

基于 FlexRay 静态段快速最优调度算法

龚志鹏^{*①②} 陈特放^① 邹复民^③ 陈意军^② 陈军根^②

^①(中南大学信息科学与工程学院 长沙 410083)

^②(湖南工程学院电气与信息工程学院 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 福州 350108)

摘要: FlexRay 正成为新一代车载通信网络。为解决 FlexRay 静态段调度(FSSS)的帧标志(FID)分配难题, 该文提出一种基于周期特征的自动模型系数矩阵生成(AMCMG)算法, 在大规模 FSSS 时, 可快速得到最优调度模型的各种消息调度属性并确定系统所需的最少 FID 数; 为进一步确定消息相位, 并最终得到完整的周期调度表, 根据不同周期消息之间调度的兼容性, 提出了一种可实现最优规划的基于相位保留规则 FID 分配(PRFIDA)算法; 最后, 仿真实验结果表明 AMCMG 算法能快速正确地建立调度模型, 同时 PRFIDA 算法可以实现消息在已知调度属性时的 FID 最优分配。

关键词: 车载通信网络; FlexRay; 静态段; 帧标志分配; 相位保留

中图分类号: TN915

文献标识码: A

文章编号: 1009-5896(2015)05-1200-07

DOI: 10.11999/JEIT140933

A Fast Optimal Scheduling Algorithm for FlexRay Static Segment

Gong Zhi-peng^{①②} Chen Te-fang^① Zou Fu-min^③ Chen Yi-jun^② Chen Jun-gen^②

^①(School of Information Science and Engineering, Central South University, Changsha 410083, China)

^②(College of Electrical and Information Engineering, Hunan Institute of Engineering, Xiangtan 411104, China)

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

Abstract: FlexRay is becoming the in-vehicle communication network of the next generation. To resolve the problem of Frame IDentification (FID) assignment in the FlexRay Static Segment Scheduling (FSSS), an Automatic Model Coefficient Matrix Generating (AMCMG) algorithm is proposed to obtain the coefficient matrix automatically based on the characteristics of period distribution. A large-scale programming model of FSSS can be generated automatically, and the scheduling properties of all kinds of messages can be derived as well as the minimum number of FID required for the system can be determined quickly. To assign the phase for each message and obtain the complete scheduling table, a Phase Reserving based FID Assignment (PRFIDA) algorithm is designed according to the compatibility of messages scheduling in different periods, which is able to keep the optimal property of the previous programming. Finally, the simulation results demonstrate that the AMCMG algorithm can build the scheduling model rapidly and correctly, and the PRFIDA algorithm can realize the FID assignment optimally based on the known scheduling properties of messages.

Key words: In-vehicle communication network; FlexRay; Static segment; Frame IDentification (FID) assignment; Phase reserving

1 引言

随着汽车技术发展, 整车电子电气架构越来越复杂庞大, 为支持日益增多的车载电子控制单元

(Electronic Control Unit, ECU)之间实时信息交换的需求, 迫切需要一个高速、高可靠、高确定性的网络连接^[1,2]。新一代车载通信网络 FlexRay 总线基于 TTP 协议和 Byteflight 协议, 由戴姆勒-克莱斯勒公司推出后得到了广泛的关注^[3,4]。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^[5,6], 而 FlexRay 总线兼具时间触发和事件触发的特点, 通过其全局同步时钟可预

2014-07-15 收到, 2014-12-17 改回

国家自然科学基金(60674003), 福建省重大专项(2013HZ0002-1), 福建省杰出青年基金(2012J06015), 湖南省教育厅高校创新基金(12K127)和湖南省科技计划项目(2012GK3078)资助课题

*通信作者: 龚志鹏 zpgong2000@126.com

知传感器的数据采集和执行部件的动作时刻，使系统在建模时可以忽略数据的传输过程，大幅提升了系统性能，其各项指标全面优于原有的CAN总线^[7]。但是，FlexRay总线的进展目前依旧较慢，只有部分厂商在少量车型的个别ECU节点采用FlexRay，如宝马X5的悬挂系统等，对基于众多ECU的有大量消息交换的整车广泛FlexRay连接尚未见报道。FlexRay协议的复杂性是影响其应用的一个主要因素，FlexRay静态段的消息调度类似于多核计算机系统的并行调度问题，但却有更复杂的约束条件，使得调度更难实现^[8]，是目前国内外研究热点和难点之一。其中文献[9, 10]对静态段在线和离线调度的时序研究，得出了最坏响应时间计算方法，文献[11]对静态段传输可靠性进行了分析，估计了消息的最小重传次数，文献[12]讨论了静态段数据封装对带宽利用率的影响和最优静态时隙长度，文献[13]则提出了静态段调度的混合遗传算法，但当消息多时由于基因链增长导致调度效率不高，且只能得到近似解，文献[14]提出一种2维装箱算法，但规划规模随消息增多而增大，最终也只能求助于启发式算法，而文献[15]基于数论的方法就静态段调度提出了一种消息分类调度模型，这种模型的输入规模与消息数量无关，可以较好地确定消息调度的最优帧标识(Frame IDentification, FID)数，但面临系数输入工作量大、模型生成困难、应用不够灵活等问题，而且优化结果并不能明确每一消息的具体分配。为此，本文针对该问题拟设计一种应对消息数量多、周期分布广的大规模FlexRay静态段最优调度实现算法，并通过采用基于周期特性的大型优化问题系数树生成方法，使调度模型能够自动生成，同时将提出一种基于相位保留规则的最优FID分配方法以完整实现FlexRay静态段的调度，有望为基于FlexRay的大规模嵌入式汽车应用提供一种实际可行的最优解决方案。

2 静态段调度模型及问题描述

FlexRay的数据通信是由不断重复的长度为TC(Time of Cycle)的基本通信周期(FlexRay Cycle, FC)构成，一个FC可以分为静态段、动态段、符号窗口和空闲时间^[16]。其中，静态段主要用于硬实时性、高度确定性且安全关键的周期性数据传输，如闭环控制系统中的传感器采集数据，控制器输出数据等^[17]。FlexRay静态段采用TDMA访问方式，其

消息传输如图1所示，一个静态段由多个等长的时隙(SLOT)构成，对于每个周期消息，其周期必须是TC的整数倍，且要求每个节点都保有一张调度表，消息只有在属于自己的SLOT到来时才能发送，每个SLOT只能传输一个数据帧并对应一个帧标记FID。FlexRay支持SLOT复用，即一个SLOT可以用于不同的消息，但必须保证不会冲突。为表述方便，下文描述周期时都以TC为单位。

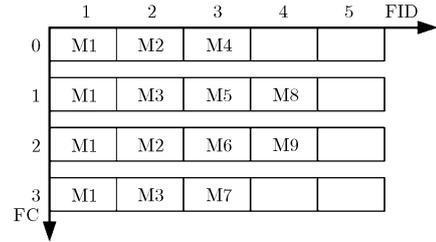


图1 静态段消息传输

所有消息都至少完成一次完整调度所需的FC数称为调度超级周期(Hyper Period, HP)。以图1为例，共有9个消息M1~M9, HP=4；其中，消息M1的周期为1，即每个FC都必须传输，所以单独占用了FID 1；M2和M3的周期为2，共用一个帧标识FID 2；消息M4~M7的周期都为4，所以共用FID 3。在FlexRay通信中，一个FC的长度即TC是有限的，考虑到每个消息只有一个FID，所以对于一个消息，在给定FID的情况下，确定其在一个HP中第1次出现在哪个FC，这个消息的调度也就确定了，该FC称为相位。故静态段最优调度的目标是为所有消息确定相位，且最后使用的或分配的FID数目最少。

为实现静态段最优调度目标，不妨假设FlexRay网络采用一个通道，需要传输的N个周期性实时消息集合 $S = \{s_1, s_2, \dots, s_N\}$ ，其中第i个消息 s_i 的属性用一个三元量表示，即 $s_i = (p_i, d_i, e_i)$ ，分别表示消息 s_i 的周期、截止期和传输时间。因本文主要研究调度建模，故假定已经数据打包，且静态帧长度已经确定，传输时间都为一个单位时间即 $e_i = 1$ ，同时假设 $d_i = p_i$ 。根据FlexRay的协议要求，通信周期长度 $TC = \gcd(p_1, p_2, \dots, p_N)$ ，调度超级周期 $HP = \text{lcm}(p_1, p_2, \dots, p_N)$ 。

2.1 消息分类模型

Klaus Schmidt等人将消息以周期进行分类后，在数论的基础上提出了一个最优调度模型如下^[15]：

$$\begin{aligned}
 \min &= n_{(1)} + \sum_{f_1 \in F^n(1)} \left\lfloor \frac{n_{(f_1)}}{f_1} + \frac{1}{f_1} \sum_{f_2 \in F^n(f_1)} \left\lfloor \frac{n_{(f_1, f_2)}}{f_2} + \frac{1}{f_2} \sum_{f_3 \in F^n(f_1, f_2)} \left\lfloor \frac{n_{(f_1, f_2, f_3)}}{f_3} + \dots \right\rfloor \right\rfloor \right\rfloor \\
 \text{s.t. } & \forall p_j \in P^n : \sum_{c_{p_j} \in C_{p_j}} n_{c_{p_j}} = N_{p_j}^n
 \end{aligned} \tag{1}$$

其中 f_1, f_2, \dots, f_K 为素数, n_{cpj} 表示在指定调度方式中所分配的消息数量, p_i 表示系统中的周期, $F^n(k)$ 表示各种周期对应的 (p_i/k) 的素数因子的集合, 即 $F^n(k) = \cup \text{factor}(p_i/k)$ 。这是一个非线性规划模型, 将优化目标中的向上取整操作适当变换后可得到一个整线性规划模型。消息分类模型的优点是输入规模只和周期分布有关, 与消息数量无关, 因此具有很大吸引力, 但在消息分类模型和周期调度表导出等方面仍存在较大问题。

2.2 问题描述

虽然文献[15]的消息分类模型输入规模不会随消息数量增大, 规划效率较高, 但应用中依然存在较大困难, 主要问题包括:

(1)消息分类模型的建立十分复杂, 模型系数矩阵获取困难, 在当消息周期类型较多情况下, 系数矩阵较为庞大, 每一个系数都需要额外的计算, 而且对于不同应用, 系数矩阵都不相同, 因此模型缺乏灵活性, 实用性较差, 没有充分利用 FlexRay 协议的特性。

(2)消息分类模型对静态段调度并不是一个完整方案, 只是估计出在最优调度情况下所需的最小 FID 数目以及消息的调度属性, 并未说明如何才能得到最优 FID 的结果, 即并未给出具体消息调度, 不能直接导出周期调度表。

所以, 消息分类模型可“预言”消息的最优值, 但其最优解只是消息的分类方法, 不能直接提供所需的具体分配方案, 且消息分类模型建立十分复杂。为此, 基于该模型与消息数量无关的优良特性, 本文将研究并提出一种模型自动生成方案, 然后根据所得的消息分类提出一种基于相位保留法的最优分配方案, 实现完整的消息调度。

3 自动模型系数矩阵生成算法

针对问题(1), 设调度超级周期为 HP, FS 为 HP 的所有素数因子集合, 即 $FS = \{f_1, f_2, \dots, f_M\}$, 其中 $f_i \leq f_{i+1}$; PR 表示调度属性集, 是 HP 的所有素数因子的不同组合排列构成的集合, 即 $PR = \{\{f_1\}, \{f_2\}, \dots, \{f_M\}, \{f_1, f_2\}, \dots, \{f_1, f_2, \dots, f_M\}\}$; $p(PR_i)$ 表示调度属性 PR_i 所对应的周期, 是 PR_i 中各元素的乘积; $n(PR_i)$ 表示采用调度属性 PR_i 的消息的数量, $\text{end}(PR_i)$ 表示 PR_i 中的最后一个元素; PS 表示所有周期的集合, 即 $PS = \{p_1, p_2, \dots, p_L\}$, 同时 $N(p_i)$ 表示周期为 p_i 的消息的数量。根据 FlexRay 协议, 调度超级周期为所有消息周期的整数倍, 也就是其余周期都是 HP 的因子。因此, HP 一旦确定, 则支持的消息周期分布就确定了。对消息分类模型经进一步整理后可得

$$\left. \begin{aligned} \min &= n(1) + \sum_{f_i \in FS, f_i \neq f_{i+1}} kp(f_i) \\ \text{s.t. } PR_i &\in PR : n(PR_i) + k(PR_i) + \sum_{f_j \notin PR_i} kp(PR_i \cup f_j) \\ &= kp(PR_i) \times \text{end}(PR_i), \\ p_j \in PS &: \sum_{p(PR_i) = p_j} n(PR_i) = N(p_j) \end{aligned} \right\} \quad (2)$$

式(2)中, 第 1 个约束条件为向上取整线性化规范操作, 第 2 个约束条件为各周期消息总数约束。其中, $kp(PR_i)$ 表示使用调度属性 PR_i 的消息所占 FID 数, 称为结果变量, $k(PR_i)$ 为松弛变量。假设整线性规的一般形式为

$$\left. \begin{aligned} \min y &= CX \\ \text{s.t. } AX &= B \\ O &\leq X \end{aligned} \right\} \quad (3)$$

其中, A 为约束方程系数矩阵, B 为约束边界(对于等式约束取上下界相等), C 为目标函数矩阵, X 为输入向量, O 为适维零向量。为生成整线性规划模型, 需求出对应的 A, B, C 3 个系数。自动模型系数矩阵生成算法的目标是根据系统周期特性, 自动产生上述系数矩阵。约束方程分为取整约束和消息总数约束两部分, 在求取系数矩阵式时需遍历所有质因子构成的组合。系数矩阵生成过程以超级周期 $HP=30$ 为例, 其周期分布为 $PS = \{1, 2, 3, 5, 6, 10, 15, 30\}$, 按式(2)可得取整约束方程生成树如图 2 所示。

每个节点在所有子节点的系数矩阵生成完毕后, 再加入本节点参数方程, 最后在根节点合成整个系数矩阵, 具体过程为:

(1)对于最底层, $A(n) = [1, 1, -\text{end}(PR_i)]$;

(2)对于其他层:

(a)深度方向合成如式(4), $A(n-1)$ 为底层一级的系数合成矩阵, O_1 为适维零矩阵, $[X_2 \ X_1]$ 是根

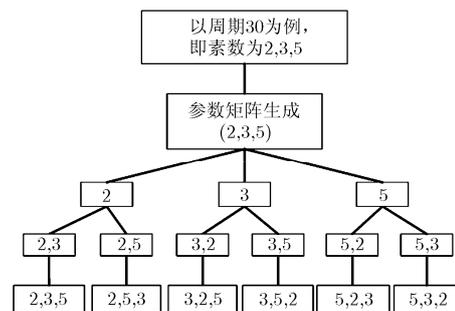


图 2 取整约束方程生成树

据式(2)为当前一层产生的系数矩阵, \mathbf{X}_1 为新增变量系数矩阵, \mathbf{X}_2 为所有子节点结果变量对应的系数矩阵。

$$\mathbf{A}(n) = \begin{bmatrix} \mathbf{A}(n-1) & \mathbf{O}_1 \\ \mathbf{X}_2 & \mathbf{X}_1 \end{bmatrix} \quad (4)$$

(b)广度方向矩阵合成如式(5), 其中 $\mathbf{A}(n-2)$ 和 $\mathbf{A}(n-1)$ 为需要合并的系数矩阵, \mathbf{O}_1 和 \mathbf{O}_2 为适维零矩阵, $\mathbf{A}(n)$ 为合并后的系数矩阵。

$$\mathbf{A}(n) = \begin{bmatrix} \mathbf{A}(n-2) & \mathbf{O}_1 \\ \mathbf{O}_2 & \mathbf{A}(n-1) \end{bmatrix} \quad (5)$$

(3)约束边界 \mathbf{B} 矩阵如式(6), 其中 \mathbf{O}_3 为适维零向量, \mathbf{B}_1 为各周期消息相应的数量构成的列向量。

$$\mathbf{B} = \begin{bmatrix} \mathbf{O}_3 \\ \mathbf{B}_1 \end{bmatrix} \quad (6)$$

(4) \mathbf{C} 为目标函数矩阵, 对应素数周期的元素置为 1, 其余置为零。

根据上述分析, 自动模型系数矩阵生成算法 AMCMG 可设计如表 1 所示。

表 1 AMCMG 算法

AMCMG (HP, \mathbf{B}_1):	
步骤 1	输入 HP, 各周期消息数量矩阵 \mathbf{B}_1 ;
步骤 2	计算 HP 素数因子集 FS;
步骤 3	获取取整矩阵 $\mathbf{AC}=\text{getcoeff}([\], \text{FS})$;
步骤 4	以 \mathbf{AC} 的行数构造列向量 \mathbf{O}_3 ;
步骤 5	将 \mathbf{O}_3 和 \mathbf{B}_1 合成参数 \mathbf{B} ;
步骤 6	生成消息数量约束参数 \mathbf{AD} ;
	以 \mathbf{B}_1 的行数及 \mathbf{AC} 的列数构造零矩阵 \mathbf{AD} ;
	for: 对所有的消息周期
	for: 对所有的分配变量
	分配变量周期=约束周期吗?
	是, 则 \mathbf{AD} 相应元素置 1;
	endfor 分配变量
	endfor 消息周期
步骤 7	用 \mathbf{AC} , \mathbf{AD} 构造 \mathbf{A} ;
步骤 8	生成目标系数矩阵 \mathbf{C} ;
步骤 9	返回参数 \mathbf{A} , \mathbf{B} , \mathbf{C} 。

表 1 中, 取整约束系数矩阵生成函数 $\text{getcoeff}()$ 可设计如表 2 所示。

3 基于相位保留的 FID 分配

针对 2.2 节中的问题(2), 即如何根据消息分类模型求解后得到的各种调度属性的消息数量, 确定每个消息的具体相位, 同时保证调度依然是最优的, 为此首先给出如下定义。

定义 1 同类消息: 周期相等且调度属性相同的所有消息称为同类消息。

表 2 取整约束系数矩阵生成函数 getcoeff

$\text{getcoeff}(\text{upfactorlist}, \text{factorlist})$:	
步骤 1	输入上级调度属性 upfactorlist , 下级调度属性 factorlist ;
步骤 2	若是最下层, 置 $\mathbf{A}=[1,1,-\text{end}(\text{upfactorlist})]$;
步骤 3	子节点处理
	for: 所有子节点
	相位扩展, 并以此为参数调用 getcoeff ;
	广度搜索返回, 按式(5)合成广度方向系数矩阵;
	endfor 所有子节点
步骤 4	当所有子节点完毕, 按式(4)合成本节点系数矩阵;
步骤 5	返回系数矩阵 \mathbf{A} 。

定义 2 兄弟类消息: 调度属性元素数相等且只有最后一个元素不相同的消息称为兄弟类消息。即对任意两个消息 s_i 和 s_j , 当最优规划后得到的调度属性分别为 $\text{PR}_i=(f_{i,1}, f_{i,2}, \dots, f_{i,m})$ 和 $\text{PR}_j=(f_{j,1}, f_{j,2}, \dots, f_{j,m})$, 如调度属性满足条件 $f_{i,l}=f_{j,l}, l=1, 2, \dots, m-1$; $f_{i,m} \neq f_{j,m}$, 则称 s_i 和 s_j 为兄弟类消息。

定义 3 父类消息: 如果消息 s_i 的调度属性为 $\text{PR}_i=(f_1, f_2, \dots, f_{m-1}, f_m)$, 称调度属性为 $\text{PR}_j=(f_1, f_2, \dots, f_{m-1})$ 的消息为 s_i 的父类消息。

定义 4 子类消息: 如果消息 s_i 的调度属性为 $\text{PR}_i=(f_1, f_2, \dots, f_m)$, 称调度属性为 $\text{PR}_j=(f_1, f_2, \dots, f_m, f_{m+1})$ 的消息为 s_i 的子类消息。根据该定义, 子类消息可以在父类消息的所有相位中调度。

对任意两个消息集 S_1 和 S_2 , 不妨设其周期分别为 p_1 和 p_2 , 通信周期长度 $\text{gcd}(p_1, p_2)=a$, 若 $a>1$, 则这两个消息集的消息可以共用 FID 进行调度, 但其前提是在调度前一个消息集的消息时, 要为后一个消息集预留相位, 这种方法叫基于相位保留 FID 分配算法(Phase Reserving based FID Assignment, PRFIDA)。显然, PRFIDA 算法的核心是确定下一个消息的相位, 当调度属性已知时, 根据上述定义, PRFIDA 算法确定下一个消息相位的规则可设计为:

(1)如果是同类消息, 尽可能保持父级调度相位不变, 即当父级相位没用完, 保持父级相位不变, 先依次用完当前相位; 否则如果当前父级相位用完, 则开启下一个新的父级相位。

(2)如果是兄弟类消息, 当下一位置应该为新相位, 则从新相位开始, 否则跳出当前父级相位, 从下一个父级相位开始。

(3)如果是子类消息, 则将本来的下一个同类消息相位扩展一位, 新扩展位为 0, 即当前相位加上新消息周期作为下一个消息相位。

(4)如果为父类消息, 当下一个位置为新相位, 则新相位为下一个消息相位, 否则如果不是新相位,

则在当前父级级相位增量作为下一个消息相位，同时收缩相位。

根据以上分析，基于相位保留的 FID 分配算法 PRFIDA 可设计如表 3 所示。

表 3 PRFIDA 算法

PRFIDA(upfactorlist, factorlist, initialposition, Xinfo)	
步骤 1	输入上下级调度属性 upfactorlist, factorlist, 初始相位 initialposition, 消息集 Xinfo;
步骤 2	先分配父节点 upfactorlist 消息; 获取父节点对应消息数量; 按顺序分配同类消息并标记; 确定下一个同类消息相位;
步骤 3	如果是最底层, 则返回;
步骤 4	为所有兄弟类消息分配相位; for: 所有子节点方向 相位标记扩展, 初始为 0; 依据当前参数调用 PRFIDA, 返回下一个相位; 相位收缩, 跳至父类消息的下一个相位; end for 所有子节点方向
步骤 5	输出消息分配相位。

定理 1 PRFIDA 算法能够实现确定调度属性下的最优 FID 分配。

证明 PRFIDA 算法是从上而下分配相位，即先分配上级的父类消息，再分配下级子类和兄弟类消息。对同类消息，相位保留法与顺序分配方法一样，能分配使用一个 FID 的所有相位而不会产生浪费；对于下级消息，PRFIDA 算法能够直接在当前相位中扩展，低一级相位不会产生相位浪费；对于兄弟类消息，PRFIDA 算法能最大限度地利用上级相位，产生最小的相位浪费，为并行消息尽可能多的保留上级相位。综上所述，PRFIDA 算法能产生最小的相位浪费，从而实现最优调度。证毕

例 假设超级周期为 $HP=30$ ，分配消息 $n(2,3,5)=30$ ，假设初始相位从 0 开始分配，则基于 PRFIDA 算法的各消息的相位依次为：0 6 12 18 24 2 8 14 20 26 4 10 16 22 28 1 7 13 19 25 3 9 15 21 27 5 11 17 23 29。

4 仿真实验及其分析

为实现静态段消息的完整调度，本文首先采用 AMCMG 进行静态段的自动建模，求解后对各类消息采用 PRFIDA 算法进行最优 FID 分配。为验证 AMCMG+PRFIDA 方案的有效性及其性能，仿真实验的硬件配置为 Intel I3 2.3G, RAM 2G，软件配置为 MATLAB2011a 环境下采用 TOMLAB/CPLEX 优化工具进行整线性规划。

首先，为验证本文方案在静态段调度建模时的

准确性和可用性，选用了一组典型的可预知结果的实例进行仿真实验。根据文献[18]，不妨采用一些重要标准规范常用的周期作为 HP，例如 2^n 以及支持多消息周期的 2310 和 30030 等，并令每种周期的消息数刚好等于消息周期，即每一种周期的消息刚好占用一个 FID。根据 FlexRay 规范，只要 HP 给定，则所支持的消息周期的集合 PS 也就确定了，所需的最少 FID 数即为消息周期数 $|PS|$ ，因此只需要检验 AMCMG 算法的优化结果与 $|PS|$ 是否相等，即可确认 AMCMG 建立的模型的正确性。随后，采用 PRFIDA 算法进行消息的 FID 分配，如果最终所需的 FID 数与 $|PS|$ 相比没有增加，即可确定实现了消息的最优分配。

表 4 给出了在上述条件下，9 种典型 HP 配置应用 AMCMG 算法生成模型并用 PRFIDA 算法进行消息分配的实验结果，其中 $|PS|$ 可作为事先确定的最优 FID 数， FID_{OTP} 为应用 AMCMG 算法生成模型并采用 CPLEX 优化工具所得的预期最优结果， FID_{PR} 为采用 PRFIDA 算法进行 FID 分配实际所需的 FID 数。以 $HP=64$ 为例，不考虑消息周期为 1 的情形，则系统所支持的消息周期集为 $PS=\{2,4,8,16,32,64\}$ ，故有消息周期数量 $|PS|=6$ ，各消息周期分别有 2, 4, 8, 16, 32 和 64 个消息需要调度，每一种消息周期的消息依次占用一个 FID，因此可事先确定的最优 FID 数 $|PS|=6$ 。结果显示，根据 AMCMG 算法建立的模型并求解后，最优结果 $FID_{OTP}=6$ ，与事先确定的最优 FID 数 $|PS|$ 保持一致，证实了 AMCMG 算法的正确性；在此基础上，进行 PRFIDA 的消息 FID 分配，可得实际所需的 FID 数 $FID_{PR}=6$ ，验证了 PRFIDA 算法能实现最优 FID 分配。

建立调度周期表是实现最优 FID 分配至关重要的一个环节，为可靠验证 PRFIDA 算法的准确性，本文模拟了一种更为复杂的调度情景。不妨令消息周期为 p_i 的消息数量为周期的 rate 倍，并向上取整，

表 4 模型生成结果与最优规划结果

HP	$ PS $	FID_{OTP}	FID_{PR}
16	4	4	4
32	5	5	5
64	6	6	6
128	7	7	7
30	7	7	7
60	11	11	11
210	15	15	15
2310	31	31	31
30030	63	63	63

即 $N(p_i)=\text{ceil}(\text{rate}\times p_i)$ ，表 5 给出了 7 种典型 HP 配置下 $\text{rate}=10.0\sim 10.4$ 时的实验结果，以 $\text{HP}=64 \wedge \text{rate}=10.4$ 为例，此时系统所支持的消息周期集为 $\text{PS}=\{2,4,8,16, 32, 64\}$ ，各周期的消息数分别为 21, 42, 84, 165, 333 和 666，采用 AMCMG 算法建立模型并得到最优规划结果 $\text{FID}_{\text{OTP}}=63$ ，即预期所需的最少 FID 数为 63，当采用 PRFIDA 算法进行具体 FID 分配时，其实际分配结果 FID_{PR} 也正好为 63，即所需的 FID 数没有额外增加。表 5 中实验结果显示，对所有情况均有 $\text{FID}_{\text{PR}}=\text{FID}_{\text{OTP}}$ ，即 PRFIDA 算法能实现确定调度属性下的最优 FID 分配。

为进一步验证本文所提方案的效率，拟对多种大规模调度的应用场景进行模拟实验。不妨设周期为 p_i 的消息数量为 $N(p_i)=\text{np}\times p_i$ ，取 $\text{HP}=8, 64, 60, 2310$ 共 4 种典型配置种情况下，使 np 从 1 到 800 依次增大，测试算法所需时间。为比较算法性能，本文同时采用了 Martin L 提出的 2 维装箱算法在同样的条件下进行消息调度，图 3 显示了各算法所需的时间。图 3(a)显示，Martin L 提出的 2 维装箱算法只能在消息数量少、周期分布简单的情况下能够获得最优解，如 $\text{HP}=8 \wedge \text{np}\leq 15$ 和 $\text{HP}=64 \wedge \text{np}\leq 2$ 时能够求解，在消息数量增加，周期关系稍微复杂时，2 维装箱算法很快就无法求解。图 3(b)显示了本文 AMCMG+PRFIDA 方案所需时间，在所有假设的实验条件中，该方案不仅皆能求得最优解，实

现消息最优 FID 分配，而且所需时间明显更短。同时，对于给定的 HP，当 np 增大时，即消息数量增大时，算法时间基本保持不变，显示了模型输入规模与消息数量无关的特性。在实验中， $\text{HP}=2310 \wedge \text{np}=800$ 、消息总数量达 24800 时，实现最优调度所需时间也只有 5.1 s。

负载均衡是 FID 分配算法的另外一个重要的性能指标，因为如果负载分布均衡，则剩余 SLOT 可以更方便地为未来系统扩展所利用。根据 FlexRay 协议，一个 FC 实际需要传输的消息数称为这个 FC 的负载。消息数量依旧设 $N(p_i)=\text{ceil}(\text{rate}\times p_i)$ ，图 4(a)~4(e)展示了在 $\text{HP}=60$ 且 $\text{rate}=10.0\sim 10.4$ 时各 FC 实际分配的消息数均衡情况，此时系统支持的消息周期集为 $\text{PS}=\{2,3,4,5,6,10,12,15, 20,30, 60\}$ ，支持的消息周期密度较大，具有较好的研究价值。当 $\text{rate}=10.0$ 时，每个周期的消息刚好能够填满整 10 个 FID，这时系统所需的 FID 数为 $10\times 11=110$ 个，每个 FC 的传输的消息数量完全相等，这时负载分配是最均衡的；当 rate 不为整数时，即使在最优规划和 PRFIDA 算法的 FID 分配时，有的 FID 不再能被完全利用，这时 FC 的负载稍有差异，如图 4(b)所示的 $\text{rate}=10.1$ 时的负载均衡情况，各 FC 负载差别最大，各负载分布统计为：1 个 FC 负载为 110, 10 个 FC 负载为 111, 28 个 FC 负载为 112, 21 个 FC 负载为 113，负载分布也基本处于平衡状态。

表 5 最优规划预期 FID 与采用 PRFIDA 算法实际分配的 FID 结果

HP	rate=10.0		rate=10.1		rate=10.2		rate=10.3		rate=10.4		rate=10.5	
	FID _{OTP}	FID _{PR}										
32	50	50	52	52	52	52	52	52	53	53	53	53
64	60	60	62	62	62	62	63	63	63	63	63	63
128	70	70	72	72	72	72	73	73	74	74	74	74
30	70	70	72	72	72	72	73	73	74	74	74	74
60	110	110	113	113	113	113	114	114	115	115	116	116
210	150	150	153	153	154	154	156	156	157	157	158	158
2310	310	310	315	315	318	318	321	321	324	324	326	326

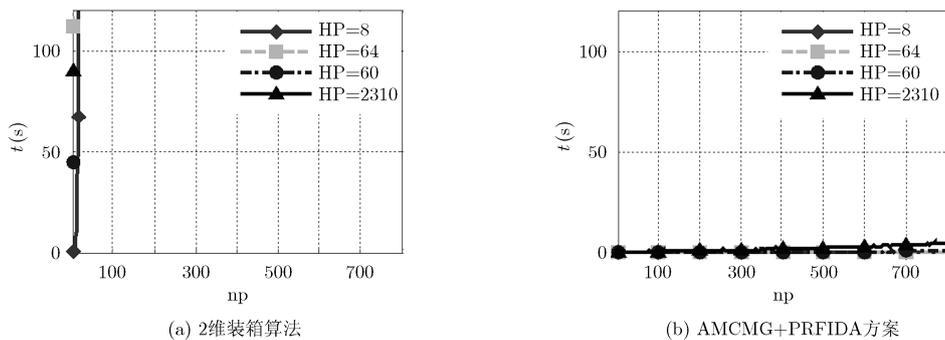


图 3 AMCMG+PRFIDA 方案和 2 维装箱算法所需时间比较

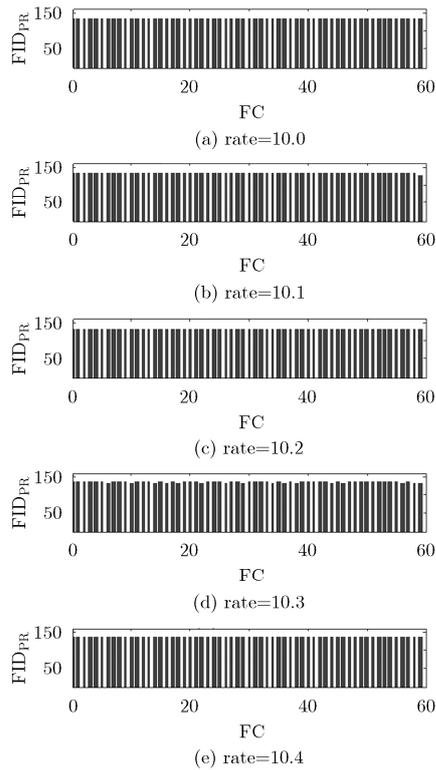


图 4 HP=60 时各通信周期的负载分布

5 结束语

本文分析了 FlexRay 静态段在整车 ECU 互联应用中所面临的调度难题, 研究了基于 FlexRay 传输的消息特性, 讨论了文献[15]中基于消息分类最优调度模型在实际应用中的不足, 并在此基础上提出了 AMCMG+PRFIDA 方案。其中, AMCMG 算法利用了 FlexRay 静态段消息周期分布特性, 使得复杂的消息分类最优调度模型能够自动生成; 为进一步确定每一个消息的具体 FID 和相位分配, 并保持先前的原有模型求解时预期的最优 FID 分配数不变, 在给定调度属性下的, 提出了 PRFIDA 算法进行 FlexRay 静态段消息 FID 分配。仿真实验结果表明, AMCMG 算法能快速正确地建立调度模型, PRFIDA 算法能使各消息在调度时尽可能共用 FID, 最大限度地减小 FID 的浪费, 最终实现静态段消息的整体最优调度。在实践中, 根据本文提出的方案, 只需确定调度超级周期和各类消息的数量即可得到最优调度结果, 并将每一个消息的调度确定到具体位置, 从而能够得到一个完整的静态段调度表。

参考文献

- [1] Fleming B. New automotive electronics technologies[J]. *IEEE Transactions on Vehicular Technology*, 2012, 7(4): 4-12.
- [2] Tuohy S and Glavin M. Next generation wired intra-vehicle networks, a review[C]. Proceedings of IEEE on Intelligent Vehicles Symposium, Gold Coast, Australia, 2013: 777-782.
- [3] TTTech. Time-triggered protocol ttp/c, high-level specification document, protocol version 1.1[OL]. <http://www.ttttech.com>, 2014.
- [4] Berwanger J and Peller M. Byteflight -a new protocol for safety critical applications [C]. Proceedings of the 2000 FISITA World Automotive Congress, Seoul, Korea, 2000: 1-7.
- [5] 王聪, 张凤荔, 王瑞锦, 等. 一种网络时延矩阵分布式自适应重建算法[J]. 电子与信息学报, 2014, 36(4): 840-846.
- [6] 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.
- [7] Longo S and Herrmann G. Robust scheduling of sampled-data networked control systems[J]. *IEEE Transactions on Control Systems Technology*, 2012, 20(6): 1613-1621.
- [8] 徐超, 何炎祥, 陈勇, 等. 一种多核系统可靠性加强的任务调度方法[J]. 电子学报, 2013, 41(5): 1019-1024.
- [9] Ouedraogo L and Kumar R. Exact response time of flexray communication protocol[C]. Proceedings of Wireless Communications and Mobile Computing Conference, Istanbul, Turkey, 2011: 789-794.
- [10] Lange R and Vasques F. Guaranteeing real-time message deadlines in the flexray static segment using a on-line scheduling approach[C]. Proceedings of the 9th IEEE International Workshop on Factory Communication Systems, Lemgo, Germany, 2012: 301-310.
- [11] Tanasa B and Bordoloi U D. Reliability-aware frame packing for the static segment of flexray[C]. Proceedings of Emsoft, Taipei, China, 2011: 175-184.
- [12] Minkoo K and Kiejn P. Frame packing for minimizing the bandwidth consumption of the flexray static segment[J]. *IEEE Transactions on Industrial Electronics*, 2013, 60(9): 4001-4008.
- [13] Shan D. A hybrid GA-based scheduling method for static segment in flexray systems[C]. Proceedings of on Control and Decision, Xuzhou, China, 2010: 1548-1552.
- [14] Martin L and Michael G. *Advances in Real-time Systems*[M]. Berlin: Springer, 2012: 323-339.
- [15] Schmidt K and Schmidt E G. Message scheduling for the flexray protocol: the static segment[J]. *IEEE Transactions on Vehicular Technology*, 2009, 58(5): 2170-2179.
- [16] FlexRay Consortium. FlexRay communication systems protocol specification, version 3.0.1[OL]. <http://www.FlexRay.com>, 2010.
- [17] Grenier M, Havet L, and Navet N. Configuring the communication on flexray: the case of the static segment[C]. Proceedings of the 4th European Congress on Embedded Real Time Software, Toulouse, France, 2008: 1-18.
- [18] AUTOSAR. Specification of flexray network management. [OL]. <http://www.autosar.org>, 2013.

- 龚志鹏: 男, 1974 年生, 博士生, 讲师, 研究方向为车载通信网络。
 陈特放: 男, 1957 年生, 博士, 教授, 博士生导师, 研究方向为交通信息工程及控制。
 邹复民: 男, 1976 年生, 博士, 教授, 研究方向为交通信息工程及控制。