393.1

文献标识码: A

文章编号: 0372-2112 (2015)01-0086-08

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2015.01.014

Research on the Scheduling of FlexRay Dynamic Segment Based on AUTOSAR

GONG Zhi-peng^{1,2}, CHEN Te-fang¹, ZOU Fu-min³, CHEN Yi-jun², LI Yan-ping²

(1. School of Information Science and Engineering, Central South University, Changsha, Hunan 410083, China;

2. College of Electrical and Information Engineering, Hunan Institute of Engineering, Xiangtan, Hunan 411104, China;

3. Fujian Key Laboratory for Automotive Electronics and Electric Drive, Fujian University of Technology, Fuzhou, Fujian 350108, China)

Abstract: Aimed at the scheduling problem of dynamic segment of FlexRay, the formulas both for message transmission period and the length of message frame are put forward firstly which are unrelated to FID assignment based on the AUTOSAR specification, and then a BIP model is built based on the load balancing of parallel multi-processor scheduling model. To solve the large scale message scheduling problem, a period increasing longest-processing-time-first (PILPT) algorithm is proposed as well as a phase reserving LPT (PRLPT) algorithm, which is proposed to prevent the longer period message from occupying the position of the shorter, and a general suggestion for the choice between the two algorithms is given finally. The simulation results show that, whether for the fixed message set of J1939 or variable message set with a large number of messages and wide length distribution, PILPT and PRLPT algorithms both have outstanding performance in optimizing ratio, load balance and the calculating speed.

Key words: FlexRay; AUTOSAR; message scheduling; dynamic segment; in-vehicle communication network

1 引言

FlexRay 是在 TTP 和 BYTEFLIGHT 的基础上发展而来的新一代汽车车载通信网络协议, 兼具时间触发和事件触发的优点, 为汽车电子所急需的高性能线控技术 (x-by-wire) 提供了一个高速率、高可靠、高确定性的实时消息传输网络, 目前仍在不断完善和稳步推进中^[1,2], 已正式被纳入国际标准 ISO10681. 汽车开放系统架构 (AUTOSAR, Automotive Open System Architecture) 也提供了

对 FlexRay 的支持, 为其国际化和大规模应用奠定了良好的基础^[3]. 为方便系统设计, 针对 FlexRay 的通信特别规范, AUTOSAR 要求系统消息的周期关系必须为 2ⁿ 倍, 从而构成一种简单周期系统, 很大程度上降低了静态段的调度难度. 为保证实时性, 基于事件触发的网络控制系统需要复杂的时基校正算法和拥塞控制规则, FlexRay 虽然在这些方面具有明显改进, 但对于偶发消息在动态段的调度仍十分困难^[4,5]. 因此, 基于 AUTOSAR 的 FlexRay 动态段调度研究逐渐得到了国内外的

广泛关注,其中 L Ouedraogo 研究了动态段消息的最坏响应时间^[6],I Park 从减小消息响应时间的角度出发,提出了一种确定 FlexRay 最优通信周期长度的方法^[7],E G Schmidt 为提高传输效率,把多个消息封装起来并得到一个非线性二进制规划模型(NLBIP)并试图用 2 步线性化 BIP 求解^[8],但是该方法会妨碍软件设计的灵活性,同时在实践应用中大规模 BIP 求解往往非常困难.为此,本文拟首先分析偶发消息在动态段的传输特性,建立一个线性二进制规划 BIP 模型,然后针对大规模消息调度中由于 BIP 模型输入规模巨大难以直接求得最优解的问题,拟提出一种基于 LPT^[9]的启发式算法,最后,本文将进一步研究大周期长消息调度问题,使各通信周期负载能快速趋于平衡而实现优化调度,从而构建一套实际可行的 FlexRay 的偶发消息调度方案.

2 问题和模型

2.1 动态段消息调度

如图 1 所示, FlexRay 总线的访问由不断重复的通信周期(FC, FlexRay Cycle)构成,每个通信周期由静态段(SS, Static Segment)、动态段(DS, Dynamic Segment)、符号窗口(SW, Symbol Window)和网络空闲时间(NIT, Network Idle Time)组成^[10].消息传输在静态段和动态段中进行,其中静态段用于传输周期性实时消息,采用 TDMA 的介质访问方式以保证传输确定性;动态段用于传输偶发消息,采用 FTDMA 的介质访问方式以提高传输效率.动态段传输的每一个消息都封装在动态帧里,在一个预先分配的动态 slot 内传输,且对应一个帧标识(FID, Frame Identifiers),帧长根据消息的实际长度确定,但必须是微时隙(MSlot, MiniSlot)的整数倍.当分配的 FID 没有消息传输时,固定占用一个 MSlot.

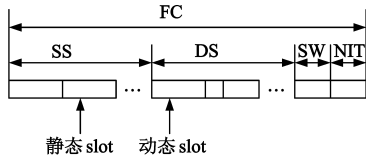


图1 FlexRay通信周期结构

2.1.1 传输周期的确定

偶发消息是传输间隔不确定的实时消息,对偶发消息 m ,其特性可用一个 3 元量(lm, r, d)表示,其中 lm 表示消息长度, r 表示消息间的最小间隔, d 表示截止期,消息必须在截止期前传输.不妨假设规定在下一次消息实例产生前,前一次已经传输完毕,即 $d \leq r$.根据 FlexRay 要求,每一次消息传输都应该预先确定好相关的时间参数,称之为调度机会(SC, Scheduling Chance).消息的第 i 个调度机会可用一个二元量来表示,即 $SC_i = (FC_i, FID_i)$.由于动态段采用了 FTDMA,所

以两个固定 SC 间的间隔并不确定,不妨令 t_i 表示 SC_i 与 SC_{i+1} 之间的间隔.为满足实时要求,必须保证任意相邻两个 SC 间的最大间隔 $t_{i\text{MAX}} \leq d$.设动态段占整个 FC 的比率为 R_{DS} ,并假设不考虑 SW 和 NIT 的影响,则连续两个 SC 间的间隔如图 2 所示,其中 T_C 为一个通信周期时间, T_{SS} 为静态段时间, T_{DS} 为动态段时间, T_{MSlot} 为一个 MSlot 时间.

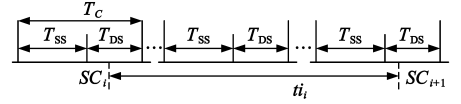


图2 调度机会间隔

假设 m_j^i 表示在调度机会 SC_i 的 FC_i 上预定传输的第 j 个消息,如果确有消息传输,则 $b_j^i = 1$,否则 $b_j^i = 0$.令 lm_j^i 表示消息长度,则相邻调度机会间隔可表示为:

$$\begin{aligned} t_i &= (FC_{i+1} - FC_i - 1) \times T_C + T_{SS} + T_{DS} \\ &\quad - \sum_{j=1}^{FID_i} ((lm_j^i - T_{MSlot}) b_j^i + T_{MSlot}) \\ &\quad + \sum_{j=1}^{FID_{i+1}} ((lm_j^{i+1} - T_{MSlot}) b_j^{i+1} + T_{MSlot}) \\ &\leq (FC_{i+1} - FC_i - 1) \times T_C + T_{SS} + T_{DS} \\ &\quad - FID_i \times T_{MSlot} + \sum_{j=1}^{FID_{i+1}} lm_j^{i+1} \end{aligned} \quad (1)$$

因为偶发消息长度之和不会超过动态段时间 T_{DS} ,为简化调度,可假设传输周期与 FID 分配无关,即 FID 可以随意分配,则有:

$$\begin{aligned} t_i &\leq (FC_{i+1} - FC_i - 1) \times T_C + T_{SS} + T_{DS} + T_{DS} \\ &= (FC_{i+1} - FC_i) \times T_C + T_{DS} \end{aligned} \quad (2)$$

其中, $T_C = T_{SS} + T_{DS}$,令

$$pc = FC_{i+1} - FC_i \quad (3)$$

则有:

$$t_i \leq pc \times T_C + T_{DS} \quad (4)$$

为满足实时需求,必须有:

$$t_i \leq d$$

即

$$pc \leq d/T_C - R_{DS} \quad (5)$$

其中 R_{DS} 为动态段在通信周期中所占比例.根据 AUTOSAR 要求,最终在 FlexRay 上预定的传输周期必须取 2^n ,若用 p 表示,则有:

$$p = 2^{\lfloor \log_2^p \rfloor} \quad (6)$$

例如:对偶发消息消息 m , $d = 100\text{ms} \wedge T_C = 20\text{ms} \wedge R_{DS} = 0.5$,则有:

$$p = 2^{\lfloor \log_2^p \rfloor} = 2^{\lfloor \log_2^{100/20-0.5} \rfloor} = 4$$

2.1.2 消息长度的确定

假设 T_{bit} 为位时间, T_{MSlot} 已经确定,则动态帧长度

可按如下公式计算

$$lm_i = \lceil (ld_i \cdot 16\text{bits} + ld_i \cdot 4\text{bits} + l_0) T_{\text{bit}} / T_{\text{MSlot}} \rceil \cdot T_{\text{MSlot}}$$

$$= \lceil (ld_i \cdot 20\text{bits} + l_0) T_{\text{bit}} / T_{\text{MSlot}} \rceil \cdot T_{\text{MSlot}} \quad (7)$$

其中 $l_0 = TSS + FSS + HEAD + CRC + FES + DTS$, 为其动态帧开销. 在速率为 10M 且无星耦情况下, 可取 $TSS \geq 6 \wedge FSS = 1 \wedge HEAD = 40 \wedge CRC = 24 \wedge FES = 2 \wedge DTS = 2T_{\text{bit}} \sim 1T_{\text{MSlot}} + 2T_{\text{bit}}$, 其中 ld_i 为有效数据长度 (以双字节计, 取值 $0 \sim 127$), 即可算出动态帧的长度分布范围 $[a, b]$, 以 T_{MSlot} 为单位则有:

$$a = \lceil l_0 \times T_{\text{bit}} / T_{\text{MSlot}} \rceil \quad (8)$$

$$b = \lceil (127 \times 20\text{bits} + l_0) \times T_{\text{bit}} / T_{\text{MSlot}} \rceil \quad (9)$$

2.2 动态段调度模型

不妨假设 FlexRay 网络采用一个通道, 且所有消息的长度和预定传输周期都已经根据上节的方法确定, 令将所有消息都至少完成一次完整传输的最少 FC 数称为一个调度周期 (SP, Scheduling Period), FC_i 表示一个 SP 中的第 i 个 FC. 假设需要传输的偶发实时消息集合为 $M_S = \{m_1, m_2, \dots, m_N\}$, 其中 m_i 表示第 i 个消息, 其属性用一个二元量 (p_i, lm_i) 表示, 分别表示消息 m_i 的在 FlexRay 的传输周期和消息长度. 令 L_i 表示 FC_i 的负载, 即分配到 FC_i 的所有消息长度之和; L_A 表示算法 A 的调度长度, 即最大负载长度; L_{OPT} 表示最优调度下的调度长度. 为确定算法 A 的性能, 算法 A 的优化比率定义为:

$$R_A = \frac{L_A}{L_{\text{OPT}}} \quad (10)$$

显然, R_A 越接近 1, 则说明算法 A 的性能越好.

2.2.1 单周期调度模型

为论述方便, 本文首先讨论只有一种传输周期的情形, 即只考虑传输周期都为 P_q 的消息集 $M_q = \{m_1^{P_q}, m_2^{P_q}, \dots, m_{N_q}^{P_q}\}$, 其中第 i 个消息的长度为 $lm_i^{P_q}$. 由于消息以 P_q 为传输周期, 各 FC 的消息分配将以 P_q 为周期循环分布, 故只需考虑开始 P_q 个 FC 的情况, 即 $FC_0 \sim FC_{P_q-1}$. 理想的调度算法是在实现所有偶发消息可行调度的同时, 尽可能节约网络资源, 使动态段的长度最短, 这时可以转换成一个并行处理器调度问题如下: 可假设有 N_q 个任务要分配到有 P_q 个处理器, 第 i 个任务的运行时间 e_i 即为消息长度 $lm_i^{P_q}$, 任务不可抢占, 同一任务不能在多个处理器上并行运行, 要求最后完成时间 C_{max} 最短, 即 $P_{P_q} \parallel npre, e_i = lm_i^{P_q}, reloc = P_q \parallel C_{\text{max}}^{[11]}$, 其中 $reloc$ 表示相对位置约束. 若二进制变量 $x_{i,j}^{P_q}$ 用于消息分配, 即当将消息 $m_i^{P_q}$ 分配给 FC_j 时, $x_{i,j}^{P_q} = 1$, 否则 $x_{i,j}^{P_q} = 0$, $L_j^{P_q}$ 表示消息集 M_q 在 FC_j 产生的负载

大小. 显然, 消息 $m_i^{P_q}$ 在 $FC_{j+n \times P_q}$ 上的贡献相同, 可得 BIP 模型如下:

$$Y_S = \min_x L_{S_{\text{max}}}^{P_q}$$

$$\text{s.t.}$$

$$L_{j+n \times P_q}^{P_q} = \sum_{i=1}^{N_q} x_{i,j}^{P_q} \times lm_i^{P_q} \leq L_{S_{\text{max}}}^{P_q},$$

$$j = 0, 1, \dots, P_q - 1; n = 0, 1, 2, \dots$$

$$\sum_{j=0}^{P_q-1} x_{i,j}^{P_q} = 1, \quad (11)$$

2.2.2 多周期模型

实际应用中往往需要考虑包含有多个传输周期的消息集, 为此设传输周期的集合为 $P_S = \{P_1, P_2, \dots, P_Q\}$, M_q 表示传输周期都为 P_q 的消息集, 则整个动态段消息的集合 $M_S = \bigcup_{q=1}^Q M_q$, L_k 表示 FC_k 的负载, 即各传输周期的消息集在 FC_k 的负载之和, 其 BIP 模型如下:

$$Y_C = \min_x L_{C_{\text{max}}}$$

$$\text{s.t.}$$

$$\forall P_q \in P_S$$

$$L_{j+n \times P_q}^{P_q} = \sum_{i=1}^{N_q} x_{i,j}^{P_q} \times lm_i^{P_q},$$

$$j = 0, 1, \dots, P_q - 1,$$

$$n = 0, 1, 2, \dots$$

$$L_k = \sum_{P_q \in P_S} L_k^{P_q} \leq L_{C_{\text{max}}},$$

$$k = 0, 1, 2, \dots, SP - 1$$

$$\sum_{j=0}^{P_q-1} x_{i,j}^{P_q} = 1 \quad (12)$$

其中第 1, 2 个方程为负载约束, 第 3 个方程为位置约束.

2.2.3 一般计算模型的构建

为构造可求解的计算模型, 假设本问题的线性规划的一般形式如下:

$$\min y = CX$$

$$\text{s.t.}$$

$$BL \leq AX \leq BU$$

$$X \in \{0, 1\}^{NV-1} \times Z \quad (13)$$

其中 NV 为变量总数, C 为目标函数矩阵, BL 和 BU 为约束边界, A 为约束矩阵, 包含负载约束矩阵 A_1 和位置约束矩阵 A_2 , 即

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \quad (14)$$

根据 BIP 模型, 负载约束方程矩阵

$$A_1 = \begin{bmatrix} D_1 & D_2 & & & -1 \\ D_1 & D_2 & D_3 & D_4 & -1 \\ D_1 & D_2 & & D_5 & -1 \\ D_1 & D_2 & D_3 & D_4 & \dots & -1 \\ & & \dots & & & -1 \end{bmatrix} \quad (15)$$

其中 D_i 为消息 m_i 相关对角矩阵,即 $D_i = l_{m_i} \times E_i$, E_i 为适维单位矩阵, l_{m_i} 为消息 m_i 的长度. 同理, 根据 BIP 模型, 位置约束方程

$$A_2 = \begin{bmatrix} I_1 & 0 \\ & I_2 & 0 \\ & & I_3 & 0 \\ & & & \dots & 0 \end{bmatrix} \quad (16)$$

其中 $I_i = 1^{p_i}$, p_i 为消息 m_i 的预定的传输周期. 显然, BIP 模型的输入规模会随着消息数量的增加而急剧增大.

3 算法设计

3.1 周期递增 LPT 算法

对于上节的 BIP 模型, 即使是单周期情形, 当 $P_q \geq 2$ 时, 也是一个 NP 难问题^[12]. 所以当输入规模较小时, BIP 模型尚可以求得最优解, 但当输入规模较大时, 一般难以求解. 实际车辆有上百个节点, 产生超过 2500 个消息, 会生成十万个以上变量, 基本上无法求解, 必须通过启发式算法求得近似解^[13]. 单周期调度是一种典型的并行处理器模型, 有些学者提出了一些启发式算法, 除文献[9]提出的 LPT 外, E G Coffman 提出了基于装箱理论的 MULTIFIT 算法^[14], D S Hochbaum 基于对偶近似提出 PTAS 算法^[15], G Chiaselotti 提出基于部分解求和的 SPS 算法^[16]等. 这些算法都有优异性能, 但对于多周期模型, 这些算法并不能直接应用. 为此, 本文在 LPT 算法基础上, 提出了一种遵守 AUTOSAR 规范的周期递增 LPT 算法 (PILPT, Period Increasing Longest-Processing-Time Algorithm).

算法 1 PILPT 算法

假设 $LPT(machine, L_{ini}, M_q)$ 为标准的 LPT 算法, $machine$ 为并行处理器数, L_{ini} 为初始负载, M_q 为消息集.

step1 将所有消息分类排序

① 将所有消息按周期分类排序, 得 $M_S = \{M_1, M_2, \dots, M_Q\}$, 其中子集 M_q 是周期为 P_q 的消息的集合, 且 $P_q < P_{q+1}$;

② 将所有子集的消息按长度递减排序, 第 q 个消息集 $M_q = \{m_1^q, m_2^q, \dots, m_{N_q}^q\}$, 其中 $l_{m_i^q} \geq l_{m_{i+1}^q}$;

setp2 $FC = 0$; 各 FC 负载初始化为 0

setp3 对各子集执行标准 LPT 算法

for q from 1 to Q 从最短周期消息集开始

$machine = P_q$;

$L_{ini} = FC(1: P_q)$; 负载初始化

$L = LPT(machine, L_{ini}, M_q)$;

$FC(j + n \times P_q) = L(j)$; 更新所有 FC 的负载

end for

定理 1 对采用 AUTOSAR 周期规范的 FlexRay 网络, 对单周期消息集 M_S , 令所有消息的传输周期都为 p 且 $p \geq 2$, 最优调度长度为 L_{OPT} , 设调度周期为 SP , 如果采用 PILPT 调度算法, 则

$$R_{PILPT} = \frac{L_{PILPT}}{L_{OPT}} \leq \frac{4}{3} - \frac{1}{3p} \quad (17)$$

当每个 FC 至少有 k 个消息时

$$R_{PILPT} = \frac{L_{PILPT}}{L_{OPT}} \leq \frac{k+1}{k} - \frac{1}{k \times p} \quad (18)$$

证明 当只有一个周期 p 时, 只需考虑前 p 个 FC 的负载分配, 其余 FC 的负载以 p 为周期循环, 故完全可以等价于一个有 p 个相同并行处理器的负载均衡调度问题, 并采用了标准 LPT 调度算法^[17], 故定理成立.

定理 2 对采用 AUTOSAR 周期规范的 FlexRay 网络, 当调度周期为 SP 且消息集 M_S 具有的周期数量 $Q \geq 2$, 令最优调度长度为 L_{OPT} , 如果采用 PILPT 调度算法, 则

$$R_{PILPT} = \frac{L_{PILPT}}{L_{OPT}} \leq 2 - \frac{1}{SP} \quad (19)$$

证明 因为采用了 PILPT 算法, 即从周期最小的消息开始调度, 每一个新来消息都能分配给当前负载最小的 FC, 因为: ① 对于同一种周期的消息, 在当前周期的所有的相位都能自由分配消息, ② 对大周期消息分配时, 由于采用 AUTOSAR 周期规范, 有 $P_{q+1} = 2^n \times P_q$, 新来消息能在所有小周期的相位中自由分配消息. 故在最后一个消息分配前, 所有 FC 都没有空闲, 其优化比率应该小于或等于 SP 个相同并行处理器负载平衡调度问题的 LS 算法^[18], 定理成立.

定理 3 对采用 AUTOSAR 周期规范的 FlexRay 网络, 当调度周期为 SP 且消息集 $M_S = \{m_1, m_2, \dots, m_N\}$, 第 i 个消息 m_i 的属性用二元量 (p_i, l_{m_i}) 表示, 假设消息长度分布为 $[a, b]$, 最优调度长度 L_{OPT} , 如果采用 PILPT 调度算法, 当消息数量增大时, 优化比率趋向于 1, 即 $R_{PILPT} \rightarrow 1$.

证明 对消息 m_i , 其在一个调度周期 SP 内产生的消息实例数量为 $f_i = SP/p_i$, 则 FC 的平均负载为

$$\bar{L} = \frac{1}{SP} \sum_{i=1}^N f_i l_{m_i} \leq L_{OPT} \quad (20)$$

根据 PILPT 算法, 在最后一个消息分配前, 所有 FC 都没有空闲, 故在最坏情况下的调度如图 3 所示, 即分配最后一个长度为 b 的消息前, 所有 FC 负载都相同.

最坏情况下, PILPT 的调度长度为

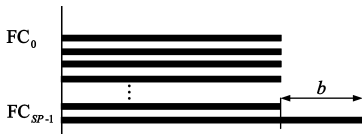


图3 PILPT调度最坏情况

$$L_{\text{PILPT}} \leq \frac{\sum f_i l m_i - b}{SP} + b \quad (21)$$

则优化比率为

$$\begin{aligned} R_{\text{PILPT}} &= \frac{L_{\text{PILPT}}}{L_{\text{OPT}}} \leq \frac{L_{\text{PILPT}}}{L} \leq \frac{\frac{\sum f_i l m_i - b}{SP} + b}{\frac{1}{SP} \sum f_i l m_i} \\ &= 1 + \frac{(SP - 1)b}{\sum f_i l m_i} \end{aligned} \quad (22)$$

当消息数量增大时, $\sum f_i l m_i$ 增大, 即有 $R_{\text{PILPT}} \rightarrow 1$.

3.2 相位保留 LPT 算法

在采用 PILPT 调度算法时, 有可能碰到大周期的长消息在最后才分配, 这样就无法实现 LPT 算法中利用短消息来填补平衡的优化措施, 但如果先分配长周期消息, 则在后面分配短周期消息时容易出现空闲. 为了防止长周期消息占用短周期消息位置, 本文提出了一种基于相位保留的 LPT 算法 (PRLPT, Phase Reserving LPT).

算法 2 PRLPT 算法

```

step1 将所有消息分类排序
      将所有消息首先按长度递减排序, 然后按周期递增排序, 得新的消息集  $M_S = \{m_1, m_2, \dots, m_N\}$ , 其中  $l m_i \geq l m_{i+1}$ , 如果  $l m_i = l m_{i+1}$ , 则  $p_i \leq p_{i+1}$ ;
setp2  $FC = 0$ ; 各 FC 负载初始化为 0
setp3 对  $M_S$  中消息进行 PRLPT 调度
for i from 1 to N 从长度最大, 周期最短的消息开始
    ① 计算当前周期  $p_i$  中, 各相位的最大值和平均值
        for j from 1 to  $p_i$ 
             $maxphase(j) = \max(FC(J))$ , 其中  $J = \{j + n \times p_i\}$ 
             $meanphase(j) = \text{mean}(FC(J))$ , 其中  $J = \{j + n \times p_i\}$ 
        end for
    ② 计算可填充相位并排序
         $phase = 0$ ;
        if 相位平均值和最大值都相等,
            则从 0 相位开始重新分配
        else if 如果各相位最大值不相等,
            则标记最小值相位,  $phase(K) = 1$ ,  $K$  为最大值最小相位的集合
            调用相位保留函数  $pr(p_i, phase)$ , 得填充位置
        else 平均值不相等
            则标记平均值最小值相位,  $phase(K) = 1$ 
            运行相位保留函数  $pr(p_i, phase)$ , 得填充位置
    ③ 根据保留相位函数提供的位置, 分配消息  $m_i$ 

```

end for

其中, 相位保留函数 $position = pr(period, phase)$ 实现如下:

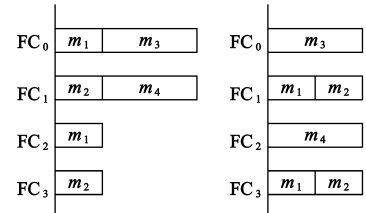
算法 3

```

step1  $p = period, ph = phase$ ; 初始赋值
step2 按顺序找出可填充位置
    while  $p > 1$ 
         $K = \text{bitxor}(ph(1:p/2), ph(p/2+1:p))$ ; 找出不能构成低周期的相位集合  $K$ 
         $position = position \cup \{(p, K)\}$ ; 依次加入可填充位置中
         $ph(K) = 0$ , 清除
         $p = p/2$ , 处理低周期
         $ph = ph(1:p)$ 
    end while

```

例如, 假设有 FlexRay 网络, 其 $SP = 4 \wedge$ 消息集 $M_S = \{m_1, m_2, m_3, m_4\} \wedge p_1 = p_2 = 2 \wedge p_3 = p_4 = 4 \wedge l m_1 = l m_2 = 2 \wedge l m_3 = l m_4 = 4$, 如果采用 PILPT, 结果如图 4(a) 所示, 调度长度为 $L_{\text{PILPT}} = 6$; 而如果采用 PRLPT 调度或最优调度, 则结果如图 4(b) 所示, 调度长度为 $L_{\text{PRLPT}} = 4$.



(a) PILPT 调度 (b) 最优调度与 PILPT 调度
图4 采用3种方法的调度结果

4 仿真实验及其分析

仿真实验中机器配置为 Intel I3 2.3G, RAM 4G, 软件运行环境为 MATLAB2011a, BIP 规划工具为 TOMLAB/CPLEX.

首先, 为验证本文方案的可用性, 故尝试将现行汽车主流总线标准 SAE J1939 (以下简称 J1939) 的消息集移植到 FlexRay 总线. J1939 是美国汽车工程师协会 (SAE, Society of Automotive Engineers) 提出的基于 CAN2.0 车载网络的标准, 汇集了车辆运行的各类参数信号, 并根据各信号类型和所属节点等特征, 对信号进行打包成消息 (即 J1939 中的参数组). 对 J1939 中的 200 个消息进一步整理如表 1 所示, 其中 pm 表示消息周期 (未标明消息周期的为非周期消息), $length$ 为消息长度, $number$ 为对应类型消息数量.

表 1 J1939 消息集

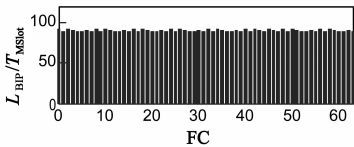
pm (ms)	10	20	50	100	250	500	1000	5000	5000	5000	-	-	-	-	-	
$length$ (bytes)	8	8	8	8	8	8	8	8	19	28	8	10	12	16	20	22
$number$	3	1	13	32	3	12	50	7	1	1	69	1	1	3	1	2

不妨假设对 J1939 中的 200 个实时消息都采用动态段进行调度,且设 FlexRay 网络速率为 $B = 10\text{Mbps} \wedge SP = 64$,即传输周期的集合 $P_S = \{1, 2, 4, 8, 16, 32, 64\}$; 设 $T_C = 5\text{ms}$,动态段占通信周期 FC 一半,即 $T_{DS} = 2.5\text{ms}$,微时隙长度 $T_{Mslot} = 40T_{bit}$;对周期信号,假设截止期等于消息周期,对非周期消息要求在 50ms 内响应,依式(6)可得各消息的传输周期 p . 如果 p 超过 $64T_C$,则以 $64T_C$ 为传输周期. 同时假设动态帧开销 $l_0 = 90\text{bits}$,根据式(7)对各消息进行封装后,得到消息帧长度 lm ,规范后的 J1939 消息集如表 2 所示.

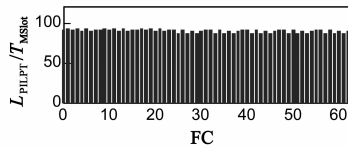
表 2 J1939 规范消息集

$p(T_C)$	1	2	8	8	8	8	16	32	64	64	64
$lm(T_{Mslot})$	5	5	5	6	7	8	5	5	5	7	10
$number$	3	1	83	1	3	3	32	3	69	1	1

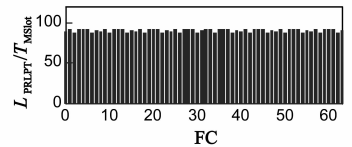
根据式(12)得到最优 BIP 模型,并采用 TOMLAB/CPLEX 工具进行规划计算. BIP 最优规划结果显示,调度长度即动态段所需的最小长度为 $92T_{Mslot}$. 依据仿真



(a) BIP最优规划



(b) PILPT算法



(c) PRLPT 算法

图 5 J1939消息集调度负载均衡

考虑到 J1939 消息长度比较单一,且消息数量有限,因而在最优规划时较易求解. 而 FlexRay 支持更广泛的消息长度分布,消息数量也更多,为进一步考察本文方案对大规模消息集的调度效果,本文不妨构造一个可变消息集 M_S . 在消息集 M_S 中,假设传输周期和消息长度已经根据式(6)和式(7)规范化,每个周期都包含 np 个消息,其长度在 $[a, b]$ 间随机分布, np 可以设置,则总的消息数量为 $N = np \times |P_S|$. 同样设 $T_{Mslot} = 40T_{bit} \wedge l_0 = 90T_{bit}$,则有 $a = 3 \wedge b = 66$. 实验中,依次采用 BIP 最优模型、PILPT 算法和 PRLPT 算法进行调度. 当 BIP 模型无法求解或求解时间过长时,最优值采用如下公式估计

$$L_{EV} = \max(\bar{L}, \max(M_S) \leq L_{OPT}) \quad (23)$$

为了比较所提出算法的性能,本文还设计了遗传算法(GA)同时进行消息调度,GA 算法具有较强的全局搜索能力,常用于各类规划^[19]. GA 算法设置的种群数

调度结果, J1939 消息集消耗的带宽占总带宽的比例为

$$B_M = \frac{L_{BIP} \times T_{Mslot}}{T_C \times B} = \frac{92 \times 40}{5 \times 10^{-3} \times 1.0 \times 10^7} = 7.36\%$$

显然, J1939 消息集只占用了 FlexRay 总线带宽的很小一部分. 实验中,如果假设非周期信号采用传输周期 $p = 64T_C$,则动态段最小长度为 $50T_{Mslot}$,占用带宽可进一步减少到 4%. 可见, FlexRay 总线还保留了很大余量,完全满足现代车辆不断增加的消息的传输需求. 同时,本实验采用了 PILPT 算法和 PRLPT 算法对 J1939 消息集进行调度,结果显示 PILPT 算法和 PRLPT 算法的调度长度都为 $93T_{Mslot}$, 只比 BIP 最优规划多 $1T_{Mslot}$,说明采用 PILPT 算法和 PRLPT 算法均获得了较好的近似解. 图 5 为在上述算法下,各 FC 的负载分配情况. 显然,采用 BIP 最优规划时,各 FC 负载分配最为均衡. 当采用 PILPT 算法和 PRLPT 算法时,最小负载都为 $88T_{Mslot}$,与对应的调度长度只相差 $5T_{Mslot}$,此时各 FC 负载与调度长度差都小于或等于最小消息长度,各 FC 的负载分配比较均衡.

量为 40,代沟为 0.9,最大遗传代数 500,并将所得结果同时给出. 表 3 列出了 $np = 1, 2, 5, 10, 100, 1000, 10000$ 时 4 种算法的调度长度和运行时间. 实验结果显示,在较少消息时,采用 BIP 最优模型优化较快,且能取得最优解,但在 $np = 5$ 即 $|M_S| = 30$ 时,所需时间即达 326s, $np = 6$ 时运行 24h 未求出最优解放弃. 实验结果表明,当消息数稍微增加,输入规模会急剧增大, BIP 模型很快就难于求解,这时采用本文基于 LPT 的启发式算法可以快速获得较满意的近似解,与估计值非常接近,且优于 GA 算法.

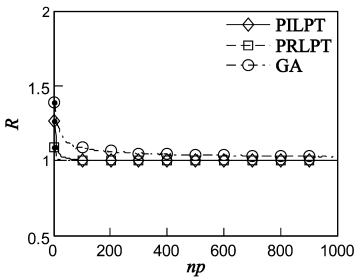
图 6 为 np 从 1 到 1000 依次增大时,采用本文调度算法与 GA 算法的调度结果比较. 图 6(a)为各算法的优化比率,当 np 增加时,优化比率都趋近于 1,但 PILPT 和 PRLPT 的性能明显优于 GA. 实验中结果显示,在最坏情况下, $R_{PILPT} = 1.5397$, $R_{PRLPT} = 1.1887$, $R_{GA} = 1.6346$. 图 6(b)为各算法所得调度长度与 L_{EV} 的绝对误

差,由实验结果可以看出,PRLPT很快接近 L_{EV} ,表现了比 PILPT 相比更好的性能,而 GA 的绝对误差有越来越大的趋势.图 6(c)为各算法所需时间,BIP 最优规划所

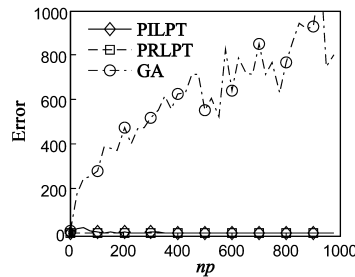
需时间随 np 增大急剧增加,很快无法求解,GA 算法也会随 np 增大,而 PILPT 和 PRLPT 所需时间很少,PRLPT 略大,但都不超过 2s,具有较高的效率.

表 3 $np=1,2,5,10,100,1000,10000$ 的优化结果

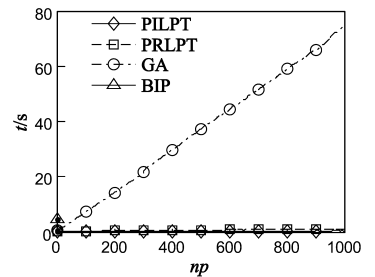
np	L_{BIP}	$t_{BIP}(s)$	L_{EV}	L_{PILPT}	$t_{PILPT}(s)$	L_{PRLPT}	$t_{PRLPT}(s)$	L_{GA}	$t_{GA}(s)$
1	62	4.582	62	62	0.053	62	0.325	74	0.471
2	64	0.131	52	84	0.002	64	0.030	85	0.313
5	159	326.8	157	197	0.003	165	0.021	213	0.499
10	-	-	351	393	0.004	358	0.037	437	0.816
100	-	-	3473	3477	0.039	3475	0.265	3766	7.514
1000	-	-	34883	34884	0.388	34885	1.024	35861	74.05
10000	-	-	349451	349453	4.125	349451	7.340	352594	777.5



(a) 优化比率



(b) 调度长度的绝对误差



(c) 算法速度

图 6 $np=1$ 到1000时调度结果

实验结果同时表明,当 $np \leq 200$ 时,PRLPT 算法能使所有长度较大的消息先调度,性能要优于 PILPT;当 $np > 200$ 时,PILPT 算法中利用短消息进行负载均衡的优势体现出来了,优化比率与 PRLPT 近似,有时甚至要优于 PRLPT,且时间更短,这时采用 PILPT 是一个更好的选择.

5 结语

针对 FlexRay 的动态段调度难题,本文首先通过基于 AUTOSAR 规范的偶发消息动态段传输特性分析,得到了与 FID 分配无关的预定传输消息的周期和长度通用计算方法,然后建立了基于并行处理器负载均衡调度的 BIP 模型.为进一步解决 BIP 模型在实际大规模求解中的困难,本文设计了一种周期递增 LPT 算法 PILPT,理论分析表明,在大规模消息调度中 PILPT 的优化比率趋近于 1.但由于消息中存在多种周期,为防止长周期消息占用短周期消息位置,本文进一步设计了一种相位保留 LPT 算法 PRLPT,使得在优先调度大周期的长消息时,尽可能少地影响后面短周期消息的调度,使各 FC 负载差很快减少到最小消息长度,尽早达到负载均衡.最后仿真实验表明,不论是对 J1939 消息集还是对消息数量更多、消息长度分布更广的可变消息集,PILPT 算法和 PRLPT 算法在优化比率、负载均衡以及算

法速度上均达到了较好的调度性能,并根据仿真实验结果最后给出了在实际应用环境中 PILPT 算法和 PRLPT 算法的一般性选择依据,构建了一套实际可行的 FlexRay 的偶发消息调度方案.

参考文献

- [1] TTTech. Time-Triggered Protocol TTP/C High-Level Specification Document, Protocol Version 1.1[OL]. <http://www.ttttech.com>, 2003.
- [2] F Scheler, W S Preikschat. The real-time systems compiler: migrating event-triggered systems to time-triggered systems, software: practice and experience[J]. Software Practice and Experience, 2011, 41(12): 1491 - 515.
- [3] AUTOSAR. Specification of FlexRay Network Management [OL]. <http://www.autosar.org>, 2011.
- [4] 叶 ■, 曾浩, 田书林, 等. 一种高速并行采样实时校正方法研究[J]. 电子学报, 2013, 41(1): 166 - 170.
Ye Peng, Zeng Hao, Tian Shu-lin, et al. Research on the real-time calibration technology for high-speed parallel sampling [J]. Acta Electronica Sinica, 2013, 41(1): 166 - 170. (in Chinese)
- [5] 罗成, 谢维信. 传感器网络拥塞避免与控制的模糊 AQM 算法[J]. 电子学报, 2014, 42(4): 679 - 684.
Luo Cheng, Xie Wei-xin. Fuzzy AQM for congestion avoidance

- and control in sensor networks[J]. Acta Electronica Sinica, 2014, 42(4): 679 – 684. (in Chinese)
- [6] L Ouedraogo, M ANatick, R Kumar. Computation of the precise worst-case response time of FlexRay dynamic messages [J]. IEEE Transactions on Automation Science and Engineering, 2014, 11(2): 1545 – 5955.
- [7] I Park, H Univ, M Sunwoo. FlexRay network parameter optimization method for automotive applications[J]. IEEE Transactions on Industrial Electronics, 2011, 58(4): 1449 – 1459.
- [8] E G Schmidt, K Schmidt. Message scheduling for the FlexRay protocol: the dynamic segment [J]. IEEE Transactions on Vehicular Technology, 2009, 58(5): 2160 – 2169.
- [9] R L Graham. Bounds on multiprocessing timing anomalies[J]. SIAM Journal on Applied Mathematics, 1969, 17(2): 416 – 429.
- [10] FlexRay Consortium. FlexRay Communication Systems Protocol Specification, Version 3. 0. 1 [OL]. <http://www.FlexRay.com>, 2010.
- [11] M Pinedo. Scheduling Theory, Algorithms Systems (4th Edition)[M]. New York: Springer, 2012. 13 – 33.
- [12] M R Gracey, D S Johnson. Computers and Intractability: A Guild to the Theory of NP-Completeness[M]. New York: W H Freeman and Company, 1978. 221 – 223.
- [13] S Shreejith, S A Fahmy, M Lukaszewycz, et al. Reconfigurable computing in next-generation automotive networks [J]. IEEE Transactions on Embedded Systems Letters, 2013, 5(1): 12 – 15.
- [14] E G Coffman, M R Garey. An application of bin-packing to multi-processor scheduling [J]. SIAM Journal on Computing, 1978, 7(1): 1 – 17.
- [15] D S Hochbaum, D B Shmoys. Using dual approximation algorithms for scheduling problems: theoretical and practical results [J]. Journal of the ACM, 1987, 34(1): 144 – 162.
- [16] G Chiaselotti, M Italia. Minimizing the makespan in nonpre-emptive parallel machinescheduling problem [J]. Journal of Mathematical Modeling and Algorithms, 2010, 9(1): 39 – 51.
- [17] E G Coffman, J R Sethi. A generalized bound on LPT sequencing [A]. Proceedings of ACM on Computer Performance Modeling Measurement and Evaluation [C]. New York: ACM, 1976. 306 – 310.
- [18] R Graham. Bounds for certain multiprocessing anomalies [J]. Bell System Technical Journal, 1966, 45(9): 1563 – 1581.
- [19] 郝海青, 马航, 庄健. 新型复杂进化全局优化算法的研究 [J]. 电子学报, 2013, 41(4): 704 – 709.
Hao Hai-qing, Ma Hang, Zhuang Jian. The study of a new complex system evolutionary algorithm for global optimal problems [J]. Acta Electronica Sinica, 2013, 41(4): 704 – 709. (in Chinese)

作者简介



龚志鹏 男, 1974 年 9 月, 湖南桃江人, 博士研究生, 讲师, 研究方向为交通信息工程及控制、车载通信网络。

E-mail: zpgong@2000@126.com



陈特放 男, 1957 年 5 月出生, 湖南涟源人, 博士生导师、教授, 研究方向为交通信息工程及控制。

E-mail: cticyt@163.com