

基于 Modbus 的双环路冗余总线控制器的设计与实现

黄文庆¹ 李春杰^{1*} 李伟华²

¹(中国科学技术大学软件学院 江苏 苏州 215123)

²(苏州博睿测控设备有限公司 江苏 苏州 215143)

摘要 针对传统过程控制系统的 PLC/DCS 对终端执行机构实时监测与控制导致系统资源消耗沉重,单点设备故障容易引发整个系统通信瘫痪等工业问题,为增强系统通信容错能力,使 PLC 与终端执行机构稳定可靠地连接,设计并实现基于 Modbus 的双环路冗余总线控制器。控制器内部采用双 CPU 冗余热备份设计,有效降低 PLC 的工作负担,提高了系统的鲁棒性和实时性,适用于工业现场。

关键词 Modbus 总线控制器 冗余

中图分类号 TP3

文献标志码 A

DOI:10.3969/j.issn.1000-386x.2024.11.039

DESIGN AND IMPLEMENTATION OF DUAL-LOOP REDUNDANT BUS CONTROLLER BASED ON MODBUS

Huang Wenqing¹ Li Chunjie^{1*} Li Weihua²

¹(School of Software Engineering, USTC, Suzhou 215123, Jiangsu, China)

²(Suzhou Bonray Measurement and Control Equipment Co., Ltd., Suzhou 215143, Jiangsu, China)

Abstract The PLC/DCS real-time monitoring and control of the terminal actuator in the traditional process control system leads to heavy system resource consumption, and the failure of a single of equipment is easy to cause communication breakdown of the whole system and other industrial problems. To enhance the communication fault-tolerant ability of the system and make PLC and terminal actuators connect stably and reliably, a dual-loop redundant bus controller based on Modbus was designed and implemented. The controller adopted dual CPU and the design of redundant heat backup, which effectively reduced the PLC workload, improved the system robustness and real-time performance, and was suitable for industrial field.

Keywords Modbus Bus controller Redundancy

0 引言

在传统使用 Modbus 总线通信的过程控制系统中,系统的鲁棒性和实时性是两项最重要的性能指标^[1-2]。系统内的各个设备在恶劣环境中保持长期稳定的工作状态反映系统的鲁棒性。实时监测生产设备状态和控制终端执行机构反映系统的实时性。

在实际生产过程中,上位机(PLC/DCS)一般都是作为 master 节点,采用轮询-等待的方式周期性扫描每个 slave 节点(终端执行机构),监测和采集每个 slave

节点的设备状态。在连续运行而不能停顿的工业生产中,这种传统的工作方式,一方面系统资源消耗沉重,周期性地轮询任务占用 PLC 较多的 CPU 资源,不能充分发挥其强大的控制优势^[3],同时系统的实时性也随之下降;另一方面,随着生产规模的不断扩大,挂载在总线上的终端执行机构的数量越来越多,总线负荷增加,一旦总线出现异常,将导致整个系统通信的瘫痪。

基于上述原因,本文设计并实现了一个基于 Modbus 的双环路冗余总线控制器,安装在上位机与终端执行机构之间,提高系统的鲁棒性和实时性。

1 Modbus 通信协议

Modbus 是 master/slave 架构, 请求/应答式的协议^[4-5]。其物理层采用 RS-485 总线时, 只有一个设备可以作为 master 节点, 其余挂载设备都只能是 slave 节点, 每个 slave 节点都分配有唯一的地址。master 节点向总线发出指令, 总线上的所有 slave 节点都能够接收到, 但是只有指定地址的 slave 节点会执行动作并返回响应。若是广播地址(地址 0), slave 节点无须返回响应。

Modbus 协议在串行链路上的通信格式如图 1 所示。ADU 包含 slave 地址、PDU 和差错校验。PDU 包含功能码和数据域, 其中请求帧中 Data 域的前两个字节表示寄存器起始地址的高位和低位。

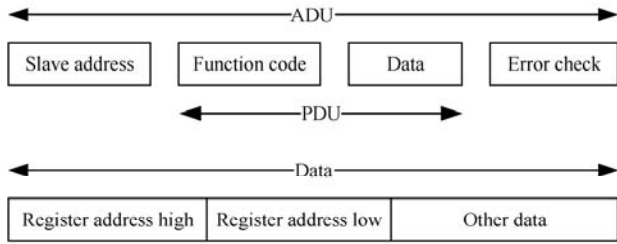


图 1 Modbus 通信格式

2 系统架构

系统架构如图 2 所示。传统的过程控制系统主要分为 PLC/DCS 和终端执行机构两部分。现在将总线控制器安装在 PLC 与终端执行机构之间。总线控制器周期性轮询总线上挂载的终端执行机构, 监测和采集终端执行机构的工作信息, 并且记录保存采集到的信息。PLC 作为上位机, 可以随时读取总线控制器记录的终端执行机构信息或者向总线控制器下发控制指令, 总线控制器经过地址转换之后再转发给目标终端执行机构, 执行相应的动作。系统采用 Modbus 双环路冗余通信机制^[6], 总线控制器在内部采用双 CPU 设计方式, 一个作为工作单元, 另一个作为冗余单元充当“热备用”。工作单元和冗余单元分别拥有两套 485 总线接口。两套 485 接口与终端执行机构采用环形连接方式, 一个作为工作通道, 另一个作为冗余通道, 两路通道构成总线环路。当工作通道出现短路、断路等异常时, 立刻切换到冗余通道继续保持通信。工作单元与冗余单元之间维持有心跳机制, 当工作单元发生异常, 冗余单元立刻切换为工作状态代替工作单元继续工作。由于工作单元与冗余单元功能、架构完全相同, 所以本文在以下内容中只介绍总线控制器的工作

单元, 冗余单元对照参考即可。

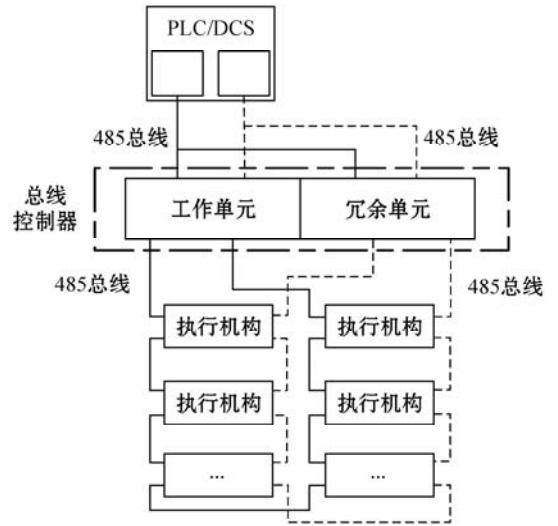


图 2 系统架构

3 硬件设计

总线控制器的主控芯片采用基于 ARM Cortex-M4 内核的 STM32F429VET6 微处理器。该芯片具有丰富的外设资源, 满足工业领域对数据采集、处理和中断响应性能的要求。RS-485 总线接口芯片采用 ADI 半导体公司生产的 ADM2486 收发器, 该芯片具有限流和热关断等优良特性。

总线控制器的工作单元硬件电路示意图如图 3 所示。以 STM32 作为主控芯片, 两路 USART 通过 RS-485 总线接口与上位机连接, 两路 USART 通过 RS-485 总线接口与终端执行机构连接。为确保 RS-485 总线端口能够在复杂的电磁环境中正常运行, 通信线路不受 ESD、EFT 和 Surge 的影响^[7], 本文设计了 RS-485 端口 EMC 保护模块。另外还有电源模块、键盘及液晶显示屏模块等。

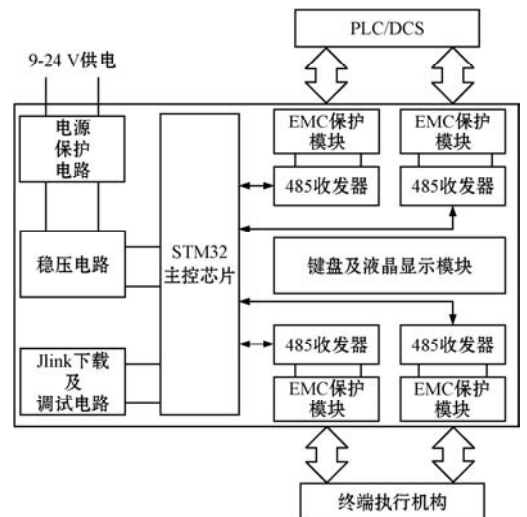


图 3 总线控制器工作单元硬件电路示意图

4 软件设计

4.1 软件架构

软件采用分层架构的设计方式^[8],总线控制器的软件架构如图 4 所示。本文采用国产的嵌入式实时操作系统 RT-Thread^[9-10] 对上层软件进行管理调度,提高代码的可移植性。软件整体架构从下至上分为驱动层、系统层、组件与服务层和功能层四层。其中,功能层负责完成用户交互任务和对接 PLC 进行监测与控制;组件与服务层主要是为上层提供具体服务,通过建立多个服务进程完成上层的指令;系统层包含操作系统与 BSP,位于上层软件与底层硬件之间,负责管理底层硬件资源。驱动层为 RT-Thread 控制 STM32 MCU 提供软件驱动接口。

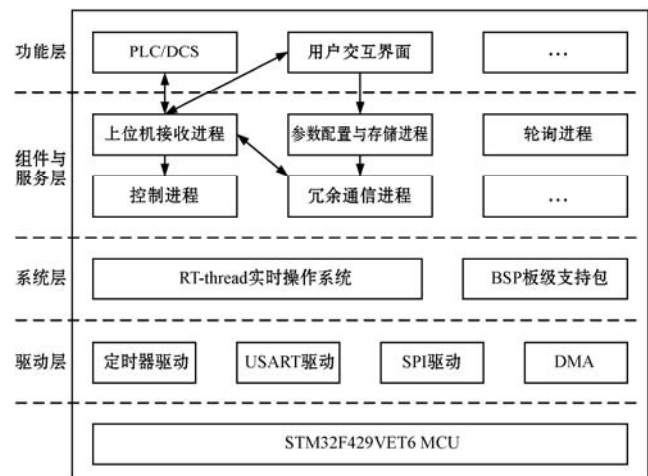


图 4 总线控制器软件架构

软件设计采用模块化的设计思想^[11],将应用程序分解成上位机接收进程、参数配置与存储进程、轮询进程、控制进程和冗余通信进程五个用户进程,最终实现控制器的软件设计。总线控制器上电开机,RT-Thread 在完成初始化之后,首先建立的第一个用户进程是参数配置与存储进程。参数配置与存储进程创建交互界面,用户通过交互界面配置总线控制器的各项通信参数、设备地址和丢信超时时间等。完成参数配置并且存储正确之后,参数配置与存储进程将创建其他四个进程。其中上位机接收进程主要负责接收来自 PLC 上位机或者用户通过交互界面发送的指令,若是状态获取指令就直接返回状态信息;若是控制指令,则将指令数据通过消息机制发送给控制进程,由控制进程控制终端执行机构执行相应的动作;若一直未收到来自 PLC 上位机的指令数据并且超过用户配置的超时时间,则向冗余通信进程发送状态异常消息。轮询进程主要负责轮询众多的终端执行机构,获取设备状态信

息。冗余通信进程主要负责工作单元与冗余单元之间周期性地数据交换,数据包包含参数信息和状态信息。并且能够根据状态信息是否异常自动完成切换,实现热备份功能。

4.2 参数配置与存储进程

参数配置与存储进程的工作流程如图 5 所示。用户可通过交互界面配置总线控制器的各项参数。用户在完成配置之后,总线控制器会将数据保存在 STM32 的内部 Flash 中。待参数存储正确之后,参数配置与存储进程将创建系统中其他进程,并且向冗余通信进程发送参数消息,由冗余通信进程将参数配置信息发送给冗余单元。这样用户只需要在工作单元上配置好参数,冗余单元的参数配置会自动完成。最后总线控制器开始运行。

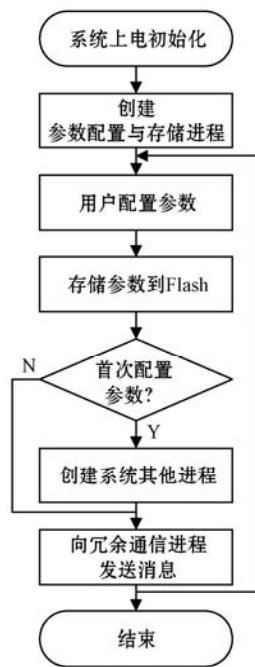


图 5 参数配置与存储进程流程

4.3 上位机接收进程

一方面,总线控制器与 PLC 通过 485 总线相连,PLC 是 master 节点,总线控制器是 slave 节点。另一方面,总线控制器与终端执行机构通过 485 总线相连,此时总线控制器是 master 节点,众多的终端执行机构是 slave 节点。因此 PLC 向总线控制器发送指令获取某个终端执行机构的信息或者控制某个终端执行机构的设备动作时,ADU 中的地址域表示总线控制器的地址,并非终端执行机构的地址。为解决此问题,本文使用 Modbus 请求帧中 Data 域的寄存器起始地址,建立与终端执行机构真实地址的关系映射表,映射关系如图 6 所示。

mFunction code = Function code;

mRegister address \longleftrightarrow ((Slave address, Register address)

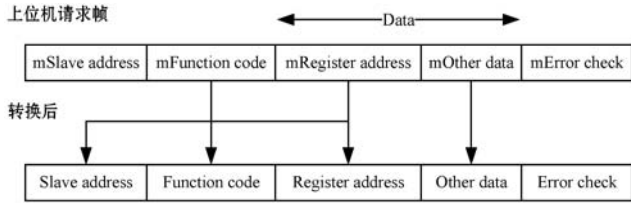


图6 地址映射关系

上位机接收进程的工作流程如图7所示。上位机接收进程在被创建之后,会开启一个总线丢信超时定时器。若在超时时间之内收到来自 PLC 上位机发送的指令,则表示 PLC 工作正常和总线线路正常,定时器重新清零计时;若没有在丢信超时时间之内接收到指令,则通过消息机制向冗余通信进程发送状态异常,由冗余通信进程根据冗余单元返回数据判断是否切换工作状态。

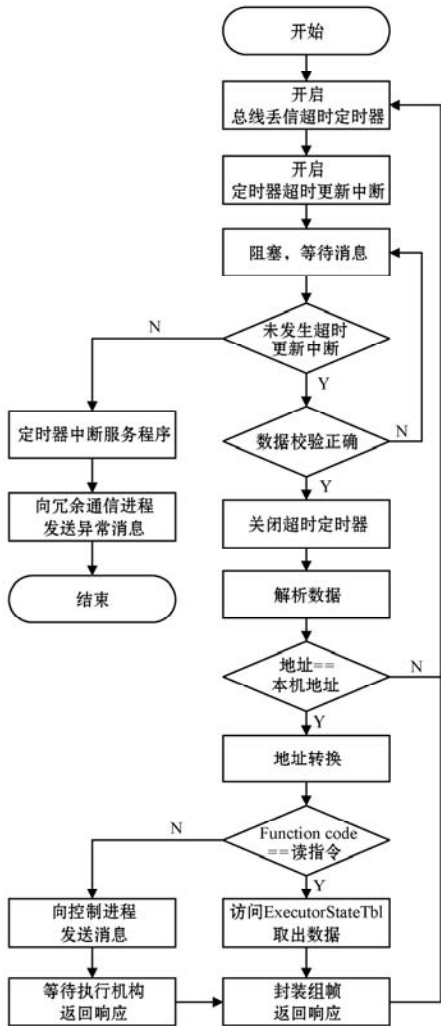


图7 上位机接收进程流程

上位机接收进程接收到数据之后,首先对数据帧进行 CRC 校验,若校验错误,直接丢弃。若校验正确就关闭超时定时器,解析数据帧获得 slave 地址、功能码等字段。若 slave 地址是本总线控制器地址,说明这一帧数据是发给本控制器的,则进行地址转换,并根据功能码执行相应动作。若功能码 Function code 是读指

令(F01/F02/F03等),就直接从执行机构设备状态表(ExecutorStateTbl)中取出状态数据,封装组帧并返回响应,重启定时器。若功能码是控制指令(F05/F06/F15/F16等),将此帧数据通过消息机制发送给控制进程。由控制进程将数据帧发送到485总线上,等待指定 slave 地址的终端执行机构执行相应动作并返回响应,然后控制进程向上位机接收进程返回响应,上位机接收进程最后封装组帧向 PLC 上位机返回响应,重启定时器。

4.4 控制进程

控制进程的工作流程如图8所示。控制进程在没有收到消息时会保持阻塞。当上位机接收进程收到来自 PLC 上位机或者用户通过交互界面现场发送的控制指令时,都会通过消息机制向控制进程发送消息,由控制进程执行具体的控制操作。由于轮询进程和控制进程都使用485总线向执行机构传输数据,因此存在总线控制权互斥使用问题。为解决此问题,本文利用 RT-thread 操作系统的互斥信号量机制,轮询进程和控制进程在每次向总线发送数据之前都必须申请到互斥信号量 mutex,这样可保证不会出现总线冲突问题。因此在控制进程收到消息之后,首先申请 mutex,申请到 mutex 之后,解析出功能码(F05/F06/F15/F16等),然后根据功能码类型重新组帧,向下发送到485总线上,开启响应超时定时器,并等待终端执行机构返回响应。若在超时时间之内收到正确响应则释放互斥信号量 mutex,返回响应数据。若定时器超时仍未收到响应数据,则切换通道,通过冗余通道重新发送。若工作通道与冗余通道都没有接收到响应数据,则释放互斥信号量 mutex,返回异常响应。

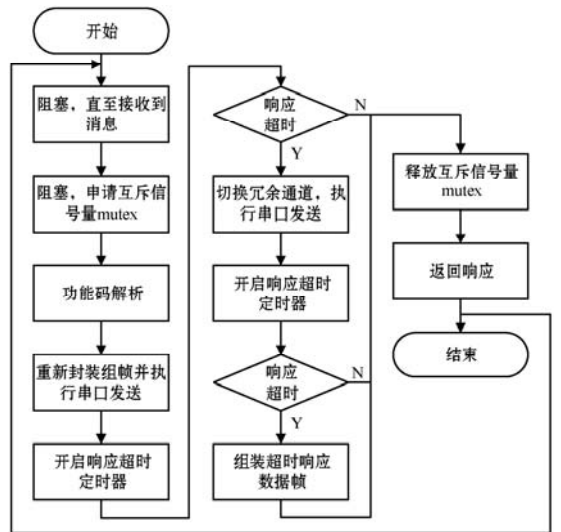


图8 控制进程流程

4.5 轮询进程

轮询进程的主要任务是周期性地读取每个执行机

构的设备状态,将设备状态记录在执行机构设备状态表(ExecutorStateTbl)中。若状态异常,则记录当前执行机构的地址为断点位置,并在交互界面上显示异常提示信息。轮询进程的工作流程如图 9 所示。

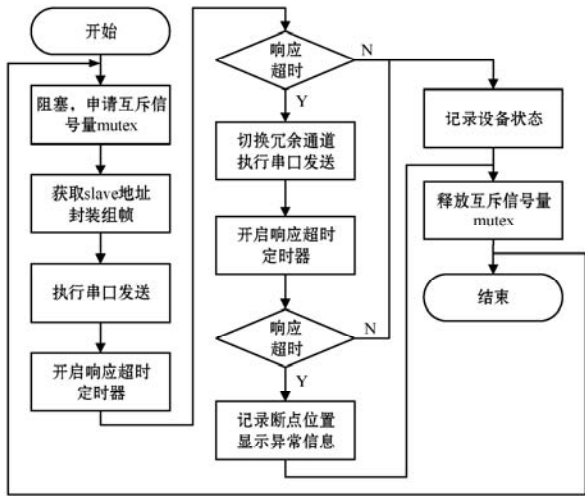


图 9 轮询进程流程

轮询进程在被创建之后就一直周期性地轮询总线上众多的终端执行机构。在每次轮询向 485 总线发送状态获取请求帧之前,轮询进程都必须申请到互斥信号量 mutex,这样可以保证不会与控制进程发生总线控制权冲突问题。获得 mutex 之后,将执行机构的地址作为本次查询的 slave 节点地址封装到 Modbus 请求帧中,然后发送到 485 总线上,开启响应超时定时器并等待响应。若在超时时间之内收到正确响应则释放 mutex,解析返回的响应数据,将状态信息记录到 ExecutorStateTbl 中。若定时器超时仍未收到响应数据,则切换通道,通过冗余通道重新发送。若工作通道与冗余通道都没有接收到响应数据,则释放 mutex,记录本次轮询的执行机构地址为断点位置,并在交互界面上显示异常提示信息。然后按照执行机构的地址顺序,继续重复上述轮询过程。

4.6 冗余通信进程

冗余通信进程是在总线控制器的工作单元 CPU 与冗余单元 CPU 之间传递数据,通信格式如图 10 所示。本文使用结构体内嵌共同体的方式定义实现该通信帧格式,由于编译器会对结构体自动进行内存对齐,容易导致内存使用率下降,因此代码中将未使用到的字节作为 reserved 保留字节,以备日后程序升级迭代使用,代码定义如下:

```

typedef struct {
    UCHAR head;           //帧首部字节
    UCHAR funcCode;      //功能码/识别码
    UCHAR length;        //数据字节长度
    UCHAR reserved1;     //保留位 1
    UCHAR reserved2;     //保留位 2
    UCHAR reserved3;     //保留位 3
    USHORT crc;          //数据校验位
    union {
        KeepAlive keepAliveParam;
        MasterDevParam masterDeviceParam;
        SlaveDevParam slaveDeviceParam;
    };
} RedundancyFrame;
    
```

```

UCHAR reserved2;           //保留位 2
UCHAR reserved3;         //保留位 3
USHORT crc;               //数据校验位
union {
    KeepAlive keepAliveParam;
    MasterDevParam masterDeviceParam;
    SlaveDevParam slaveDeviceParam;
};
} RedundancyFrame;
    
```

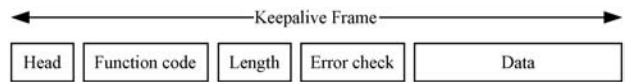


图 10 冗余通信格式

功能码 funcCode 决定冗余通信进程传递的数据帧类型,数据长度和数据内容。帧类型主要分为 keepAliveParam、masterDeviceParam、slaveDeviceParam 三种。keepAliveParam 表示心跳数据,包含状态信息和断点位置信息;masterDeviceParam 表示 PLC 上位机总线通信参数数据;slaveDeviceParam 表示终端执行机构的总线通信参数数据。

冗余通信进程的工作流程如图 11 所示。工作单元与冗余单元周期性地互相发送与接收冗余通信帧。冗余通信进程每一次发送数据之前,都检查一下消息队列中是否有来自参数配置与存储进程或者上位机接收进程的消息。如果没有消息就默认此帧类型是 keepAliveParam 类型,状态位 state 正常。如果消息队列中有消息,则根据 funcCode 判断帧类型。如果是 keepAliveParam,表示接收到了来自上位机接收进程的消息,标志着上位机接收进程的总线丢信定时器已经超时,此时更改 state 为异常,封装组帧,将此帧数据发送给冗余单元,并等待冗余单元返回的数据。如果冗余单元返回的 keepAliveParam 状态位是正常,表示其上位机接收进程的总线丢信定时器未超时,则工作单元主动放弃运行,由冗余单元代替工作单元继续运行。否则,不发生切换,工作单元继续运行。同理,冗余单元在接收到工作单元的超时异常帧后,首先判断自身的总线丢信定时器是否超时,如果超时,则不切换;否则,冗余单元立刻切换成工作单元继续运行,同时报警提示。切换的触发条件如表 1 所示。

表 1 工作状态切换的触发条件

| 工作单元总线丢信定时器 | 冗余单元总线丢信定时器 | 是否切换 |
|-------------|-------------|------|
| 超时 | 超时 | 否 |
| 超时 | 未超时 | 是 |
| 未超时 | 超时 | 否 |
| 未超时 | 未超时 | 否 |

如果 funcCode 是其他两种类型,则表示收到的是参数配置消息,只需封装组帧,然后将此帧数据发送给冗余单元,等待冗余单元返回 ACK。冗余单元收到后自动完成参数配置。

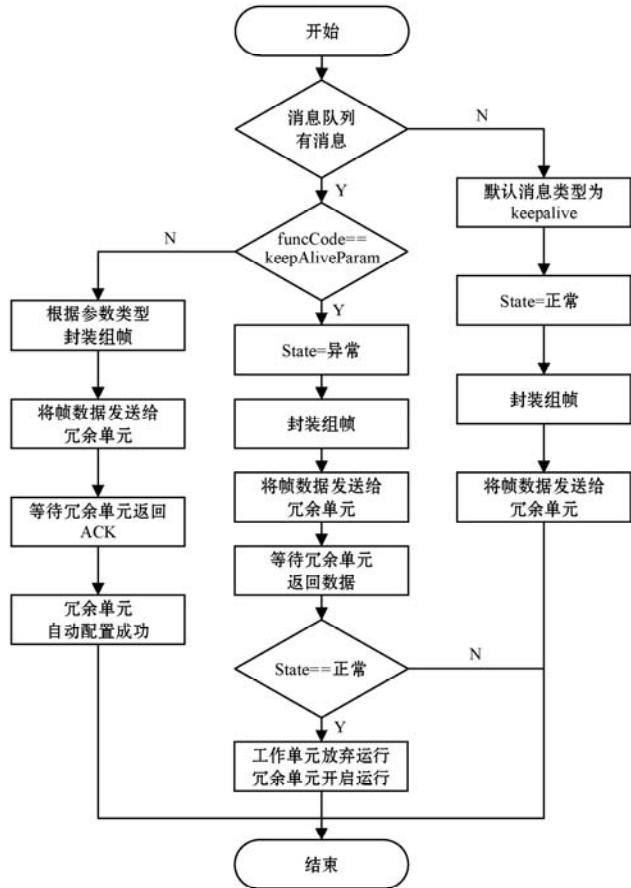


图 11 冗余通信进程流程

5 结语

将本文基于 Modbus 协议设计并实现的冗余总线控制器组装到现有的工厂测试系统中进行测试,测试结果和工业现场应用均能够做到长时间稳定可靠地运行,满足工作要求。

该控制器有效降低了 PLC/DCS 上位机的 CPU 资源消耗,使 PLC 能充分地发挥其控制优势,提高系统的实时性。同时总线控制器内部采用工作和冗余双 CPU 设计,每个 CPU 分别拥有两套 485 总线接口与终端执行机构成总线环路,对 485 总线接口设计了 EMC 保护电路,使得总线控制器具有一定的抗干扰、防浪涌的能力,以应对复杂的工业环境,有效提高了系统的鲁棒性。本文为解决传统过程控制系统的鲁棒性和实时性问题提供了一种解决方案。

参 考 文 献

[1] 宋振鹏,孟祥印,徐鑫凯,等. 基于 Modbus 与 ZigBee 的通信冗余过程控制系统开发[J]. 电子技术应用,2016,42

(1):61-64.

- [2] 许兰波. 基于 Modbus RTU 的 PLC 与 LabVIEW 通信液位监控系统设计[J]. 信息技术与信息化,2021(2):111-113.
- [3] 马峰. 基于 PC 的 PLC 控制系统应用[J]. 山西冶金,2021,44(2):172-173.
- [4] 王伟刚,朴顺善. 基于 MODBUS 的电动执行器远程控制系统实现[J]. 自动化与仪器仪表,2009(6):96-97.
- [5] 黄剑,张红梅,张向利,等. 基于 Modbus 协议的数据采集与并行加密通信系统[J]. 计算机应用与软件,2017,34(3):260-266.
- [6] 程雨茂,朱祖清,刘淑坤. 基于 Modbus 总线协议的冗余可靠性研究[J]. 计算机应用与软件,2007,34(3):106-113.
- [7] 徐加征. USB3.0 接口电路的 EMC 和安全设计[J]. 安全与电磁兼容,2020(1):62-64.
- [8] 高峰,崔琪,杨常府. 分层模块化软件设计在变电站通信管理机中的应用[J]. 电力自动化设备,2004(6):95-97.
- [9] 熊翔翔,全召. RT-Thread Smart 微内核操作系统概述[J]. 单片机与嵌入式系统应用,2021,21(3):9-12.
- [10] 邱伟. RT-Thread: 两大安全措施保障嵌入式软件开发的安全[J]. 电子产品世界,2021,28(4):9.
- [11] 朱康军. 产品模块化设计关键技术的研究[D]. 长沙:湖南师范大学,2020.

(上接第 198 页)

- [6] 苏慧慧,彭艺,曲文博. 基于疯狂自适应鱼群算法的认知无线电频谱分配[J]. 应用科学学报,2020,38(6):882-889.
- [7] 徐航,张达敏,王依柔,等. 基于改进二进制灰狼算法的频谱分配[J]. 计算机工程与设计,2021,42(5):1353-1359.
- [8] 孙汉卿,刘征,王桂芝,等. 基于多态蚁群优化算法的认知无线电频谱分配[J]. 计算机应用与软件,2020,37(12):260-265,321.
- [9] Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016, 96: 120-133.
- [10] 何庆,徐钦帅,魏康园. 基于改进正弦余弦算法的无线传感器节点部署优化[J]. 计算机应用,2019,39(7):2035-2043.
- [11] Abualigah L, Dulaimi A J. A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm[J]. Cluster Computing, 2021, 24: 2161-2176.
- [12] 郑健. 离散正弦余弦算法求解大规模 0-1 背包问题[J]. 山东大学学报(理学版),2020,55(11):87-95.
- [13] Zhang Q, Wan P, Wang Y, et al. A spectrum allocation method based on random drift swarm optimization algorithm [C]//IEEE International Conference on Communication Software & Networks, 2017.
- [14] 孙海建. 基于粒子群算法和遗传算法的频谱分配研究[D]. 长春:吉林大学,2015.
- [15] 赵知劲,彭振,郑仕链,等. 基于量子遗传算法的认知无线电频谱分配[J]. 物理学报,2009,58(2):1358-1363.

基于低秩因子最优估计的特征级数据融合

杨俊玲¹ 许琪²

¹(郑州澍青医学高等专科学校 河南 郑州 450000)

²(西安交通大学计算机科学与技术学院 陕西 西安 710049)

摘要 为了提升信息融合系统对噪声的鲁棒性以及计算效率,并且考虑特征之间的关联性,提出一种基于低秩因子最优估计的特征级数据融合方法。通过优化潜在特征向量与从线性组合估计值中提取的余弦相似性度量,找到潜在信号特征空间的估计值。然后利用随机矩阵理论进行特征及数据融合,用于求解具有不同噪声水平的约束数据驱动优化问题。在两个数据集上的实验结果表明,该方法效果显著。

关键词 低秩因子 最优估计 相似性度量 特征级融合

中图分类号 TP391.41

文献标志码 A

DOI:10.3969/j.issn.1000-386x.2024.11.040

FEATURE LEVEL DATA FUSION BASED ON OPTIMAL ESTIMATION OF LOW RANK FACTORS

Yang Junling¹ Xu Qi²

¹(Zhengzhou Shuqing Medical College, Zhengzhou 450000, Henan, China)

²(School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, Shaanxi, China)

Abstract In order to improve the robustness to noise and computational efficiency, and considering the correlation between features, a feature level data fusion method based on optimal estimation of low rank factor is proposed. By optimizing the cosine similarity measure between the potential feature vector and the vector extracted from the linear combination estimation, the estimation of the potential signal feature space was obtained. The stochastic matrix theory was used for feature and data fusion to solve constrained data-driven optimization problems with different noise levels. Experimental results on two datasets show that the effect of the proposed method is remarkable.

Keywords Low rank factor Optimal estimation Similarity measure Feature level fusion

0 引言

信息融合指的是使用不同数据提取特征集并进行融合,生成系统的单一表示,在许多不同的领域都有广泛的应用,包括人类活动检测和健康监测、遥感和分析和图像分割等^[1-2]。随着数据时代的发展,如何实现高效的信息融合一直是学者们研究的重点。

现有的信息融合方法主要可以分为三类:决策级融合、匹配分数级融合和特征级融合。其中,决策级融合对多个数据或单一源数据的不同表示形式进行的接受/拒绝决策的集成^[3]。文献[4]针对提出了一种基

于协同过滤和后验概率加权的决策级融合识别方法,有效地提升了识别精度。文献[5]引入神经网络决策机制,有效提升了决策过程的非线性描述特性,有效提升了融合的效果。虽然决策级融合有着广泛的应用,但是由于信息量有限,难以处理复杂问题。匹配分数级融合是在原始数据矩阵的层次上进行融合,目的是构建一个更高质量的数据源,捕获每个模式中存在的信息。文献[6]提出一种基于迁移学习的带拒绝识别阈值的手指静脉与手指关节纹共同决策同一主体的双模态分数级融合识别方法。文献[7]提出了一种基于分数级融合技术的多模式生物特征识别系统,用于构建鲁棒的人体识别系统。虽然该方法在生物特征识别

上具有较好的效果,但由于过分依赖主观评分和先验知识,其应用局限性较大。

特征级融合使用不同的特征提取算法在同一个或不同的数据集上分别生成的特征集,这些特征集融合在一起产生系统的单一表示。文献[8]提出了一种基于混沌遗传算法的多核学习的特征级融合算法。文献[9]针对多波段同步融合图像清晰度不高的问题,提出一种基于注意力机制生成对抗网络的图像特征级融合方法。文献[10]将熵作为评价指标引入特征级融合中,有效提升了对轴承故障不确定性的预测精度。然而上述方法忽略了各个特征级之间的关联性,并且不同特征集的组合提高了变量的维数,导致计算复杂度增大,而且对噪声的鲁棒性较差。

为解决上述问题,本文提出了一种基于低秩因子最优估计的特征级数据融合方法。通过优化潜在特征向量与从线性组合估计值中提取的余弦相似性度量,找到潜在信号特征空间的估计值,并且利用随机矩阵理论求解具有不同噪声水平的约束数据驱动优化问题。实验验证了提出方法的优越性。

1 本文模型

1.1 问题描述

假设存在噪声矩阵 $\{\tilde{Y}_i\}_{i=1}^q$ 的 q 个信号,每个矩阵在高阶噪声矩阵中嵌入一个低阶扰动,即:

$$\tilde{Y}_i - \gamma Y_i = \Theta_i B B^H \in \mathbf{R}^{n \times n} \quad (1)$$

式中: $B = [b_1, b_2, \dots, b_r] \in \mathbf{R}^{N \times r}$ 是具有正交列的潜在信号矩阵, b_j 是第 j 个潜在信号(特征)向量, $\Theta_i = \text{diag}(\theta_{i,1}, \theta_{i,2}, \dots, \theta_{i,r}) \in \mathbf{R}^{r \times r}$, $\theta_{i,j} \in \mathbf{R}$ 是第 i 个矩阵样本中第 j 个信号向量的信噪比, Y_i 表示噪声矩阵,与 $\{Y_j\}_{j \neq i}$ 无关, γ 是噪声级的参数,随着 γ 的增大,噪声越来越密集。

本文定义了观测的特征向量矩阵, \tilde{B}_j 和 $\tilde{\Theta}_j$, $j \in \{1, 2, \dots, r\}$,通过

$$\tilde{B}_j = [\tilde{b}_{1,j}, \tilde{b}_{2,j}, \dots, \tilde{b}_{q,j}] \quad (2)$$

形成了样本矩阵的第 j 个潜在信号向量 b_j 相对应的信息特征向量。

本文定义了 $\tilde{B}(u) = [\tilde{b}_1(u), \tilde{b}_2(u), \dots, \tilde{b}_r(u)]$,表示 $u = [u_1, u_2, \dots, u_q] \in \mathbf{R}^q$,其中:

$$\tilde{b}_j(u) = \frac{\sum_{i=1}^q u_i \tilde{b}_{i,j}}{\left\| \sum_{i=1}^q u_i \tilde{b}_{i,j} \right\|_2} \quad (3)$$

式中:每个 $\tilde{b}_{i,j}$ 本身就是潜在向量 b_j 的有偏估计, $\tilde{b}_{i,j}(u)$ 位于一个近似于第 j 个潜在信号向量 b_j 的子空间中。

对于每个 $j \in \{1, 2, \dots, r\}$,从每个样本的信息特征向量计算每个潜在信号向量 b_j 。更准确地说,本文希望从每个样本矩阵的极值信息特征向量的线性壳中找到 b_j 的最佳线性估计,即:

$$u_j^{\text{opt}} = \arg \max_{u \in \mathbf{R}^q} \langle b_j, \hat{b}_j(u) \rangle \quad (4)$$

然而,由于该目标函数假定潜在向量 b_j ,因此不能如上文所述评估该目标函数,这个约束意味着优化问题本质上是潜在的。本文致力于解决潜在优化问题的求解,并得到 u_j^{opt} 的闭式解。

1.2 理论分析

为了便于理论分析,本文对噪声矩阵提出了一些假设,如下所示。

假设 1 对于每个 $i \in \{1, 2, \dots, q\}$, Y_i 具有与特征值无关的渐近不变特征向量,令 μ_A 表示矩阵 A 的经验特征值分布。

$$\mu_A = \frac{1}{N} \sum_{j=1}^N \delta_{\lambda_j}(A) \quad (5)$$

式中: $\lambda_1(A) \leq \lambda_2(A) \leq \dots \leq \lambda_N(A)$ 是矩阵 A 的特征值。

假设 2 对于每个 $i \in \{1, 2, \dots, q\}$,经验特征值分布 μ_{Y_i} 收敛效果不好,几乎收敛到紧支撑非随机度量指标 μ_i 。为了评价式(4)的目标函数,本文定义了函数 $f(\cdot): \mathbf{R}^q \rightarrow \mathbf{R}$ 。

$$f(u) = \langle b_j, \hat{b}_j(u) \rangle \quad (6)$$

定理 1 有

$$u_j^{\text{opt}} \propto (\tilde{B}_j^H \tilde{B}_j)^{-1} \tilde{B}_j^T b_j \quad (7)$$

可以进一步计算 u_j^{opt} 的最佳函数值,即:

$$f(u_j^{\text{opt}}) = \sqrt{b_j^H \tilde{B}_j (\tilde{B}_j^H \tilde{B}_j)^{\dagger} \tilde{B}_j^H b_j} \quad (8)$$

证明:可以计算函数 $f(u)$ 的梯度:

$$\nabla_u f(u) = \frac{1}{\|\tilde{B}_j u\|} \tilde{B}_j^T b_j - \frac{b_j^T \tilde{B}_j u}{\|\tilde{B}_j u\|^3} \tilde{B}_j^T \tilde{B}_j u \quad (9)$$

函数 $f(\cdot)$ 的梯度的任何根满足:

$$u \propto (\tilde{B}_j^T \tilde{B}_j)^{-1} (\tilde{B}_j^T b_j) \quad (10)$$

$(\tilde{B}_j^T \tilde{B}_j)$ 有不可逆的可能性,在这种情况下,应该用伪逆 $(\tilde{B}_j^T \tilde{B}_j)^{-1}$ 替换逆 $(\tilde{B}_j^T \tilde{B}_j)^{-1}$ 。

由于函数在 u 中是奇对称的,如果 u^* 是极大值,那么 $-u^*$ 是函数的极小值。加上函数在 u 的正比例下保持不变,这说明对应于最大值的方向的表达式为 $u_j^{\text{opt}} = (\tilde{B}_j^T \tilde{B}_j)^{-1} (\tilde{B}_j^T b_j)$ 。

在函数 $f(u)$ 中嵌入这个优化器,可以得到: