

通过实时网络实现多轴运动控制的同步

作者: Jens Sorensen, Dara O'Sullivan和Christian Aaen

摘要

实时确定性以太网协议(例如EtherCAT)已经能够支持多轴运动控制系统的同步运行。¹该同步包含两方面含义。首先,各个控制节点之间的命令和指令的传递必须与一个公共时钟同步;其次,控制算法和反馈函数的执行必须与同一个时钟同步。第一种同步很好理解,它是网络控制器的固有部分。然而,第二种同步到目前为止一直为人所忽视,如今成为运动控制性能的瓶颈。

本文介绍从网络控制器到电机终端和传感器全程保持电机驱动同步的新概念。所提出的技术能够大幅改善同步,从而显著提高控制性能。

问题陈述和现有技术

为了解释现有解决方案的局限性,考虑一个两轴网络运动控制系统,如图1所示。运动控制主机通过实时网络向两个伺服控制器发送命令和指令值,每个伺服控制器构成网络上的一个从机节点。伺服控制器本身由网络控制器、电机控制器、功率逆变器

和电机组成。实时网络协议采用不同的方法使从机节点与主机同步,一种常用方法是在每个节点处配置一个本地同步时钟。这种对时间的共识确保了所有伺服轴的指令值和命令均紧密同步。换言之,实时网络上的所有网络控制器都保持同步。

通常,在网络控制器和电机控制器之间有两条中断线。第一条通知电机控制器何时需要收集输入并将其放到网络上。第二条通知电机控制器何时从网络中读取数据。遵照这种方法,运动控制器和电机控制器之间以同步方式进行数据交换,并且可以实现非常高的定时精度。但是,仅将同步数据传送到电机控制器还不够;电机控制器还必须能以同步方式响应数据。如果没有这一能力,电机控制器就无法充分利用网络的定时精度。在响应指令值和命令时,电机控制器的I/O会出现问题。

电机控制器中的每个I/O(例如脉宽调制(PWM)定时器和ADC)都具有固有的延迟和时间量化。例如,我们来看图2所示的为功率逆变器产生栅极驱动信号的PWM定时器。该定时器通过比较指令值Mx与上下计数器来产生栅极信号。当控制算法改变Mx时,

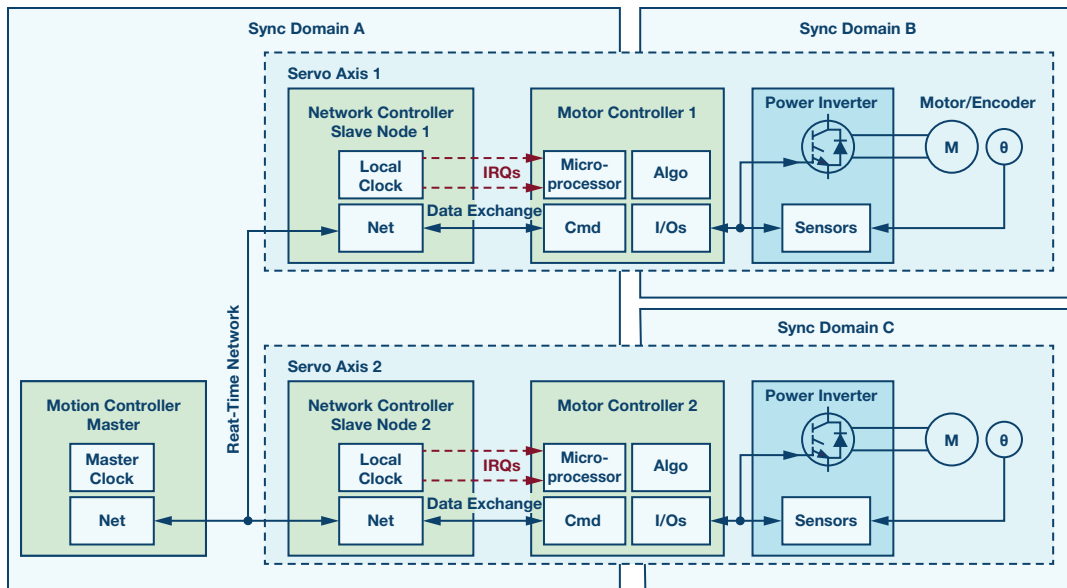


图1. 典型的两轴网络运动控制系统。

新的占空比要到下一个PWM周期才会生效。这相当于一个零阶保持效应，意味着每个PWM周期T内占空比仅更新一次或两次（若使用双更新模式）。

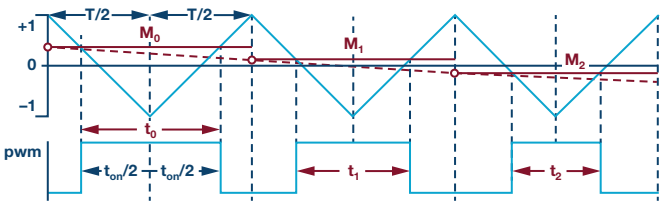


图2. PWM定时器的占空比更新。

在实时网络上，无论数据交换如何紧密地同步，PWM定时器的时间量化将最终成为轴同步的决定性因素。当接收到新的指令值时，无法在新的占空比生效之前对其进行响应。这导致时间不确定性最长可达一个PWM周期（通常在50 μs至100 μs范围内）。实际上，网络同步周期和PWM周期之间将存在一个未定义且可变的相位关系。将其与实时网络上低于1 μs的时间不确定性相比较，很显然，电机控制器的I/O在网络同步运动控制中起到更至关重要的作用。实际上，决定同步精度的不是实时网络，而是系统I/O。

再次参考图1，该系统具有A、B、C三个同步域，它们没有绑定在一起。它们实际上并不同步，具有最长可达一个PWM周期的可变不确定性。

同步不确定性及应用影响

在面向机器人和机床等应用的高性能多轴伺服系统中，可以清楚地看到时间不确定性的影响。在I/O级的电机控制轴之间的时间偏移量变化会对机器人或机床的最终三维定位精度产生直接且显著的影响。

考虑一个简单的运动曲线，如图3所示。在此示例中，电机速度指令值（蓝色曲线）上升后再下降。如果斜坡速率在机电系统的能力范围内，则实际速度预期值将遵循指令值。但是，如果在系统中任何位置存在延迟，则实际速度（红色曲线）将滞后于指令值，从而导致位置误差 $\Delta\theta$ 。

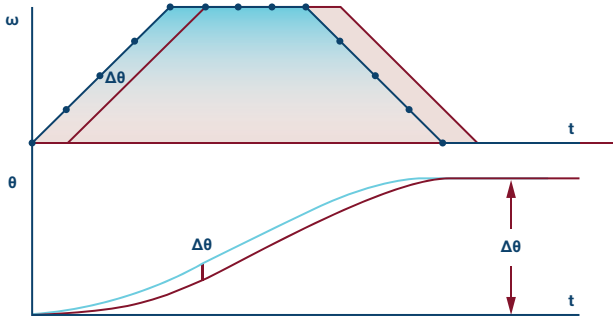


图3. 时序延迟对位置精度的影响。

在多轴机器中，根据机器的机械结构将目标位置(x, y, z)转换为角度轴向描述($\theta_1, \dots, \theta_n$)。角度轴向描述为每个轴定义了一系列相等时间间隔的位置/速度命令。轴之间的任何时序差异都会导致机器的精度降低。考虑图4所示的两轴示例。机器的目标路径以一组(x, y)坐标来描述。延迟使y轴命令产生时序误差，最终导致不规则的实际路径。

在某些情况下，通过适当的补偿可以最大程度地降低固定延迟的影响。然而，更关键的是无法对可变且未知的延迟进行补偿。此外，可变延迟会导致控制环路增益发生改变，从而使调整环路以获得最佳性能变得很困难。

应该注意的是，系统中任何地方的延迟都会导致机器精度不准确。因此，尽可能减小或消除延迟才能提高生产率和最终产品质量。

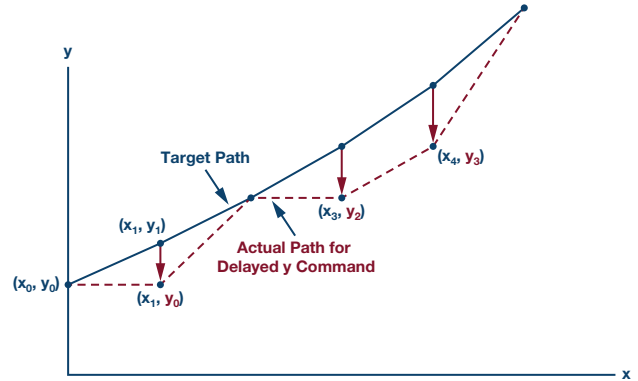


图4. 时序延迟对位置精度的影响。

同步和新型控制拓扑

传统的运动控制方法如图5的上半部分所示。运动控制器（通常为PLC）通过实时网络将位置指令(θ^*)发送到电机控制器。电机控制器由三个级联的反馈环路组成，包括控制转矩/电流(T/I)的内部环路、控制速度(ω)的中间环路和另一个控制位置(θ)的环路。转矩环路的带宽最高，位置环路的带宽最低。来自工厂的反馈保持在电机控制器本地，并与控制算法和脉宽调制器紧密同步。

采用这种系统拓扑，运动控制器和电机控制器之间通过位置指令值来实现轴的同步，但是在CNC加工等极高精度应用中，与电机控制器的I/O（反馈和PWM）同步的相关性只会使之成为问题。位置环路通常具有相当低的带宽，因此对I/O同步较为不敏感。这意味着即使网络与I/O位于不同的同步域中，指令级的节点同步性能通常也能接受。

虽然图5上半部分所示的控制拓扑很常见，但也可以使用其他的控制分区方法，例如在运动控制器侧实现位置和/或速度环路，并通过网络传送速度/转矩指令值。工业领域近期趋于转向一种新的分区方法，即所有的控制环路都由电机控制器转移至网络主机侧功能强大的运动控制器（见图5的下半部分所示）。在实时网络上交换的数据是电机控制器的电压指令(v^*)和运动控制器的工厂反馈(i, ω, θ)。这种控制拓扑由功能强大的多核PLC和实时网络实现，具有诸多优势。首先，该架构具有很高的可扩展性。还可以轻松地添加/移除轴，无需担心电机控制器的处理能力。其次，由于轨迹规划和运动控制都在同一个中心位置完成，因此可以提高精度。

新的控制拓扑也有缺点。在电机控制器中去除了控制算法，因此损失了代码执行和I/O的紧密同步。控制环路的带宽越高，损失I/O同步的问题就越大。转矩/电流环路对同步特别敏感。

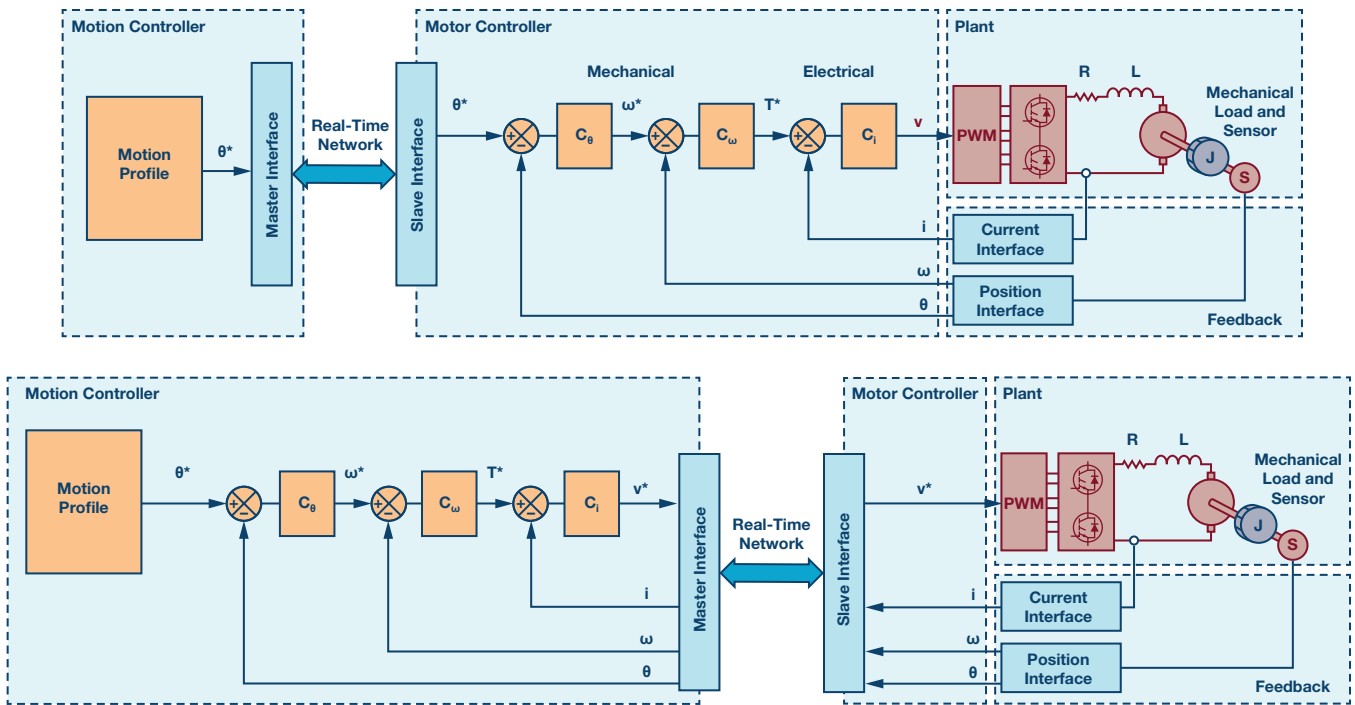


图5. 传统(上图)和新兴(下图)的运动控制拓扑。

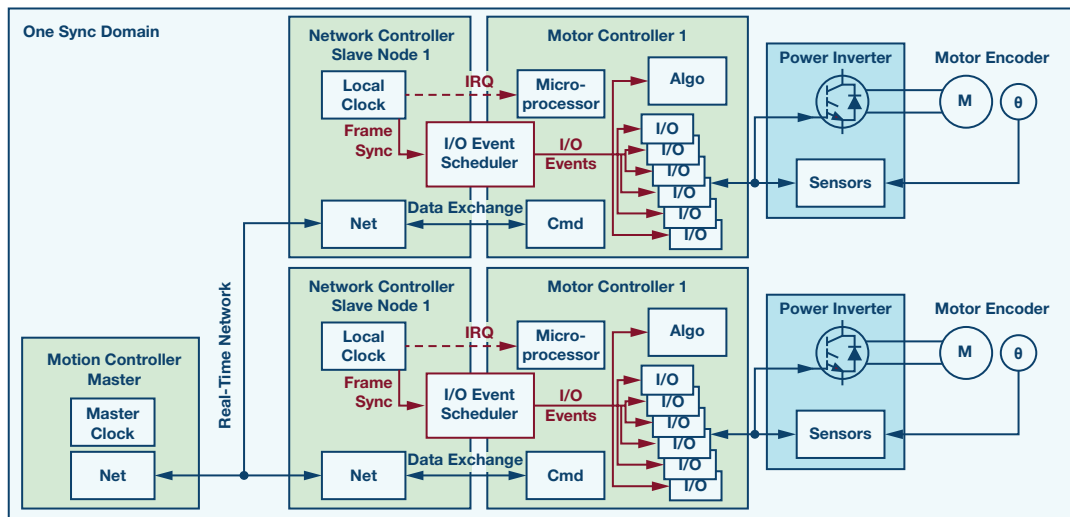


图6. I/O调度器将同步域绑定在一起。

建议的解决方案

将更快速的控制环路移至运动控制器提出了从网络主机直到电机终端全程同步的要求。

总体思路是使所有轴的I/O与网络同步,以使它们全都在一个同步域中运行。图6显示了一个位于网络控制器和电机控制器之间的I/O事件调度器。I/O事件调度器的主要功能是为所有外设生成同步/复位脉冲,使它们与网络流量保持同步。I/O事件调度器获取帧同步信号,该信号来源于网络控制器的本地时钟,并为必须与网络保持同步的所有I/O输出适当的硬件触发信号。

每个I/O都有自己的一组时序/复位要求,这意味着I/O事件调度器必须为每个I/O提供定制的触发信号。虽然触发信号的要求不同,但它们仍拥有一些通用法则。首先,所有触发信号必须以帧同步为基准。其次,存在与每个触发信号相关联的延迟/偏移。该延迟与I/O的固有延迟有关,例如,ADC的转换时间或sinc滤波器的群延迟。第三,存在I/O响应时间,例如,来自ADC的数据传输。I/O事件调度器掌握每个I/O的时序要求,并根据本地时钟连续调整触发/复位脉冲。生成I/O事件调度器每个输出脉冲的原理概述如图7所示。

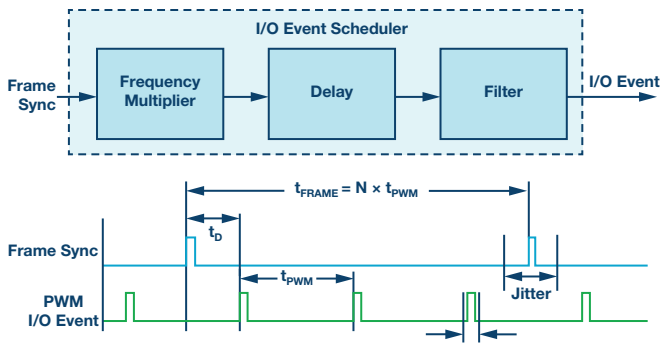


图7. I/O调度器生成触发脉冲。

在大多数网络运动控制系统中，帧速率以及帧同步速率等于或低于电机控制器的PWM更新速率。这意味着I/O事件调度器必须每帧周期提供至少一个、也可能是多个触发脉冲。例如，如果帧速率为10 kHz且PWM速率为10 kHz，则I/O事件调度器必须为每一个网络帧提供1个触发脉冲，类似地，如果帧速率为1 kHz且PWM速率为10 kHz，I/O事件调度器必须为每一个网络帧提供10个触发脉冲。这相当于图7中的倍频器。对每个同步脉冲施加延迟时间 t_0 ，以补偿每个I/O的固有延迟。I/O事件调度器的最后一个要素是智能滤波功能。每个网络上都会存在一些流量抖动。滤波器可减少触发脉冲的抖动，并确保帧同步频率的变化率受到限制。

图7的下半部分显示了PWM同步的一个示例时序图。请注意，本例中帧同步频率是PWM频率的倍数以及I/O触发信号抖动是如何减小的。

实现方案

图8显示了一个在网络运动控制系统中实施并进行测试的推荐的同步方案示例。网络主机采用Beckhoff CX2020 PLC，并连接到PC用于开发和部署PLC程序。实时网络协议（红色箭头）为EtherCAT。

电机控制器主要采用ADI公司的fido5200和ADSP-CM408。两者结合，为网络连接的电机驱动器提供高度集成的芯片组。

fido5200是一个带有两个以太网端口的实时以太网多协议(REM)交换芯片。它在主机处理器和工业以太网物理层之间提供一个灵活的接口。fido5200包括一个可配置的定时器控制单元(TCU)，可针对各种工业以太网协议实现先进的同步方案。还可以借助专用定时器引脚实现输入捕获和方波信号输出等附加功能。定时器输入/输出与本地同步时间保持同相，因此也与网络流量保持同相。这使其不仅可以同步单个从机节点的I/O，而且可以同步整个网络中的从机节点。

REM交换芯片有两个以太网端口，因此可连接两个Phys (PHY1和PHY2)。该拓扑支持环形和线形网络。但在本实验设置中，

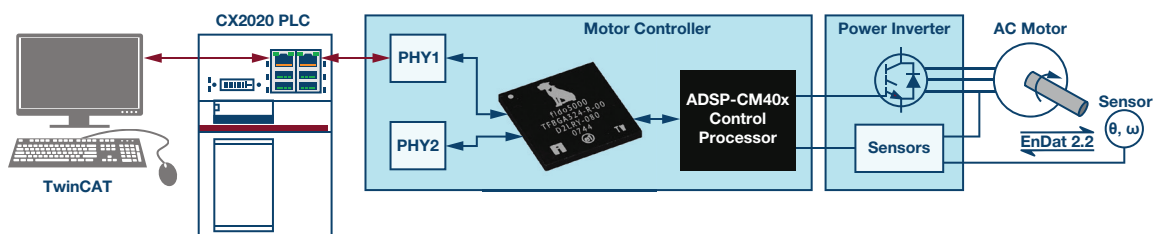


图8. 同步方案的实现。

作为演示说明，仅使用一个从机节点，并且只有一个以太网端口处于活动状态。

REM交换芯片通过并行存储器总线与主机处理器通信，确保了高吞吐量和低延迟。

用于实现电机控制器的主机处理器采用ADSP-CM408。它是基于ARM® Cortex®-M4F内核的专用处理器，用于实现控制和应用功能。该处理器包括支持工业控制应用的外设，如用于PWM逆变器控制的定时器、ADC采样和位置编码器接口。为了使所有外设与网络保持同步，采用了一个灵活的触发路由单元(TRU)。TRU将fido5200的TCU生成的触发信号重定向至ADSP-CM408上的所有时序关键型外设。这些外设包括脉宽调制器、用于相电流测量的sinc滤波器、ADC和绝对编码器接口。同步I/O的原理如图9所示。

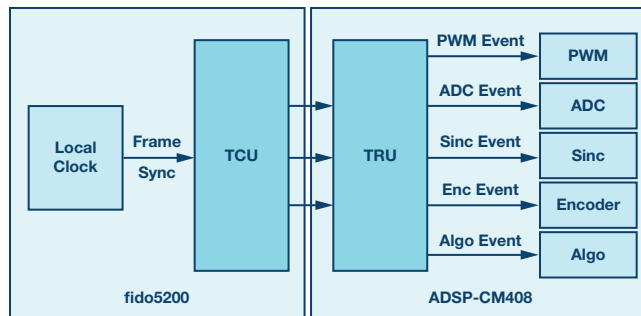


图9. 为I/O生成同步事件。

在图9中，请注意如何利用REM交换芯片上的TCU和电机控制处理器上的TRU来实现I/O事件调度器。换言之，该功能由两个集成电路实现。

电机控制器反馈三相伺服电机的相电流和转子位置。相电流使用隔离式 Σ - Δ ADC测量，转子位置则使用EnDat绝对编码器测量。 Σ - Δ ADC和编码器都直接连接至ADSP-CM408，无需任何外部FPGA或CPLD。

PWM开关频率为10 kHz，每个PWM周期执行一次控制算法。如本文所述，TCU在每个PWM周期内为ADSP-CM408提供一次同步脉冲。

实验结果

实验设置照片如图10所示。为了说明系统的同步功能，设置PLC使之运行一个持续200 μ s的程序任务。任务时间还决定了EtherCAT网络上的帧速率。电机控制器以PWM方式运行，并且控制更新周期为100 μ s (10 kHz)，因此需要以此速率生成同步脉冲。结果如图11所示。

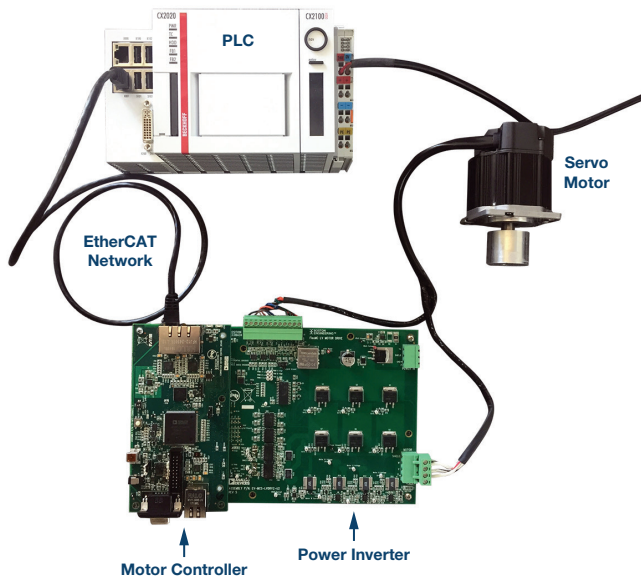


图10. 同步方案的实现。

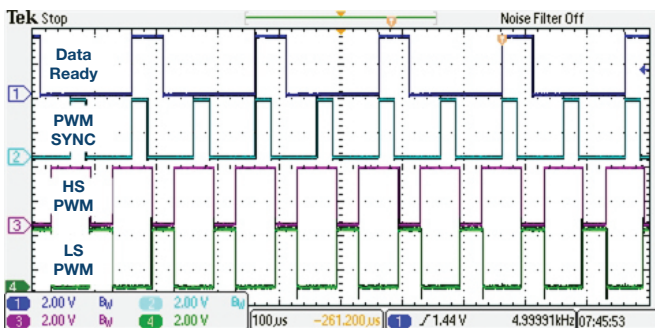


图11. 为I/O生成同步事件。

Data Ready（数据就绪）信号指示REM交换芯片何时向电机控制应用提供网络数据。信号每200 μs置位一次，与EtherCAT帧速率

Jens Sorensen [jens.sorensen@analog.com]是ADI公司系统应用工程师，负责工业应用的电机控制解决方案。他拥有丹麦奥尔堡大学电气工程学士学位。他的主要兴趣在于控制算法、电源电子和控制处理器。



Jens Sorensen

Dara O'Sullivan [dara.osullivan@analog.com]是ADI公司自动化、能源与传感器业务部电机和电源控制团队(MPC)的高级系统应用工程师。其专长领域是交流电机控制应用的功率转换和控制。Dara拥有爱尔兰科克大学工程学士、工程硕士和博士学位。自2001年起，Dara便从事研究、咨询和工业领域的工业与可再生能源应用方面的工作。



Dara O'Sullivan

Christian Aaen [christian.aaen@analog.com]是ADI公司的确定性以太网技术部的软件系统设计工程师。他的专业领域是嵌入式软件设计，并具有功率转换和电机驱动方面的技术背景。他拥有丹麦奥尔堡大学理学学士和理学硕士学位。



Christian Aaen

相对应。PWM同步信号也由REM交换芯片产生，用于使电机控制器的I/O与网络流量保持同步。由于PWM周期为100 μs，REM交换芯片每个EtherCAT帧调度两次PWM同步脉冲。图11中下方的两个信号HSPWM和LSPWM是其中一个电机相位的高端和低端PWM。请注意PWM信号是如何与网络流量同步的。

总结

实时以太网广泛用于运动控制系统，一些协议可实现精度小于1 μs的时间同步。但是，同步仅涉及网络主机和从机之间的数据通信。现有的网络解决方案不包括运动控制I/O同步，这限制了可实现的控制性能。

本文提出的同步方案可以实现从网络主机直至电机终端的全程同步。由于同步性能大幅改善，该方案能够显著提高控制性能。该方案还可提供跨多个轴的无缝同步。可以轻松地添加轴，并根据单个电机控制器定制同步。

同步基于I/O事件调度器，该调度器位于网络控制器和电机控制器之间。I/O事件调度器可实时高速编程，并且可进行调节以最小化抖动/频率变化效应。

本文提出的方案已经在实验设置中得到了验证，并展示了其结果。实验采用的通信协议是EtherCAT。然而，建议的方案适用于任何实时以太网协议。

总结

¹ Jie Ma, “基于EtherCAT的多自由度运动控制系统设计与实现。”2016年第六届仪器测量、计算机、通信与控制国际会议，2016年7月。