此,对基于 CAN 总线的机器人通信系统进行实时性分析十分必要.

**Table 2** Verification results

**表 2** 验证结果

| 名称 | 结果 |
|---|---|
| 安全性 | 满足 |
| 活性 | 满足 |
| 无饥饿性 | 满足 |
| 下位关节节点之间不可通信 | 满足 |
| 同一时刻总线上只有一个节点传输数据 | 满足 |
| 同一时刻有多个节点请求发送数据 | 满足 |
| 高优先级节点总能获得总线权 | 满足 |

### 4.2 实时性分析与验证

CAN 总线的通信延迟时间可以分为 4 个部分,包括数据生成延迟、仲裁延迟、传输延迟和接收数据延迟.这里,CAN 总线的总延迟时间用 *Rm* 表示,仲裁延迟用 *Tm* 表示,传输延迟用 *Cm* 表示,在这 4 部分的延迟时间中,数据生成延迟和接收数据延迟主要由节点的主控制器产生,很大程度上取决于数据采集系统的设计方案和主控制器的执行速度等,按照现在的处理器执行速度而言,该部分时间可以忽略不计.因此,报文总延迟时间可以表示为

$$Rm=Tm+Cm.$$

其中,数据传输延迟受报文长度和总线上报文传输速度的影响,根据 CAN 数据帧的标准帧和扩展帧格式可以知道,数据帧包括 44 位(标准帧)或者 64 位(扩展帧)比特位.根据位填充的规定,每出现 5 个连续的相同的位,就要在其后面加一个填充位,但是界定符、应答段和结束段不需要加填充位,所以在这两种不同的帧格式中,会有 34 位或者 54 比特位是参与位填充的.在帧的数据段可以封装 1~8 个字节的信息,所以在封装有 *N* 个字节的数据帧中,最多可以填充的位数是$(N\times8+34(54)-1)/5$.设定发送一个比特位的时间是 *Tbit*,那当分别发送标准帧和扩展帧的情况下,它们的传输时间公式可以表示为[18]

$$C1=\{N\times8+44+(N\times8+34-1)/5\}\times T,$$

$$C2=\{N\times8+64+(N\times8+54-1)/5\}\times T.$$

目前,CAN 总线上报文的最快传输速度可以达到 1Mbps,这种情况下,发送一位的时间是 1μs.综上所述,CAN 总线的通信时延主要取决于仲裁延迟,因此,对 CAN 通信实时性的影响也主要在于仲裁延迟.这里的仲裁延迟是指低优先级节点在发送报文时,因为仲裁失败并等待总线空闲而产生的延迟时间.本文中,重点考虑了仲裁延迟对 CAN 实时性的影响.

为记录节点的仲裁延迟,在关节控制器时间自动机中引入了局部 *clock* 变量 *x*,用来记录节点参与仲裁的时间,包括仲裁失败后等待的时间和重新仲裁的时间.我们首先针对采用静态优先级策略的 CAN 总线进行了实验,发现:当 CAN 总线上挂接 7 个节点时,能在允许时间范围内完成总线仲裁;但是随着挂接的节点数逐渐增加,优先级较低的节点最坏仲裁时延会不断增大,甚至会退出总线仲裁,进而无法获得总线权.这就严重影响了通信的实时性.

为了解决通信的实时性问题,对 CAN 进行了扩展,产生了利用时间触发的通信协议 TTCAN[19].TTCAN 是基于表的静态调度算法,其总线上的通信是按照矩阵周期的调度安排周期性循环进行的.在网络通信开始之前,需要针对整个网络拓扑结构和应用层协议综合考虑,制定出矩阵周期.TTCAN 协议虽然较 CAN 而言提高了实时性,但是它不具备灵活性,一旦网络完成搭建,便不允许增加或减少通信节点.基于这个原因,Zuberi 最早将 EDF 调度算法引入到 CAN 总线中[20],得出动态调度算法在软实时消息的调度上要优于静态调度算法,但是采用 EDF 算法非常容易引起优先级倒置的现象发生.后续又有很多文献针对 EDF 算法进行了改进,在此基础上得到很多优化的动态调度算法[21].

在这些研究的基础上,本文引入了易实现且不用频繁构建调度表的改进的动态优先级方法,该方法的主要

思想是:从 CAN 报文仲裁场的 11 位标识符中预留一个最高优先级系列,在仲裁过程中,当节点仲裁失败的次数达到一定值后,将该节点的优先级进行升级,提升到最高优先级,即,将预留的序列赋给该节点.为保证正常通信,当节点仲裁成功完成传输后,要将该节点的优先级恢复到原来的优先级.这种方法可以避免为提升节点优先级而导致出现多个节点拥有相同优先级的问题发生.

基于这个方法,在 UPPAAL 中重新对关节控制器进行了建模,如图 11 所示.在该时间自动机中引入了动态优先级策略,为记录节点仲裁失败的次数,引入了整型变量 *failNum*,本文中设定:当仲裁失败的次数为 3 次时,对该节点的优先级进行提升.在自动机中,该方法通过函数 *Raise_pri*()实现,当节点通过提升优先级而获得总线完成传输后,要恢复其优先级,保证通信正常运行.其优先级的恢复由函数 *reset_pri*()实现.

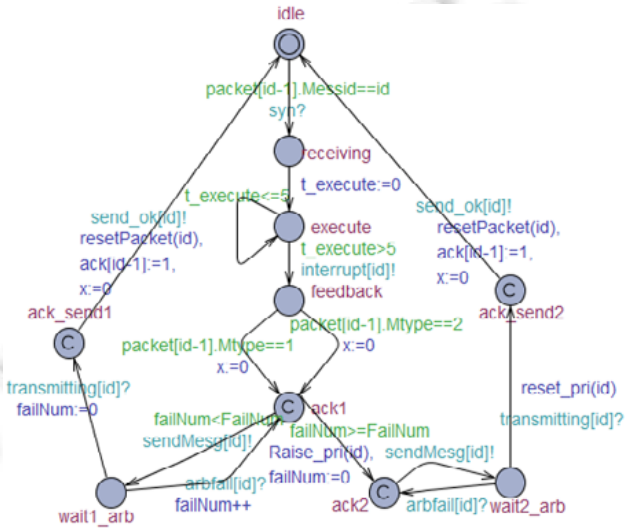


Fig.11 Module of joint_controller with dynamic policy

图 11 加入动态优先级后的关节控制器模型

在 UPPAAL 中,分别针对采用静态优先级策略和采用动态优先级策略的系统模型进行了实验,在实验中,不断增加总线节点的数量并记录节点完成仲裁的最坏仲裁延迟,实验结果如图 12 所示,图中横坐标表示总线上挂接的节点数,纵坐标表示节点完成仲裁的最坏仲裁延迟.其中,曲线 *s*[*n*]表示采用静态优先级策略的实验结果,曲线 *d*[*n*]代表采用动态优先级策略后的实验结果.

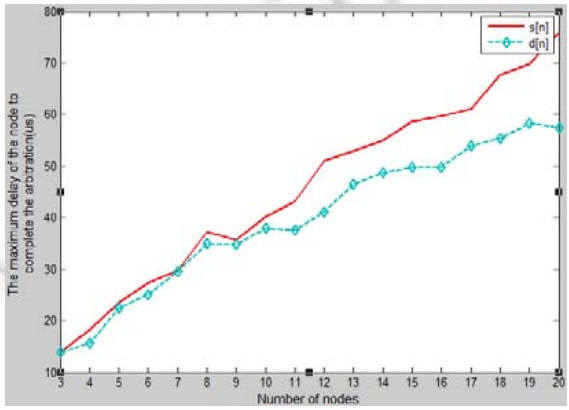


Fig.12 Results of the two methods

图 12 两种方法的实验对比结果

从实验结果可以看出:采用静态优先级策略时,当总线上的节点数达到 11 个后,最坏仲裁时延增长速率变大,当达到 17 个以后速率增长迅速变大,这表明当总线上的负载量越来越大时仲裁延迟会越来越大,这就在一定程度上影响了机器人通信的实时性.部署动态优先级后,这种情况明显改善,即使节点数达到 20 个,节点最坏仲裁延迟增长速率也不是特别迅速,并且相对比较缓和,这就说明采用动态优先级策略后能有效地减小节点仲裁产生的延迟,进而能够提升基于 CAN 的机器人通信的实时性.能够满足机器人的通信需求,并且还可以在一定程度上提升总线的负载,对服务型机器人的发展提供一定的保障.

## 5 相关工作

在参考了大量关于 CAN 协议调度方面的参考文献后,受这些研究的启发,想到在基于 CAN 的机器人通信系统中加入动态优先级策略,进而分析加入动态优先级策略后对该系统实时性的影响.其中,一些重要参考文献如下.

文献[22]针对工业实时通信中标准 CAN 总线协议在网络拥塞情况下出现某些帧无法发送和丢帧等问题,提出了一种静态和动态相结合的调度算法.该算法首先建立一张帧信息表;接着,利用该表来确定当前帧的类型;然后,通过表中的优先级进行实时传输.如果当前信道上出现相同优先级的帧,则使用动态调度算法进行调度传输.

文献[21]中,Burns 等人对 CAN 的原始调度性进行了分析,并指出了不足之处,即:在实际应用中,消息最终可能会错过最后截止时间而无法实现调度.作者针对这个问题,在文中提供了对原有方法的修改分析,并且引入了一种适用于 CAN 的最优优先级排序的方法.

文献[23]中,Davis 等人提出 CAN 的现有可调度性分析基于最高优先级,而在实际应用中,一些 CAN 设备驱动程序实现了 FIFO 而不是基于优先级的队列.在文中,作者对响应时间和 CAN 消息的最优优先级分配策略进行了分析,提出:其中一些节点使用 FIFO 队列,而其他节点使用优先队列.但是引入 FIFO 调度后,会对 CAN 的实时性造成影响.

由于 CAN 采用的是固定优先级策略和非破坏性仲裁机制,所以当总线上的负载量很大时,总是高优先级的节点获得总线而低优先级节点不能仲裁成功,进而会导致这些低优先级节点的饥饿现象.文献[24]中,Anwar 等人针对低优先级消息饥饿的问题提出一个方案,为系统分配一个可调节的消息 ID 窗口来识别 CAN 总线上的消息,代替了原来 CAN 协议中用单个标识符标识消息 ID 的方法.在该方案中,消息的优先级将在运行时进行调整,而不会对消息的内容或含义产生任何影响.该方法有助于在控制器局域网上实现动态消息调度.通过实验证明,该方案能有效地降低 CAN 网络上的饥饿概率.

文献[25]中,Murtaza 等人同样针对低优先级节点可能面临的饥饿现象提出了一种方法,就是在总线上引入一个 *master* 节点,由这个节点来解决饥饿现象.在文中,作者也引入了动态优先级的思想,他的主要做法是为每个节点分配两个不同的优先级,并将所有的这些节点信息都存入 *master* 节点中,当总线上出现饥饿现象时,该 *master* 节点根据记录的信息动态地改变饥饿节点的优先级.当总线正常传输时,*master* 节点只是不断监测总线,只有监测到饥饿节点时,该 *master* 节点才会处于活动状态.在整个通信过程中,*master* 节点保证每个节点都能够公平的占用总线.

## 6 总 结

本文提出了基于CAN的现场总线型控制系统的形式化验证方法,将基于CAN的机器人系统与模型检测方法连接起来,给出系统形式化架构和构建时间自动机模型的方法步骤,并提供了各个模块在 UPPAAL 中的具体实现;应用该形式化方法对基于 CAN 的机器人通信系统进行了功能正确性验证,在验证的基础上,对 CAN 的实时性进行了分析,发现采用常规的 CAN 总线不能保证日益发展的机器人通信的实时性,因此,在模型中部署了改进的动态优先级策略,并对模型的实时性进行了分析和验证.实验表明:采用动态优先级不仅能改善常规CAN总线的实时性,还能够在一定程度上改善CAN总线的负载.该研究表明:设计人员可以根据应用在CAN通

信的设计中通过改进仲裁策略,动态调整优先级来满足关键应用的实时性要求.此外,本文还分享了一些建模方面的经验.本文的不足之处是没有考虑传输失败后重传的情况,因为重传机制也会在一定程度上增加 CAN 总线的负载量,对低优先级节点的仲裁产生影响.

未来我们会将传输失败因素考虑进去,并借助概率模型检测技术分析在不同传输失败概率情况下对 CAN 总线负载的影响.进一步完善动态优先级策略,使得该方法应用范围更广泛,算法更健壮,并分析和验证引入传输失败概率后,该方法的健壮性和效果.

**References**:


[1] Nilsson DK, Larson UE, Picasso F, Jonsson E. A first simulation of attacks in the automotive network communications protocol flexray. In: Proc. of the Int'l Workshop on Computational Intelligence in Security for Information Systems (Cisis 2008). Genova: DBLP, 2009. 84–91. [doi: 10.1007/978-3-540-88181-0_11]

[2] Wang DQ, Gao SY, Chen YQ, Wang Y, Liu Q. Intelligent control system based on CAN-bus for car doors and windows. In: Proc. of the Int'l Conf. on Anti-Counterfeiting, Security, and Identification in Communication. IEEE, 2009. 242–245. [doi: 10.1109/ICASID.2009.5276906]

[3] Sabelhaus AP, Bruce J, Caluwaerts K, Manovi P. System design and locomotion of SUPERball, an untethered tensegrity robot. In: Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA). IEEE, 2015. 2867−2873. [doi: 10.1109/ICRA.2015.7139590]

[4] Kaneko K, Harada K, Kanehiro F, Miyamori G. Humanoid robot HRP-3. In: Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems. IEEE, 2008. 2471−2478. [doi: 10.1109/IROS.2008.4650604]

[5] Kim JY, Park IW, Lee J, Kim MS, Cho BK, Oh JH. System design and dynamic walking of humanoid robot KHR-2. In: Proc. of the IEEE Int'l Conf. on Robotics and Automation. IEEE, 2005. 1431−1436. [doi: 10.1109/ROBOT.2005.1570316]

[6] Li XY, Huang M, Zhan J, Ni YL, Pang FY. CANoe-Based modeling and simulation for heavy lorry CAN bus network. In: Proc. of the AsiaSim 2012. Berlin, Heidelberg: Springer-Verlag, 2012. 107−114. [doi: https://doi.org/10.1007/978-3-642-34387-2_13]

[7] Liu T, Wang SL, Zhan NJ. Safety verification of trajectory planning for multiple robots. Ruan Jian Xue Bao/Journal of Software, 2017,28(5):1118−1127 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5207.htm [doi: 10.13328/j.cnki.jos.005207]

[8] Resten Y, Maler O, Marcus M, Pnueli A, Shahar E. Symbolic model checking with rich assertional languages. Theoretical Computer Science, 2001,256(1-2):93−112. [doi: https://doi.org/10.1016/S0304-3975(00)00103-1]

[9] Xiao D, Zhu YF, Liu SL, Wang DX, Luo YQ. Applied Mechanics & Materials. 2015,716−717:1382−1386. [doi: 10.4028/www.scientific.net/AMM.716-717.1382]

[10] Alur R, Dill DL. A theory of timed automata. Theoretical Computer Science, 1994,126(2):183−235. [doi: 10.1016/0304-3975(94)90010-8]

[11] Berendsen J, Gebremichael B, Vaandrager FW, Zhang MM. Formal specification and analysis of zeroconf using uppaalS. ACM Trans. on Embedded Computing Systems, 2011,10(3):1−32. [doi: 10.1145/1952522.1952527]

[12] Pan C, Guo J, Zhu LF. Modeling and verification of CAN bus with application layer using UPPAAL. Electronic Notes in Theoretical Computer Science, 2014,309:31−49. [doi: https://doi.org/10.1016/j.entcs.2014.12.004]

[13] Gu JS, Silva CWD. Development and implementation of a real-time open-architecture control system for industrial robot systems. Engineering Applications of Artificial Intelligence, 2004,17(5):469−483. [doi: 10.1016/j.engappai.2004.03.010]

[14] Behrmann G, David A, Larsen KG. A tutorial on UPPAAL. In: Bernardo M, ed. Proc. of the Formal Methods for the Design of Real-Time Systems. Springer-Verlag, 2004. 200−236. [doi: 10.1007/978-3-540-30080-9_7]

[15] Dai SX, Hong M, Guo B, Yang QH, Huang W, Xu BP. Schedulability analysis model for multiprocessor real-time systems using UPPAAL. Ruan Jian Xue Bao/Journal of Software, 2015,26(2):279−296 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4781.htm [doi: 10.13328/j.cnki.jos.004781]

[16] Battiston A. Software in C++ for communication between CAN bus and ROS in a robot vehicle [Ph.D. Thesis]. Padua University, 2014.

[17] Behrmann G, David A, Larsen KG. A tutorial on uppaal. In: Proc. of the Formal Methods for the Design of Real-Time Systems. Berlin, Heidelberg: Springer-Verlag, 2004. 200–236.

[18] Dong YK. Research on scheduling algorithm based on CAN bus [Ph.D. Thesis]. Beijing: Tsinghua University, 2011 (in Chinese).

[19] Führer T, Müller B, Dieterle W, Hartwich F, Hugel R, Walther M, Gmbh RB. Time triggered communication on CAN. In: Proc. of the Time Triggered CAN (TTCAN). Bibliogr, 2001.

[20] Zuberi KM, Shin KG. Non-Preemptive scheduling of messages on controller area network for real-time control applications. In: Proc. of the 1st IEEE Real-Time Technology and Applications Symp. (RTAS'95). Chicago: IEEE Computer Society, 1995. 240−249.

[21] Davis RI, Burns A, Bril RJ, Lukkien JJ. Controller area network (CAN) schedulability analysis: Refuted, revisited and revised. Real-Time Systems, 2007,35(3):239−272. [doi: 10.1007/s11241-007-9012-7]

[22] Liu JF, Gui WH, Huang ZW, *et al*. Study of the application on a scheduling algorithm of CAN bus in locomotive brake. Journal of Chinese Computer Systems, 2009,30(1):183−187 (in Chinese with English abstract).

[23] Davis RI, Kollmann S, Pollex V, *et al*. Controller area network (CAN) schedulability analysis with FIFO queues. In: Proc. of the Euromicro Conf. on Real-Time Systems. IEEE Computer Society, 2011. 45−56. [doi: 10.1109/ECRTS.2011.13]

[24] Anwar K, Khan ZA. Dynamic priority based message scheduling on controller area network. In: Proc. of the Int'l Conf. on Electrical Engineering. IEEE, 2007. 1−6. [doi: 10.1109/ICEE.2007.4287302]

[25] Murtaza AF, Khan ZA. Starvation free controller area network using master node. In: Proc. of the Int'l Conf. on Electrical Engineering. IEEE, 2008. 1−6. [doi: 10.1109/ICEE.2008.4553945]

**附中文参考文献:**

[7] 刘涛,王淑灵,詹乃军.多机器人路径规划的安全性验证.软件学报,2017,28(5):1118−1127. http://www.jos.org.cn/1000-9825/5207.htm [doi: 10.13328/j.cnki.jos.005207]

[15] 代声馨,洪玫,郭兵,杨秋辉,黄蔚,徐保平.多处理器实时系统可调度性分析的 UPPAAL 模型.软件学报,2015,26(2):279−296. http://www.jos.org.cn/1000-9825/4781.htm [doi: 10.13328/j.cnki.jos.004781]

[18] 董寅康.基于 CAN 总线的调度算法的研究[博士学位论文].北京:清华大学,2011.

[22] 刘剑锋,桂卫华,黄志武,等.一种 CAN 总线调度算法在机车制动机上的应用研究.小型微型计算机系统,2009,30(1):183−187.

**孟瑶**(1987－),女,河北保定人,硕士,主要研究领域为机器人系统软件安全,计算机网络协议分析.

**王瑞**(1981－),女,博士,教授,CCF 专业会员,主要研究领域为机器人安全验证.

**李晓娟**(1968－),女,博士,教授,CCF 专业会员,主要研究领域为系统形式建模与分析,机器人系统软件安全,计算机网络协议分析.

**张杰**(1967－),女,副教授,主要研究领域为嵌入式系统,形式化验证.

**关永**(1966－),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为高可靠嵌入式系统,形式化验证.