A comparison of the key features of our architecture with other works reported is shown in Tab.1. The proposed SA architecture exhibits 80% improvement in the sensing delay when compared to the conventional SA and is superior to Refs.[9], [10] and [11]. However, the number of transistors for the proposed SA is 20, and the number of transistors for the conventional SA is only 8. The number of the transistors for the proposed SA is 2.5 times greater than that of the conventional SA, which leads to the increase in power consumption. The propose SA occupies an active area of 1400 μm^2. But the area of the proposed SA accounts for only a very small part of the area of the whole 16 kB SRAM. For example, the area of the proposed SA accounts for only 0.1% of the whole SRAM, and the area overhead does not have a significant impact on the whole area of SRAM. The power consumption of Ref.[10] decreases by 75%, but the sensing delay of Ref.[10] increases by 194% compared to the proposed SA. Ref.[8] sensing delay is decreased by 44%, but its power consumption is increased by 218% compared to the proposed SA.

Tab.1　Simulation results and comparison

method	Technology	Sensing delay	Power consumption	Area
Conventional SA	0.13 μm	1.5 ns	0.024 mW	\\
Ref.[8]	0.18μm	0.38 ns	0.29 mW	38.6 μm^2
Ref.[9]	0.13μm	1.9 ns	\\	300 μm^2
Ref.[10]	65 nm	2 ns	0.018 mW	\\
Ref.[11]	28 nm	0.71 ns	\\	\\
Proposed SA	0.13 μm	0.68 ns	0.072 mW	1400 μm^2

Fig. 7 shows the timing scheme of the proposed SA in a 0.25 kB SRAM. In this SRAM, the sensing signal SAEA is set at "1" when ΔV_{BL} =260 mV. The ΔV_{BL} is greater than ΔV_{BLmin} of the proposed SA due to noise, offset and some other factors that can affect the performance of the SA[12-15]. Therefore, enough redundant ΔV_{BL} is required to make sure that the SA works at the right state. The layout of the SRAM and the sense amplifier is illustrated in Fig. 8.

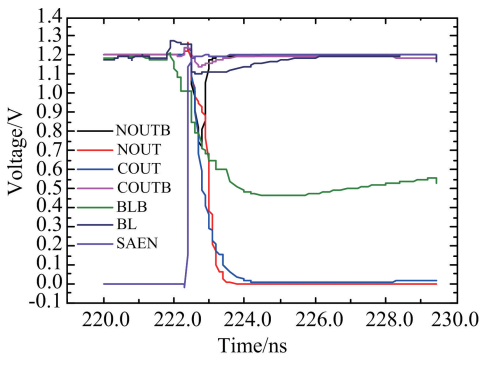

Fig.7　Timing scheme of the proposed SA in a 0.25 KB SRAM

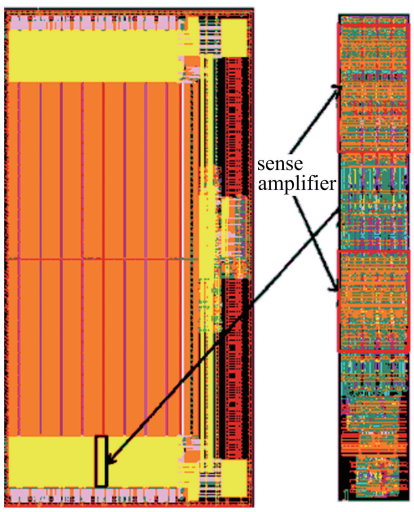

Fig.8　SRAM and sense amplifier layout

3　Conclusion

Compared with the conventional SA, the ΔV_{BLmin} and the sensing delay of the proposed SA are reduced by 95% and 80%, respectively. Although the power consumption and the area of the proposed SA are a little larger than that of the conventional SA at the same ΔV_{BL}, the proposed

SA is capable of reducing bit-line swing and shortening sensing delay.

References

[1] CHRISANTHOPOULOS A, TSIATOUHAS Y, ARAPOYANNI A, et al. SRAM oriented memory sense amplifier design in 0.18 μm CMOS technology [C]// IEEE International Symposium on Circuits and Systems. Phoenix-Scottsdale, USA: IEEE Press, 2002: 145-148.

[2] WEN L, CHENG X, ZHOU K J, et al. Bit-Interleaving-enabled 8T SRAM with shared data-aware write and reference-based sense amplifier [J]. IEEE Transactions on Circuits & Systems II Express Briefs, 2016, 63(7): 643-647.

[3] PENG C Y, TAO Y, LU W, et al. A novel cascade control replica-bitline delay technique for reducing timing process-variation of SRAM sense amplifier [J]. IEICE Electronics Express, 2015, 12(5): 20150102.

[4] MUKHOPADHYAY S, MAHMOODI H, Roy K. Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2006, 24(12): 1859-1880.

[5] DO A T, KONG Z H, YEO K S, et al. Design and sensitivity analysis of a new current-mode sense amplifier for low-power SRAM [J]. IEEE Transactions on Very Large Scale Integration Systems, 2011, 19(2): 196-204.

[6] NOH K J, KIM J H, LEE C H, et al. A new dual asymmetric bit-line sense amplifier for low-voltage dynamic random access memory [J]. IEICE Electronics Express, 2013, 10(18): 20130647.

[7] UDDIN N, THIEDE A. Common gate cross-coupled differential amplifier for near-field sensors [J]. Electronics Letters, 2009, 45(18): 918-920.

[8] DO A T, LOW Y S J, KONG Z H, et al. A full current-mode sense amplifier for low-power SRAM applications [C]// IEEE Asia Pacific Conference on Circuits and Systems. Macao, China: IEEE Press, 2008: 1402-1405.

[9] ZHANG H. Design of a high speed sense amplifier circuit [J]. Microelectronics, 2015: 45(3): 294-297.

[10] LEE H, ALZATE J G, DORRANCE R, et al. Design of a fast and low-power sense amplifier and writing circuit for high-speed MRAM [J]. IEEE Transactions on Magnetics, 2015, 51(5): 1-7.

[11] GUNDU A K, HASHMI M S, GROVER A. A new sense amplifier topology with improved performance for high speed SRAM applications [C]// 29th International Conference on VLSI Design and 15th International Conference on Embedded Systems. Kolkata, India: IEEE Press, 2016: 185-190.

[12] JEONG H, KIM T, KANG K, et al. Switching pMOS sense amplifier for high-density low-voltage single-ended SRAM [J]. IEEE Transactions on Circuits & Systems I Regular Papers, 2017, 62(6):1555-1563.

[13] CHOI W, PARK J S, KANG G. Dynamic stability estimation for latch-type voltage sense amplifier[C]// International SoC Design Conference. Jeju, South Korea: IEEE Press, 2014: 218-219.

[14] CHANG M F, SHEN S J, LIU C C, et al. An offset-tolerant current-sampling-based sense amplifier for Sub-100nA-cell-current nonvolatile memory [C]// IEEE International Solid-state Circuits Conference. San Francisco, USA: IEEE Press, 2011: 206-208.

[15] LI Y, WEN L, ZHANG Y, et al. An area-efficient dual replica-bitline delay technique for process-variation-tolerant low voltage SRAM sense amplifier timing [J]. IEICE Electronics Express, 2014, 11(3): 20130992.

第48卷第9期 中国科学技术大学学报 Vol.48,No.9
2018年9月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Sep. 2018

文章编号:0253-2778(2018)09-0703-08

基于ROS和CANopen协议的控制器实时通信系统构建

贾鹏飞[1],王容川[2],徐林森[2],陈丹惠[2],李开霞[2]

(1.中国科技大学自动化系,安徽合肥 230027;2.中国科学院合肥物质研究院,安徽合肥 230059)

摘要:传统机器人软件存在开发效率低和控制器通讯协议不一致等问题,针对这些问题提出用机器人操作系统(robot operation system ,ROS)软件框架来提高软件开发效率;用高性能的工业控制协议CANopen作为控制器通讯方案;借助于模块化机械臂硬件平台以进一步提高软件开发速度;最后构建了基于ROS面向模块化机械臂的CANopen协议控制器通讯软件系统.实验证明,该方案下的通讯软件系统既具有构建过程高效,软件灵活性高的特点,又能满足一般机器人对实时性的要求.

关键词:机器人操作系统;CANopen协议;控制器通讯;进程节点;实时传输

中图分类号:TP391 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2018.09.003

引用格式:贾鹏飞,王容川,徐林森,等. 基于ROS和CANopen协议的控制器实时通信系统构建[J]. 中国科学技术大学学报,2018,48(9):703-710.

JIA Pengfei, WANG Rongchuan, XYU Linsen,et al. Building software system of real-time controller communication by CANopen protocol based on ROS[J]. Journal of University of Science and Technology of China, 2018,48(9):703-710.

Building software system of real-time controller communication by CANopen protocol based on ROS

JIA Pengfei[1], WANG Rongchuan[2], XYU Linsen[2], CHEN Danhui[2], LI Kaixia[2]

(1. *Department of Automation, University of Science and Technology of China, Hefei* 230027, *China*;
2.*Hefei Institutes of Physical Science of Chinese Academy of Sciences, Hefei* 230059, *China*)

Abstract: The traditional robot software is facing problems of low developing efficiency and disunity of protocol of controller communication, aiming at these problems, the robot operating system (ROS) software framework was used to improve the efficiency of software development; the use of high performance industrial control protocol CANopen as the communication scheme of controller; and the help of modular manipulator hardware platform to further improve the speed of software development. Finally, the software system of controller communication by CANopen protocol based on ROS for modular manipulator was constructed. The experimental results show that the communication software system not only has the high developing efficiency and flexibility, but also meets the requirements for the real-time performance of the general robot.

Key words: ROS; CANopen protocol; controller communication; process node; real-time transmission

收稿日期:2017-10-30;**修回日期**:2018-04-10

基金项目:国家自然基金(51405469)资助.

作者简介:贾鹏飞,1993年生,男,硕士生.研究方向:机械臂运动控制,电机控制器通讯. E-mail:1352932725@qq.com

通讯作者:徐林森,博士/研究员. E-mail:lsxu@iamt.ac.cn

0 引言

随着机器人应用范围的不断扩大,机器人软件的开发出现了越来越多的问题.一方面,功能越来越复杂的机器人使得机器人软件的灵活性和开发效率越来越低下;另一方面,各种应用机器人在控制器通信协议上的不一致也使机器人软件的复用性受到了很大限制,因此使用合适的机器软件平台来增加编程效率以及用一个通用的总线通信协议来增加软件复用性十分重要.

机器人涉及的领域日益广泛,不同的应用对机器人的软件系统有不同的功能要求,这也对机器人软件平台提出了不同的应用需求,包括需要有很好的软件灵活性,良好的人性化设置和较高的软件开发效率等.相对于其他机器人软件平台而言,ROS[1]能更好地满足这些要求.ROS 是最新的标准机器人软件框架或平台.通过 ROS,人们能利用数量可观的示例代码和开源软件快速地完成机器人编程与控制[2].

CAN 总线是一种应用于工业自动化控制的现场数据总线[3].CAN 总线有着高可靠性和实时性的特点,这使得它在对系统性能有高要求的场合有良好的表现.从 OSI 的 7 层网络模型的角度来看,CAN 现场总线只定义了物理层和数据链路层,而 CANopen 则补充定义了其应用层.目前,CANopen 协议已经在运动控制、车辆工业、轨道交通、电机驱动、工程机械、船舶海运等行业得到了广泛的应用.

文献[4]基于 ROS 构造了新型水下机器人控制系统,并使用-RS232 作为机械臂控制器通信串口,相对于 RS232 串口,CAN 总线在机器人控制器网络环境中更能有效支持实时性控制.文献[5]针对一般 Linux 系统上 ROS 固有的实时性问题提出 RGMP-Linux 和 ROS 混合机器人操作系统并验证其可靠性,但没有考虑高性能总线通信协议对 ROS 控制实时性的影响.

文献[6]在机器人控制系统上应用了 CANopen 协议,证明 CANopen 协议能使通信系统具有较低的延迟和总线负载率;文献[7]基于 CANopen 协议在四足行走机器人平台上构建了一个闭环实时控制网络并证明了方案的有效性和可靠性.以上两个文献没有使用可靠的软件平台,构建的系统缺乏灵活性和复用性.

本文首先提出利用 ROS 软件框架使构建系统的过程更加高效,构建的软件更加具有模块化特性和高复用性;并结合使用高性能的控制器通信协议 CANopen 来提高整个系统的实时性能.

1 ROS 和 CANopen 协议的基本概念和特点

ROS 是一个分布式的具有结构化通信层的软件框架.同时 ROS 软件库也集成了大量用于机器人应用的软件工具和源代码;CANopen 协议的设计很好地采用了分层和面向对象的设计理念并在少量节点和短途通讯领域有着实时性优势.

1.1 ROS 基本概念和特点

ROS 是一种现阶段被广泛应用的机器人操作与控制系统软件框架.ROS 使用了当前流行的面向服务的软件技术,它通过网络协议将节点间的数据通信解耦,因此 ROS 能高效地集成不同语言和不同功能的代码.ROS 也支持一种类似于代码储存库的联合系统,该系统从另一个角度实现了工程的协作及发布.该设计能够让一个项目的开发和实现从文件系统到用户接口完全独立决策[8].

ROS 为软件提供了一个点对点的通信方式,简化了程序间通信的编程.为了实现进程之间的通信机制,ROS 使用了两个抽象:节点和主题.节点可以看作一个软件模块或者是正在执行运算任务的一个进程;主题则是传递进程之间消息的通道,一个系统有很多节点组成.

当许多节点同时运行时,可以很方便地将端对端的通讯绘制成一幅图.如图 1 所示,正在运行的某个进程可以看作图中的节点,而进程间的联系则可以看成节点间的箭头连接.每一个消息都要发送到对应的主题,一个节点的发送数据等价于该节点正在向主题发布消息.一个节点也可以订阅某个主题,而不需要另一个节点同时发布这个主题.发布者和订阅者相对独立,在这里主题可以被看成是消息的暂存区.

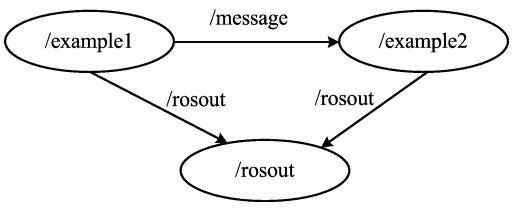

图 1 ROS 抽象:节点与主题

Fig.1 ROS abstract: Note and theme

1.2　CANopen 基本概念和特点

CANopen 协议从总体上分为通讯对象、对象字典和应用对象 3 层，CANopen 具体的通讯结构如图 2 所示.

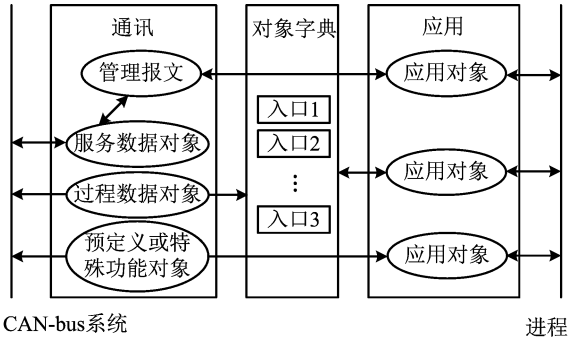

图 2　CANopen 协议通信结构

Fig.2　Communication structure of CANopen protocol

3 层结构中的最复杂的是对象字典，它使用索引号描述了应用对象和 CANopen 报文之间的关系. CANopen 通讯层是最重要的部分，主要内容是 CANopen 协议的通信规则，这部分内容是开发者需要重点关注的. 用户应用层是外部应用的用户依照实际的需求编写的应用对象.本文实验使用服务数据对象 SDO 来传递网络系统中的配置信息和机械臂的关节角度信息，SDO 报文发送快速、传送可靠. SDO 报文发送的过程遵守握手协议，报文发送方需要知道接收方的确认信息才能继续进行[9].

CANopen 的创始人在设计 CANopen 时，对其定义为小网络控制信号的实时通信.满足这些特点的设定包括：报文传输采用 11 bit 的 ID 域，以尽量减小传输时间；网络控制报文均采用数据最小字节数.比如心跳报文，只有 1 个字节数据；实时更新的过程数据无需接收方报文应答，采用生产消费模型，降低总线负载；需要接收方确认的配置参数一般都采用快速单字传输，1 个报文最多传送 1 个 32 字节的参数变量，避免了分帧引起的实时性降低.

CANopen 协议在少量节点和非长途通讯场合有良好的表现.CAN 总线应用最多的是车辆控制工业，其网络中的通讯节点数量和传输的通讯的信息都很少.在这些场合人们选择 CAN 而非 RS485，主要是因为 CAN 有随时发送的实时性优势.在机器人控制领域中，单机器人的各通讯部件在地域上比较集中，而且通讯部件的数量一般在几个到几十个之间.这些特点使得 CANopen 协议在理论上能够很好地发挥实时性的优势.

2　基于 ROS 模块化机械臂控制器 CANopen 协议实时通讯软件

机器人机械臂的关节运动控制是机器人技术中的热门，为了让运动控制软件和关节电机硬件细节解耦需要一个软硬件通信模块作为中介.为通讯模块选择合适的协议以及为包括通讯和运动控制的整个软件选择良好的平台具有重要意义.本文的软件系统面向的是模块化机械臂，由于模块化机械臂的各关节构型完全相同，这使得伺服电机的通讯过程完全相同，很大程度上简化了软件的开发过程并且增加了软件模块的复用性.本文将以单个关节作为软件的操作对象，介绍软件系统的构建过程.

2.1　硬件平台

本文的硬件使用雄克模块化机械臂系统，如图 3 左，该 6 自由度机械臂系统由 3 个相同构型的关节和 2 个连杆连接而成，通过改变关节的数量和连杆的参数可以很方便地产生不同用途的机械臂.单个关节的构型如图 3 右，该关节拥有两个自由度[10].

图 3　6 轴模块化机械臂以及单关节构型

Fig.3　6-axis modular robotic arm and single joint configuration

关节的主体部分是一个球形连接器，球体内置有两个伺服电机、两个电机驱动器、两个码盘和一个电子控制单元.关节转盘由电机来驱动旋转，两个转盘的转轴互相垂直.此外，该球形关节制作精密、结构稳定，耐用性也很好.目前该型号关节的转轴转角范围大约为 $-170°\sim+170°$，最高转速可达 72°/s.

2.2　硬件系统

本实验的硬件组件图如图 4 所示，整个硬件系统由主控计算机，PCAN-USB，球形关节内部的控制单元，电机驱动器，伺服电机和码盘组成.主控计算机采用 Intel Core i3 双核处理器，时钟频率为 3.4 GHz，拥有 4 G 内存和 1 T 硬盘空间.以上配置满足本次实验的计算配置要求.主控计算机和球形关节内的控制单元由 PCAN-USB 连接，PCAN-

USB 能将 CAN 接口变换为 USB 接口，它使接入 CAN 网络非常容易.本实验用的 PCAN-USB 的波特率高达 1 Mbit/s，并同时支持 CAN2.0A 和 CAN2.0B 协议.PCAN-USB 和主控计算机之间用 USB 相连，和关节控制单元用 CAN 总线相连.

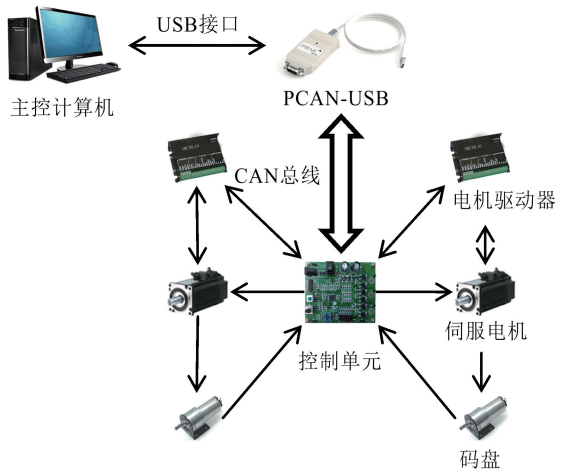

图 4 硬件组件图

Fig.4 Hardware components

球形关节内一共有一个控制单元和两个伺服电机子系统，一个伺服电机子系统包含一个码盘、伺服电机和电机驱动器.控制单元负责将主控计算机发送过来的指令转发至电机和电机驱动器或者将电机和码盘传感器的状态信息传送至主控计算机.电机驱动器通过将主控计算机的指令转化为位置、速度或力矩信息来实现对电机的精确控制.

2.3 软件系统

基于 ROS 的 CANopen 协议通信软件系统包括命令生成模块、报文发送模块、报文解析模块和数据处理模块.CANopen 协议的内容体现在报文发送和报文解析模块中.由于 PCAN-USB 的厂商提供了基本的 CAN 接口，软件的构建就可以专注于 CANopen 协议而不用关心 CAN 基本协议的细节.

2.3.1 软件系统的架构

整个系统的架构图如图 5 所示(图 5 罗列了与本文软件系统相关的重点部件及其相互关系，非重点部件没有指明)，整个系统可以抽象地看成 3 个层次.最外层是承载整个软件系统的硬件，紧靠着硬件的层次是 Linux 操作系统层，最里面一层就是 ROS 软件层.使用的 ROS 版本号是最新的 Kinetic，Linux 版本是 Ubuntu16.0.

ROS 软件层是本文系统的最高层，旨在 ROS 软件框架下实现机器人控制器通讯程序.ROS 通过

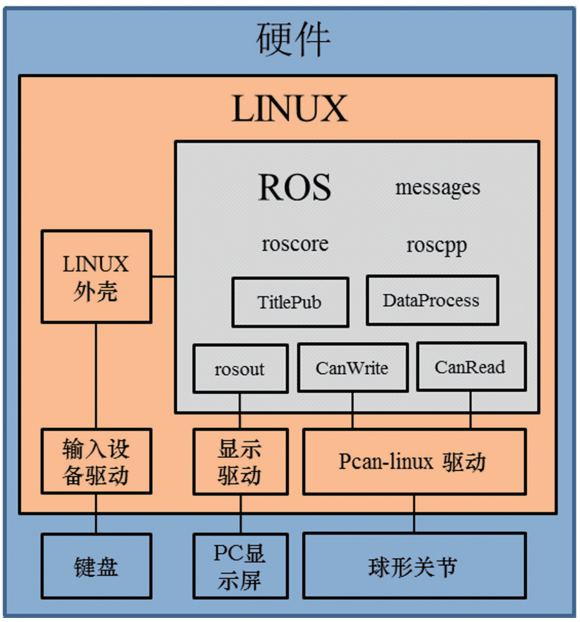

图 5 软件系统架构图

Fig.5 Software system framework

不同节点的相互协作来实现控制器通讯功能.本文实现的主要节点包括 TitlePub，DataProcess，CANwrite 和 CANread.其中 TitlePub 和 CANwrite 节点一起协作完成将指令传递至球形关节控制器来控制电机运作的任务.报文内容的解释由 CANread 节点提供，而 DataProcess 节点在 CANread 对报文分析的基础上进一步处理来完成关节状态的记录和测试任务.ROSout 节点是 ROS 的自带节点，它为其他节点提供了将消息显示在 PC 屏幕上的服务.ROScore，messages 和 ROScpp 集成了大量机器人软件开发所需的库函数，工具和服务，他们作为 ROS 软件框架的一部分为用户自定义的节点提供基础.

Linux 操作系统层位于 ROS 之下，是 ROS 所有节点的运行环境，为 ROS 提供所需要的工具和服务.工具包括有显示屏、外部输入设备和球形关节 CAN 总线的驱动程序.此外，Linux 还提供 shell 程序以让 ROS 能接受外部命令，在本文的软件中，实验人员通过 Linux 命令行启动 ROS 的节点并完成相关功能.

底层就是上文所描述的硬件系统，包括计算机和球形关节，这里便不再详述.

2.3.2 软件系统的 ROS 节点结构

本文软件系统的主要节点结构和主题如图 6 所示.图中 myBot 是本软件系统的工程包名，以该包名开头的节点和主题属于本文研究的软件系统.其

他 ROS 节点由于和本文研究重点无关，不再介绍.

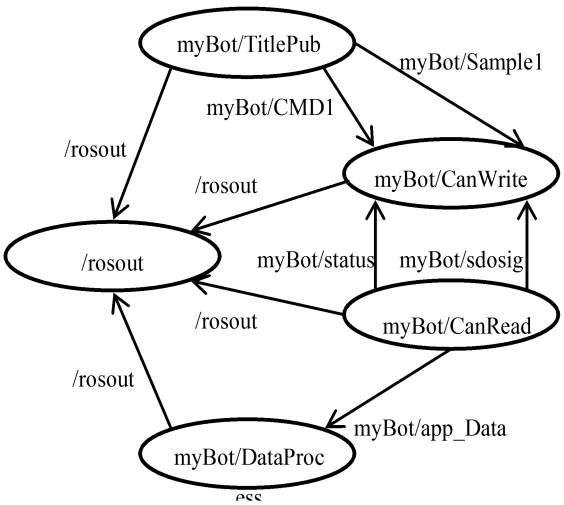

图 6　软件系统主要节点结构和主题

Fig.6　Main node structure and theme of software system

（Ⅰ）命令生成（TitlePub）

命令生成节点是最高层的进程，它的主要作用是根据操作人员的输入产生电机的宏观控制命令并以字符串的形式发送至报文发送模块，后者将进一步解释控制命令并产生目标 CAN 报文.宏观电机控制命令包括电机在每个时刻的位置、速度、码盘信息采样的时间以及采样频率.图 6 中的 myBot/TitlePub 表示命令生成节点，它使电机的转动速度或位置命令发送至 myBot/CMD1 主题，将码盘采样时间和频率发送至 myBot/Sample1 主题.这里的一个主题对应关节中的一个轴或自由度，如果要加上其他轴或者更多的模块化关节则可以添加主题 myBot/CMD2，myBot/Sample2，以此类推.

（Ⅱ）报文发送（CANwrite）

报文发送节点实现了 CANopen 协议的主要部分，它是命令生成的下一层进程，该节点根据 CANopen 协议解释命令生成节点中产生的命令字符串，产生 CAN 报文并写入 CAN 总线.写入的 CAN 报文将直接执行对电机的控制.报文发送节点的名称为 myBot/CANwrite，是 my-Bot/CMD1 以及 myBot/Sample1 主题的订阅者.通过这种方式，报文发送节点就可以接收上层控制命令并完成解释工作.此外，报文发送节点还订阅了 myBot/status 和 myBot/sdosig 主题，前者是电机的状态信号，后者是 SDO 报文的接收确认信号.报文发送节点分别根据这两个信号来诊断电机状态和完成报文的发送.

（Ⅲ）报文解析（CANread）

报文解析节点的主要作用是读取 CAN 总线里面的报文；然后根据协议对报文的内容进行解析.它和报文发送节点实现了完整 CANopen 协议.myBot/CANread 节点就是报文解析节点.报文解析节点可以认为是一个选择器，根据报文的不同信息进行区别处理.如果是一个系统状态信息报文，则解析出状态并发送至对应主题 myBot/status；如果是一个 SDO 报文，则解读出报文的发送状态并发送至 myBot/sdosig 主题.进一步，如果该报文的发送状态显示成功，则把该报文的数据提取出来发送至 myBot/app_data 主题以对其处理.

（Ⅳ）数据处理（DataProcess）

数据处理节点的功能是接受报文的原始数据流并进行后期处理.在本文的系统中后期处理包括将报文中的速度数据按照整型格式显示到计算机屏幕上；然后将数据保存至打开的 txt 文件中以便之后使用数学软件 Matlab 对其进行分析处理.myBot/DataProcess 节点订阅了报文解析节点发布的 myBot/app_data 主题，并通过/ROSout 主题在屏幕上显示数据.

该节点结构的关键在于节点功能的分层，通过让报文发送和命令生成分解控制器指令的发布流程，就可以让 CANopen 协议的实现细节和用户发布指令的格式习惯分开，使程序更利于模块化.由于报文解析节点涉及 CANopen 协议的确认信号和电机关键状态信号，通过让数据处理节点分担报文解析节点的大部分数据处理功能可以让报文解析更加简练快速，从而使得上层应用程序更快地获得关键信号，提高系统的整体响应速度.如果要实现整个模块化机械臂的控制器通讯，只需要向图 6 所示的节点结构中加入更多的和 CMD1，Sample1 同类型的主题并且在报文发送节点中加入和主题对应的消息处理函数即可.

2.3.3　CANopen 通讯过程的实现

CANopen 协议通信依靠上文所述的报文发送和报文解析节点的相互合作来实现.图 7 以伺服电机的标准位置模式[11]为例说明了本文软件系统在实现 CANopen 协议通信过程中的节点交互情况.图中球形关节对象字典是系统唯一的硬件层.图中用浅灰色（黄）标识的方框表示 ROS 中的过程函数，灰色（蓝）标识的方框表示过程函数里面的下一层操作.黑色的虚线箭头表示通过共享内存的间接通信，灰色实箭头表示直接的消息发送和调用动作，在灰色箭头上的冒号是主题名称和具体消息的分隔符：主题：具体

消息.

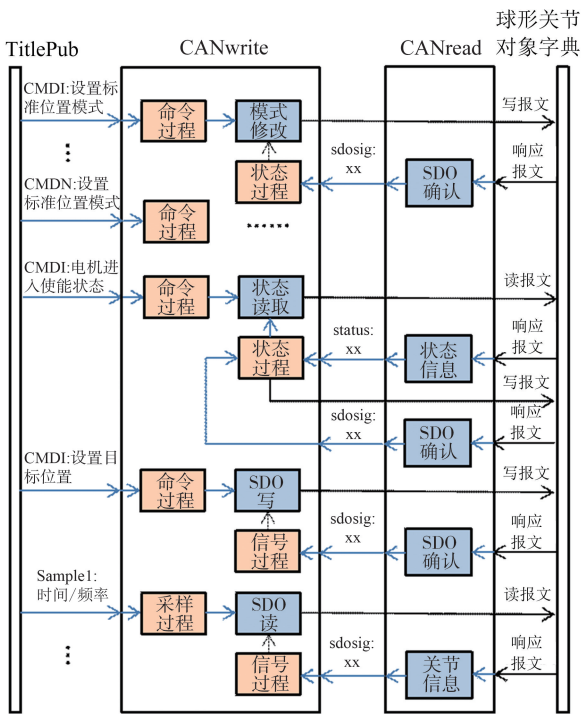

图 7 ROS 节点在运行位置模式时的交互图

Fig.7 Interaction diagram of ROS nodes in operation location mode

通过 CMD1,CMD2 等主题,命令生成节点向报文发送节点传送电机控制指令,每个 CMD 主题都对应了机械臂上的一个轴或一个自由度.每个 CMD 主题中的消息到达 CANwrite 后,CANwrite 都会为其创建一个新线程.线程的管理由 ROS 自带函数工具 ROSspinner 完成.这样就使得对多个轴的控制实现并行控制,满足其实时性需求.以上几点对于 Sample 主题也是一样的.ROS 节点会对每个主题产生一个过程处理函数,在本文的系统中这相当于一个线程的主函数.本文的 CANwrite 节点对应于不同的主题有 4 种不同的操作过程:命令过程的功能是产生主要的电机控制动作;信号过程用来处理 SDO 报文协议的确认信号;状态过程用来应对关节的状态信息;采样过程用来获得关节的周期运动状态.

电机标准位置模式可以在 CANopen 子协议 DSP402 手册中查看.所有的模式包括标准位置模式都有很多共同操作,比如模式位设置,电机状态切换,运动参数修改.这些共同操作构成了本文 CANwrite 节点的主要内容,上述电机控制操作包含在命令过程和状态过程中,命令过程是一个选择器,通过 CMD 主题中的控制字符串选择不同的具体操作,比如模式修改和状态读取等.状态过程是一个状态机,它会实时保存目前电机的状态并自动向电机发送控制字符切换状态到要求的状态.所有过程具体操作的最主要功能就是根据协议发送一些 S-DO 的写或者读报文.报文发送至对象字典后,对象字典返回的报文会直接由 CANread 节点读取并处理,报文的解析工作将随之完成.CANread 节点的主要功能也是一个选择器:按照 CANopen 协议对报文数据进行功能的划分,这些功能包括 SDO 的确认、状态读取和关节信息.CANread 和 CANwrite 一起实现了 CANopen 协议的主要部分.

3 软件系统的各项特性评估

下面分析并评估了软件系统的实时性、开发效率、灵活性、健壮性和安全性.一些特性可以通过详细的实验进行评估,另外一些则通过对系统结构的分析得到.

3.1 实时性

本文实验包括运行软件、绘制电机速度跟踪曲线,然后分析软件系统的实时性能.

3.1.1 实验过程

软件系统中的命令生成节点可以通过发送一些控制字符串来通过报文发送节点间接控制电机动作.一个控制字符串对应了一个电机控制动作.本文实验通过对命令生成节点编码创建了一个对电机的控制流程,实验过程的控制流程图如图 8 所示.

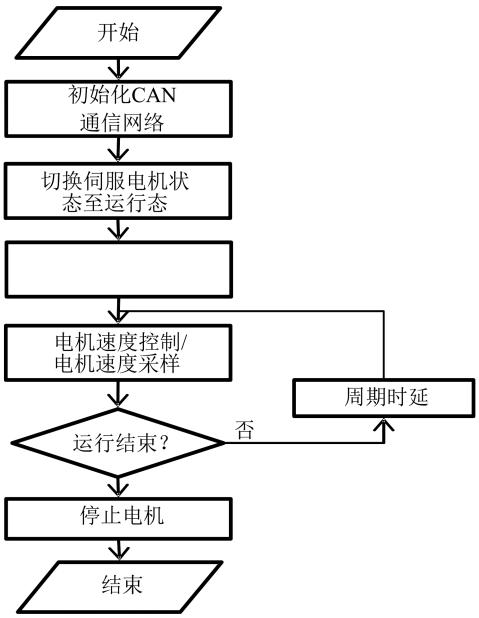

图 8 总体实验流程图

Fig.8 Overall experimental process

首先用程序对 CAN 网络进行初始化,初始化