

基于软件中间件技术的 MATLAB - LabMap 分布式控制架构研究

Cecil Bruce-Boye
University of Applied Sciences
3 Stephensonstrasse
Lübeck, 23562 Germany

Rüdiger zum Beck
University of Applied Sciences
3 Stephensonstrasse
Lübeck, 23562 Germany

Dmitry A. Kazakov
cbb software GmbH
1 Charlottenstrasse
Lübeck, 23560 Germany

摘要: 本文描述了一个状态观察机控制器分布式控制系统。分布式控制基于软件中间件技术。LabMap/LabNet 实现了软硬件的抽象。仿真控制平台采用 MATLAB/Simulink。本文描述的方法给如下情形一种指导思路: 计算机资源有限并分散在几个硬件系统, 尤其在嵌入式应用领域。

1. 前言

实际工程中, 很多控制系统不是集中在某个特定的地方。输入和输出操作分别在分布式系统的不同节点进行。同样地, 控制行为不仅能跟本地的输入输出绑定, 也能跟远端的输入输出绑定, 甚至实现分布式控制。软件中间件技术能应用到整个系统的各个部分, 提供了分布式数据访问、系统软硬件集成等。LabMap[2]就是自动控制领域的一种软件中间件。LabMap 将应用层从硬件层抽象出来, 将硬件层从应用层剥离出去。LabMap 的另一个重要优点是提供大量软硬件构件的无缝集成, 及构件化的软件设计[1,3,4]。

MATLAB/Simulink [5]是世界上应用最广泛的设计、仿真、测试和控制系统的最终产品工具之一。虽然 MATLAB/Simulink 本身也支持对硬件的直接操作, 但这需要相对比较昂贵的插入卡, 其支持的硬件也非常有限。

LabMap 对 MATLAB/Simulink 的扩展接口大大扩展了其在主要工程领域的应用。这个接口使 MATLAB/Simulink 在仿真模式下实时控制硬件。这些操作不需要实时的环境。MATLAB/Simulink 的仿真时间被映射成接口的实时操作时间。

LabMap 软总线支持网络功能, 这点为控制系统的研究和创建提供了更广泛的空间(即使在没有嵌入式目标系统的情况下)。但同时也出现一些问题, 如不能再依赖系统的硬件实时时间等。本文提示了软实时可成为一个替代的选择。

2. 相关的工作

在自动化控制领域, 有大量不同体系结构的软件中间件产品。软件中间件的体系结构需满足多种需求, 有时甚至是一些专有的需求。

OPC[8]是一种面向开放数据连接的技术。与 LabMap 类似, OPC 以变量的形式处理数据。OPC 基于客户机/服务器模式。许多硬件生产厂商提供其硬件产品的 OPC 服务器。MATLAB OPC 工具箱 2[9]给 Simulink 提供了一个

OPC 客户机, 但没有提供发布内部信号所需的服务器部件。OPC 的弱点是相对脱离实际及客户机/服务器模式, 不适合开发分布式控制的应用。当前, OPC 通过引入 OPC DX[12] 改变了严格的客户机/服务器模式, 支持变量的服务器-服务器分布模式, 但客户机-客户机的模式仍然不行。

MATLAB 分布式计算工具箱 2[10]提供了一个可选的途径, 但其不支持 Simulink。

CORBA[11]是一种软总线技术, 没有客户机/服务器模式问题, 但缺乏时间限制方面的描述。另外, CORBA 基于方法激活机制, 并不能很好地适用到 MATLAB/Simulink 框架。

3. 软件中间件体系结构

在自动化控制领域, 有大量不同体系结构的软件中间件产品。软件中间件的体系结构需满足多种需求, 有时甚至是一些专有的需求。OPC[8]是一种面向开放数据连接的技术。与 LabMap 类似, OPC 以变量的形式处理数据。虽然 MATLAB OPC 工具箱 2[9]给 Simulink 提供了一个 OPC 客户机, 但没有提供发布内部信号所需的服务器部件。一般情况下, 客户机/服务器模式不适合开发分布式控制的应用。MATLAB 分布式计算工具箱 2[10]提供了一个可选的途径, 但其不支持 Simulink。CORBA[11]是一种软总线技术, 没有客户机/服务器模式问题, 但缺乏时间限制方面的描述。另外, CORBA 基于方法激活机制, 并不能很好地适用到 MATLAB/Simulink 框架。

从应用软件角度, 软件中间件 LabMap 呈现的是一系列的变量。每一个变量都有类型、当前值、和时间戳等属性。不需要其他配置。系统运行后, 应用程序可以立刻访问变量。每个寄存器变量支持四种基本的操作:

- **获取。**安全地读取每个寄存器变量的值, 读取时保证值和时间戳的一致性。这样就避免了应用中多任务同时访问同一个变量时的变量锁定问题。应用程序不清楚变量值的来源, 采用策略获取变量的值。
- **设置。**安全地设置寄存器变量的值。
- **请求。**从底层硬件请求一个寄存器变量的最新值。这个操作由底层的硬件驱动完成。这是个异步操作, 上层应用不会被阻止直到该 I/O 操作完成。

- 发送。一个寄存器变量的当前值被发送到底层硬件。跟请求操作类似，上层应用并不清楚这个操作，而是由底层的硬件驱动完成。

软件中间件异步地完成大部分过程中激活的操作。硬件的 I/O 操作是最重要的一个例子。为了同步这些操作，软件中间件提供了多种同步机制：

- 等待 I/O 操作完成。一个应用程序可能进入有限时间的等待状态，直至包含变量的 I/O 操作完成。
- 等待值发生变化。
- 黑板。应用程序有时候需要跟踪某个变量值的整个变化过程，比如通过跟踪信号观察信号波形。通常的方法是通过变量值的来源和应用程序之间的某种通告机制获取变量值的变化。如果不占用系统资源，这种点对点的机制是很难实现的，当应用程序异常结束时也存在资源泄漏问题。软总线采用了另一种方法：将变量值的所有变化记录在黑板上并保留一段时间，所有应用程序都可以通过检查黑板的内容跟踪变量值的变化。

4. 网络接口

软总线的抽象机制隐藏了应用及其部件的分布特性。另外，软总线将网络功能抽象成一个硬件接口，即高度优化的网络接口“LabNet”。LabNet 对应 OSI 模型的表现层和应用层，但其大部分应用基于以太网 TCP/IP 套接字（Socket）。

LabNet 接口提供了很好的分布式控制支持，其主要特点如下：

- LabNet 对应用程序完全透明。
- 分布的不同子系统之间的同步对应用不可见。
- 软件中间件的请求，比如 I/O 操作通过 LabNet 传送到远端子系统并执行，被控制的硬件就像直接本地主机互联。
- 测试单元被一致地执行，而不是通过不同网络主机的兼容单元分开执行。
- 应用程序对变量的写操作是安全保护的，不会出现两个应用程序同时写一个变量。另外，软总线支持本地变量和全局变量。本地变量只对同一网段主机上的应用程序可见。全局变量可以从任何主机访问。针对全局变量的操作也是全局的。
- 每一个变量的值都带有时间戳。硬件驱动负责提供一个准确的时间戳。应用程序可以通过检查变量的时间戳检查其一致性。
- 采用端到端的同步机制保证时间戳的一致性。时间戳被透明地从一端的时钟传送到另一端。这个功能可用于本地时钟同步，同时也可采用任何外部时钟同步，如 NTP。
- 可从远端主机上查看本地变量。

5. 分布式状态观察机控制器系统

图 1 显示了本文采用的控制测试系统，由两台运行 MATLAB/Simulink 的主机系统组成。两台主机通过以太网互联。其中一台直接连接 WAGO I/O 系统[7]，获取数字编码器（HEDL 5540 AI2）的旋转速率，并设置电动马达（DC Maxon A-max 32）的放大器（4-Q-DC LSC 30/2）的电压。两台主机都运行 LabMap，并通过软总线交换数据。LabMap 为每个变量提供时间戳。LabNet 的网络接口在图上标识为“Net”。端到端的时间同步机制保证时间戳的一致性，时间戳被透明地从一端传送到另一端，因此不需要任何子系统时钟的同步。WAGO I/O 系统通过 LabMap 的 Modbus 接口与软总线相连。

6. 状态观察机和控制器

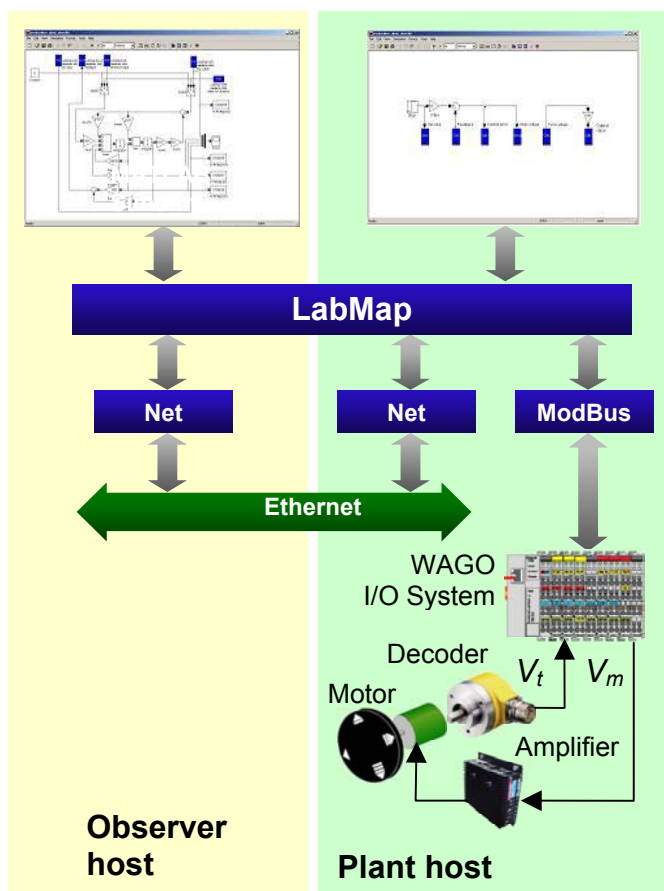


图 1 测试系统

在图 1 显示的测试系统中，采用 Luenberger 状态观察机控制器[6]。目标（Plant）机和观察机分别运行在两台不同网络的主机上。每台主机都是常用的 PC。图 2 显示了控制器模型概要。该模型中采用 Ackermann 公式计算状态观察机 gc 和状态控制器 rc。为了保证输入-输出准确，过滤器 vc 跟状态控制器 rc 关联计算。下面显示了 MATLAB m 文件：

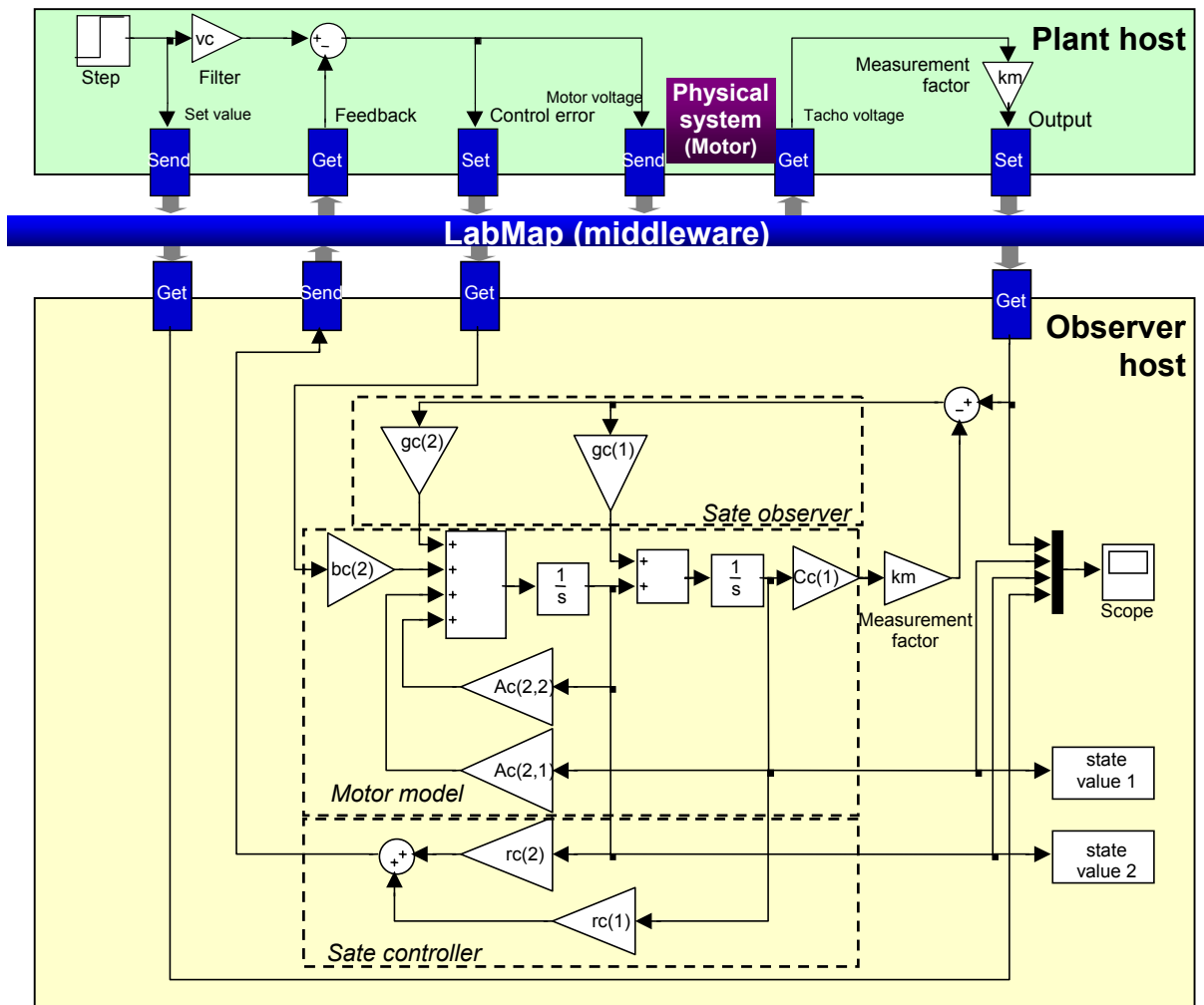


图2 MATLAB/Simulink模型，不同主机上运行的模型的部分，不同背景颜色区分不同主机

本实验系统显示了分布式控制器对外部干扰的鲁棒性

```

ObserverCalc.m
%calculation of state space controller and
%luenberger observer
%the system identificatin was carried out
%by the system identification
%software toolbox IDICON
num = [3.2637] %result from system identification
den = [1 10.862 11.205] %frequency domain
Ac=[0 1; -den(3) -den(2)]; %state spaces discription
bc=[0;num];
Cc=[1 0];

km=den(3)/num %measurment factor
cm=km*Cc;
p=[-10 -20]; %pole placement observer
gc=(acker(Ac',Cc',p))'; %state observer
gc=[19.138; -19.082]
prc = [-2 -2.5]; %pole placement controller
rc=acker(Ac,bc,prc); %state controller
rc=[-1.9013 -1.9493]
vc=inv(cm*inv(bc*rc-Ac)*bc);%filter vc=0.44622

```

图3显示了控制转速，包括输出值和两个状态变量。从图上可以看出，不管控制器在网络上如何分布及数据在网络上传输引起的滞后时延，控制性能都良好。

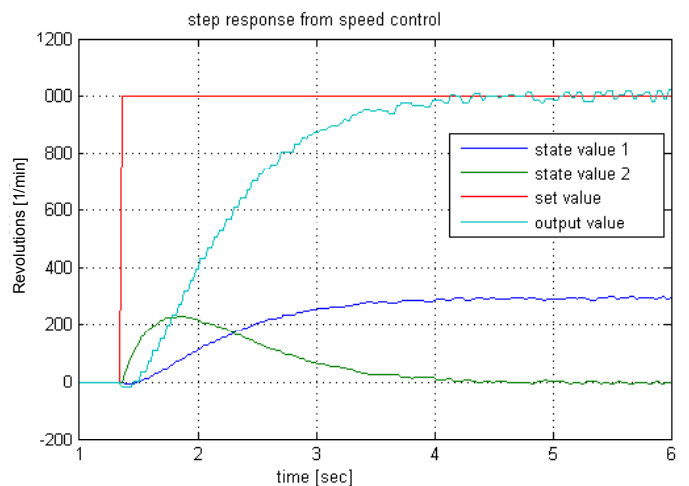


图3 状态控制器

在前面讨论的状态机控制器的基础上，图 4 显示了其开循环控制和关循环控制。在开始的 5.5 秒，直流马达运行在开循环模式，没有任何反馈信号。输出大概是每分钟 400 转而不是参考的每分钟 1000 转。但将状态机控制器切

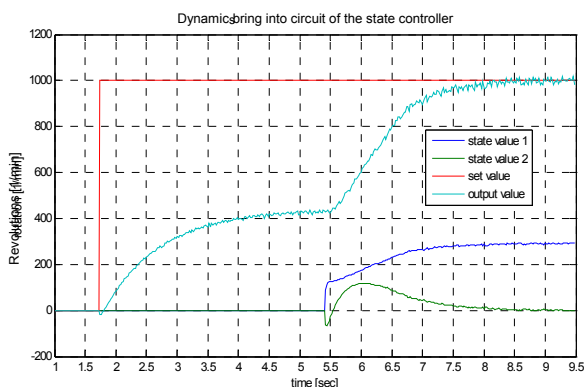


图 4 开循环和关循环控制性能

换到关循环控制后，其性能是令人满意的，输出值和参考值匹配。

7. 结论

本文描述的方法显示了在计算资源缺乏并分布在多个不同的硬件部件时分布式控制的可行性。在这种情况下，保证资源均衡，满足功能性和非功能性的控制系统需求是非常关键的，尤其在嵌入式应用领域。

另外，当使用 MATLAB/Simulink 不直接支持或软件无法仿真的硬件时，这个方法给工程师提供了一个 MATLAB/Simulink 下的应用框架。

LabMap 提供了将应用开发从硬件层剥离的接口。分布式应用通过变量与硬件通信，软件中间件通过指定硬件相关的协议和接口完成实际的数据发送。

下一步的工作将调查该方法实现分布式控制元件迁移的适应性。

当前的框架已经成功应用在吕贝克和不来梅应用科学大学的控制实验室的教学工作上[13]，实现了多组学生之间的实验室资源共享。学生在自己的观察机主机上设计控制器，然后通过图 1 所示的软件中间件访问实验室的目标 (Plant) 机，从而完成实验。

参考文献

[1] "An approach to distributed remote control based on middleware technology, MATLAB/Simulink - LabMap/LabNet framework", Conference CISSE (International Joint Conferences on Computer, Information and Systems, Sciences and Engineering) 2005, IEEE, 13-12-2005.
 [2] <http://www.cbb-software.com/labmap.html>

[3] C. Bruce-Boye, D. Kazakov, A. Fechner, "The hard and soft option LabMap the ultimate auto test platform", Testing Technology International, May 2001
 [4] C. Bruce-Boye, D. Kazakov, "Distributed data acquisition and control via software bus", International Industrial Ethernet Development High Level Forum 2004 (IEHF 2004) in Peking, Automation Panorama No. 5
 [5] M. Schetzen, V.K. Ingle, "Discrete systems laboratory in MATLAB," Thompson Engineering, 2000
 [6] D. Luenberger, "An introduction to observers." IEEE Trans. Automatic Control, AC-16 1971
 [7] <http://www.wago.com>
 [8] F. Iwanitz, J. Lange "OPC - Fundamentals, Implementation and Application", 2002
 [9] MATLAB OPC Toolbox 2 <http://www.mathworks.com>
 [10] MATLAB Distributed Computing Toolbox 2, <http://www.mathworks.com>
 [11] "The Common Object Request Broker: Architecture and Specification", OMG Document 99-10
 [12] "OPC DX 1.00 Specification 2003-03-11"
 [13] H.-W. Philippsen „Einstieg in die Regelungstechnik“, 2003, ISBN 3-446-22377-0