

架构的力量——论控制系统的发展

文/宋华振
贝加莱工业自动化（中国）有限公司

在中文的语境里，谈论“巧”的时候经常会用到“四两拨千斤”，这是一个设计问题，精巧的结构能发挥这个作用，方向正确了，就无须消耗太多的力量，而人们在乎成本，有效的架构就可以降低成本。

传统PLC局限带来的问题

比较常见的问题是历史遗留的架构，传统PLC的确比较擅长逻辑控制，自20世纪70年代到现在很长一段时期里都存在着这种情况。像印刷里的套色、并条里的匀整及CNC与机器人系统，通常都是专用系统，包括最近看到的光伏设备里的温度控制，需要非常强的自整定、温区解耦要求，普通的PLC无法达到要求，故产生了许多专用型控制器。控制系统架构设计如图1所示。

实际上自20世纪90年代开始，贝加莱推出了基于定性分时多任务操作系统的可编程序计算机控制器（PCC），就已经支持高级语言编程，能够将复杂的回路调节、时间严苛型任务及运动控制（CNC）集成一体。因此，像印刷套色系统替代专

宋华振，贝加莱工业自动化（中国）有限公司技术传播经理，SAC/TC124/SC4，SAC/TC159/SC5委员，OPC基金会中国技术顾问，中国自动化学会集成自动化分会委员。



用套色、壁厚控制及多轴向经编机等许多应用，得益于这一架构，贝加莱用通用控制器代替了原来的专用控制器，使得系统整体成本得到降低，单一系统可以代替多个系统，如图2所示。

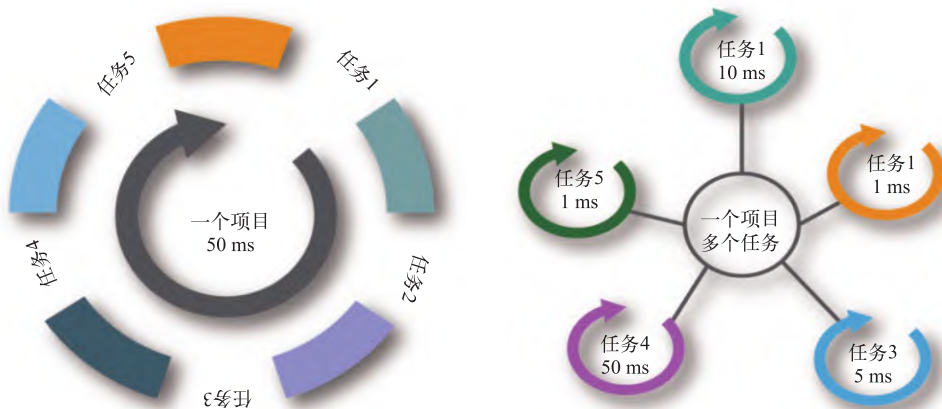


图1 控制系统架构设计

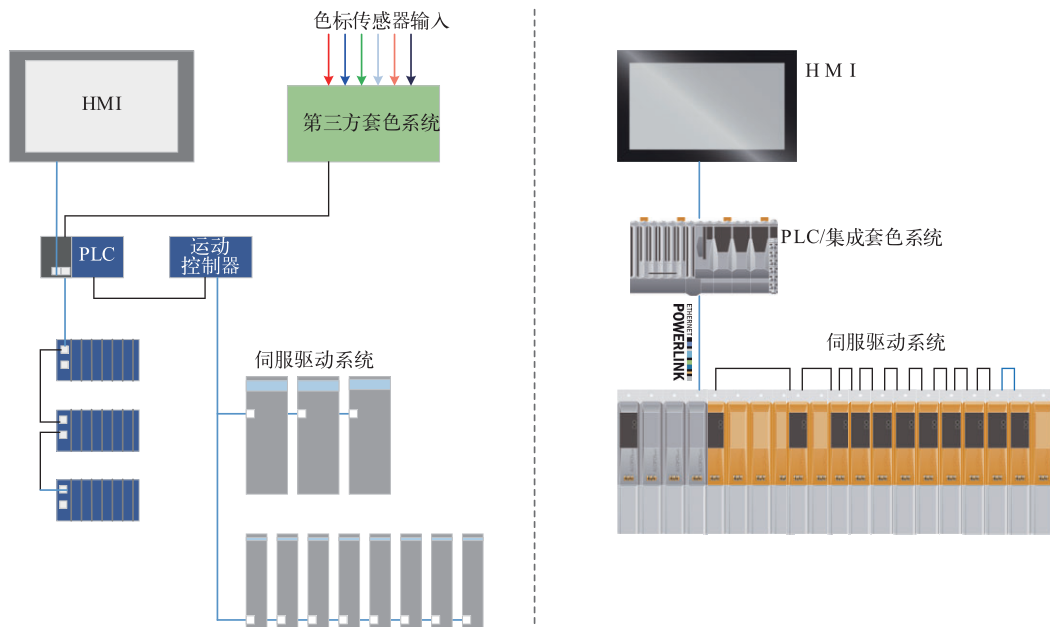


图2 单一系统代替多个系统

今天，包括Hypervisor的PC，集成Linux/RTOS，使得像边缘调度、策略性任务也可以与实时控制任务集成，包括机器学习类都可以被集成到统一平台。

当然，必须承认这里有一个工艺Know-How移植的问题，以前认为自主掌握Know-How，专用系统因为工艺Know-How而非硬件值钱，但封闭的系统无法真正赢得竞争力。对于价格来说，通用平台具有更高性价比。

在运动控制领域，CNC通常也是一个专用系统，因为它所需要处理的插补运算都需要反复的计算，动态处理。因此，会看到一些旧的架构采用一个CNC系统来处理，同时用另一个控制器去处理插补任务，但有时需要用到CNC功能的非典型CNC（如车铣刨磨）机器中的CNC仅仅只是简单的直线插补或圆弧插补，而由于普通的PLC+运动控制模块处理不了，就增加了一个CNC系统，简单的CNC架构一个系统，价格不高，但这两个系统还得协调。

机器人领域也存在同样情况，传统意义上，普通控制器处理不了机器人，因此，机器人有专用的控制系统，通常运行Windows+RTOS架构，随着工艺的集成度越来越高，会出现一些机器人针对打磨、焊接涂装等领域的工艺Know-How的封装，机器人系统因为集成了行业Know-How，而变得更为

专业。但在另外一些机器应用中，例如注塑机的取件、包装中的码垛及印刷装版等场景，机器人工作相对单一，为此专门配置机器人系统控制似乎不那么合算，完全可以利用通用控制器的处理能力，并能够在在一个软件架构下编程，实现机器人与机器在位置、速度及时间上的高精度同步，这就是贝加莱的Machine Centric Robotics的理念。对于机器制造商来说，可以直接在机器中集成一个机器人，这个应用中机器是中心，机器人相当于一个运动执行单元。集成架构实现实时与边缘智能集成，如图3所示。

基于开放架构的控制器，借用开放芯片世界中的处理能力、存储和网络通信能力，将Know-How以软件形式运行，这样的架构就更为满足集成性高、开放连接多的新时代机器配置。

降低成本带来的组合

另一种情况是厂商把系统拆分为一个个产品、组件来理解，以纯采购的模式分析系统，寻找最便宜的解决方案，例如，笔者见过一个机器上用了A家的PLC、B家的HMI（用于机器操作）、C家的运动控制器、D家的驱动器、E家的电动机、F家的工业PC（用于Windows画面）、H家的I/O端子、I家的变频器，里面有J家的电源、K家的低压电器及一个新开发的电路板，这是笔者见过最壮观的自动化产品展示柜。

这些东西如果出现了问题，那么谁来解决问题呢？并且工程师写程序、调试参数怎么办？硬

件看上去便宜了，但是软件方面却带来了很大的问题。

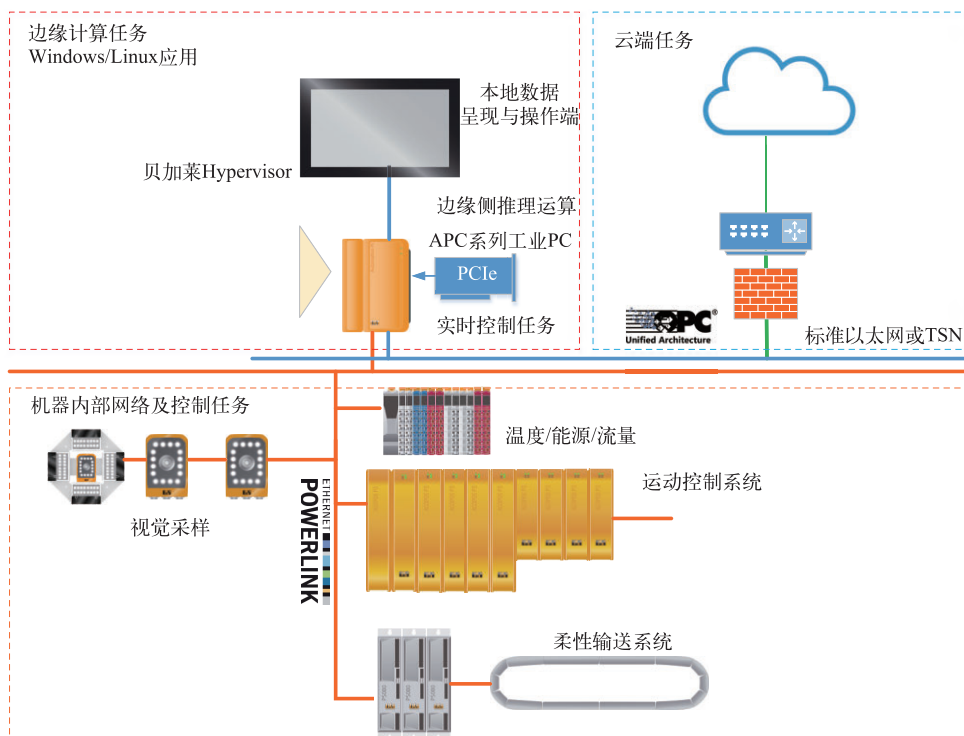


图3 集成架构实现实时与边缘智能集成

集成系统有好处，如果一个控制器即可完成逻辑、运动控制及高级算法，并没有必要用集成系统，人们总是忽略了工程师的成本及耗费在软件上的成本。在今天这种局面下，人工成本不断上升，软件越来越复杂，这种架构可能也比较少见。

历史遗留问题——规划的问题

还有一种情况，也采用多个系统，但并非是从选择多个便组件的视角。经常会出现多个PLC或多个PC堆叠的情况，也有几个HMI的，并非是模块化，通过与很多用户交流，发现这是历史遗留问题。这个机器系统的格局是在不同阶段做的，后来又增加了一个需求，原来的控制器做不了，加一个新的系统，跟前面不兼容，后来又需要增加一个子系统，为此又增加了一个专门处理这个复杂任务的PC或PLC。这些系统之间或是没有关系，或是增加了复杂的接口或专门的转化模块，并非是最优系统设计，且成本较高，退出成本也较高，想更换系统时，迁移成本会很高。

为什么用户设计系统会出现很多架构上其实并

不经济的问题？是产业发展太快造成的，设计时并未有长远的规划，导致了系统的架屋叠床的局面，单一的组件上看上去成本低，但当新的子系统进入时，由于无法兼容，并且原有程序无法轻易移植，因此必须保留，多次这样的设计，导致最终成本很高，但是又出于无奈，重新来过又需要大量的开发与测试，结果导致积重难返。

系统规划是一个“磨刀不误砍柴工”的事情，必须预先考虑到未来的潜在需求，规划人员必须站在5~10年的视野来看待系统架构。不仅仅是硬件架构，更多是未来的机器软件架构，能够长期可扩展，而在新增需求时，如何避免大动干戈的调整，因为推倒重来的代价实际上是非常高的。

换个视角，又发现工程师在整个项目的决策中受到了很多不利因素的影响，做出了妥协，或是因为时间与进度关系，或是因为市场销售方面的决策因素，再或是因为某种特殊政策（单一品牌战略），导致了无法选择更为全局长远的架构规划。《用工程师思维来工作》这本书中提到工程师思维很重要的一个点——让工程师来决策，但很难做到。

系统架构设计

举个例子，为何在进行电子凸轮裁切这类任务的时候切不好？因为架构已经决定了。而贝加莱的驱动系统可以直接把外接编码器接入到驱动系统（多个编码器槽直接在驱动器上），这个架构带来

的好处在于，可以直接在驱动器中计算同步外接编码器的位置或速度，这使得它可以获得更高的响应速度，而采用PLC的I/O反馈外接编码器来跟随的模式，就达不到高速下的高精度裁切。本地计算的智能驱动系统如图4所示。

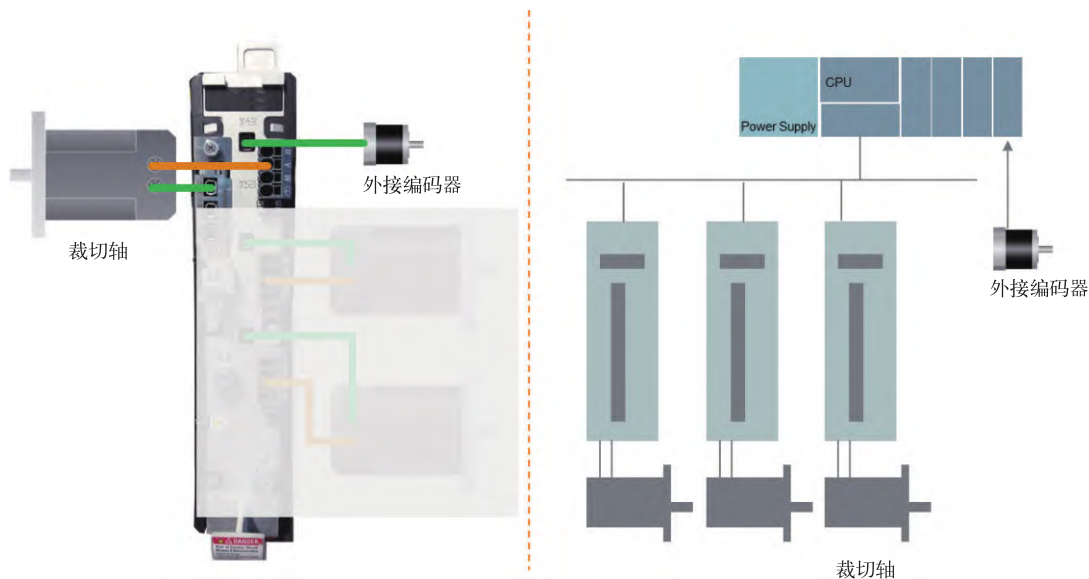


图4 本地计算的智能驱动系统

像Powerlink，有些人觉得循环周期100 μs好像不算快，但是，Powerlink的架构并非是一个主站集中控制模式下的运行，而是基于分布式架构的广播帧方式支持交叉通信，因此，很多同步任务并不需要去主站，而是轴间可以通过广播帧实现直接跟随关系，因此，不必通信周期那么快，却可以实现高的控制周期。包括PRC省略握手过程的机制等优化方法，这都是架构的力量。

系统架构设计实际上在于“巧”，所谓的“四两拨千斤”就是这个意思，从产品思维出身和从方案思维出身的企业决定了对于“系统”的理解是不同的，而做系统的人会有全局优化的概念，做产品要更多从局部思考问题，很难做到全局最优的设计。做控制系统出身的厂商和电动机出身的厂商，看待运动控制的视角也是不同的，控制作为主站，需要全局协同，而做电动机出身的厂商做运动控制，其视角主要在电动机本身的性能与对象的匹配这个问题上。

底层架构的影响

开放架构非常适合于自主开发的应用，所谓开

放，其实主要是指采用“开放世界”的技术，即横向产业技术，客户愿意用已有的封闭专用系统，因为这样快，虽然那时就和客户谈自主研发工艺Know-How的重要性，但受制于对封闭系统的路径依赖，并未掌握核心工艺，这种依赖会随着系统越来越强而越来越依赖，最终导致今天缺乏工业软件，虽然很多人把工业软件聚焦到了CAD/CAE，其实嵌入式应用的Know-How一样是普遍存在于工业系统的。

还有一个问题在于对工业软件的理解，商用软件或在设计端的工业软件与在运行端嵌入式的软件、DCS这样的软件的发展模式实际是完全相反的，因此，用软件的发展思维来发展工业软件一定会出问题的。

IT系统的软件是一种自上而下的架构，先有骨架再去丰满，而OT的软件研发是先有应用问题需要解决，然后再标准化的过程，工业软件的模式则必须经历不断的工程迭代过程，这时候，必须考虑非常强的底层架构设计，让这样的系统具有扩展能力，而又不会影响原来的架构。EA