

7-3-2020

Modeling Computation Entity of CPS Based on Dynamic Behavior

Ye Feng

School of Computer Science and Technology, China University of Mining Technology, Xuzhou 221116, China;

Xia Yang

School of Computer Science and Technology, China University of Mining Technology, Xuzhou 221116, China;

Zhaoxiang Shen

School of Computer Science and Technology, China University of Mining Technology, Xuzhou 221116, China;

Yunchao Zhu

School of Computer Science and Technology, China University of Mining Technology, Xuzhou 221116, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Modeling Computation Entity of CPS Based on Dynamic Behavior

Abstract

Abstract: The Cyber-Physical System (CPS) has been highly concerned since it was proposed. However, the key of creating a good CPS system critically depends on the prophase design of the system model. Currently, it has become the focus and difficulty of research at home and abroad that models a CPS system with comprehensive, robustness, security. *According to the previous studies, the CPS system was divided into computing entity, physical entity and control entity. A modeling method of computation entity of CPS was provided, which is DBM-UML (Dynamic behavior modeling-UML). With this method, an analysis of the entire system's structure was given firstly and the corresponding state machine was abstracted. In another word, the system was modeled firstly by the dynamic behavior modeling of the state machine. Then the detailed modeling of the computation entity of the system by the UML modeling language was proposed. The logical relationship and feedback mechanism of the system were proposed and the system was modeled better.*

Keywords

CPS, UML, dynamic behavior modeling, state machine

Recommended Citation

Ye Feng, Xia Yang, Shen Zhaoxiang, Zhu Yunchao. Modeling Computation Entity of CPS Based on Dynamic Behavior[J]. Journal of System Simulation, 2016, 28(5): 1003-1008.

基于动态行为建模的 CPS 计算实体建模方法

叶枫, 夏阳, 申朝祥, 朱云超

(中国矿业大学计算机科学与技术学院, 江苏 徐州 221116)

摘要: 信息物理融合系统(Cyber-Physical System, CPS)自给出以来就备受关注。一个 CPS 系统设计的好坏关键取决于其设计前期的系统模型的创建。如何对一个 CPS 系统进行全方位、健壮性、安全性的建模是国内外研究的焦点和难点。在前人的基础之上, 把 CPS 系统分为物理实体、计算实体和控制实体, 给出了一种 CPS 系统计算实体的建模方法: 基于动态行为建模的 UML 建模方法 DBM-UML (Dynamic behavior modeling-UML), 对整个系统的结构进行详细的分析, 将其对应的状态机抽象出来, 先通过状态机的动态行为建模对系统进行模型化, 再通过 UML 建模语言对整个系统计算实体进行详细化建模, 把系统的逻辑关系和反馈机制等详尽的表现出来, 完成系统的计算实体建模。

关键词: 信息物理融合; UML; 动态行为建模; 状态机

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X (2016) 05-1003-07

Modeling Computation Entity of CPS Based on Dynamic Behavior

Ye Feng, Xia Yang, Shen Zhaoxiang, Zhu Yunchao

(School of Computer Science and Technology, China University of Mining Technology, Xuzhou 221116, China)

Abstract: The Cyber-Physical System (CPS) has been highly concerned since it was proposed. However, the key of creating a good CPS system critically depends on the prophase design of the system model. Currently, it has become the focus and difficulty of research at home and abroad that models a CPS system with comprehensive, robustness, security. According to the previous studies, the CPS system was divided into computing entity, physical entity and control entity. A modeling method of computation entity of CPS was provided, which is DBM-UML (Dynamic behavior modeling-UML). With this method, an analysis of the entire system's structure was given firstly and the corresponding state machine was abstracted. In another word, the system was modeled firstly by the dynamic behavior modeling of the state machine. Then the detailed modeling of the computation entity of the system by the UML modeling language was proposed. The logical relationship and feedback mechanism of the system were proposed and the system was modeled better.

Keywords: CPS; UML; dynamic behavior modeling; state machine

引言

信息物理系统(Cyber-Physical System, CPS)在



收稿日期: 2014-11-20 修回日期: 2015-01-23;
基金项目: 国家自然科学基金(61170066);
作者简介: 叶枫(1991-), 男, 江苏徐州, 硕士生, 研究方向为 CPS 建模方法; 夏阳(1962-), 男, 江苏徐州, 博士, 教授, 研究方向为网络计算, 电子商务, 系统仿真; 申朝祥(1989-), 男, 湖南邵阳, 硕士生, 研究方向为系统建模。

物联网的基础上给出通过人机交互接口实现和物理进程达到所谓的人、机、物的三维融合, 并给出从根本上解决信息空间与物理世界的交互问题的方法。在 CPS 系统中, 计算过程和物理过程通过有线网络和无线网络进行交互, 同时 CPS 系统又能在信息空间中控制物理实体^[1]。

UML 统一建模语言, 在面向对象软件设计过

程中起着非常重要的作用。作为一种成熟的建模语言, UML 在 CPS 建模理论研究中也是一种常见的工具。但是, UML 语言一般和 Simulink 相结合对 CPS 系统物理实体进行建模, 把 UML 应用到计算实体建模的相关研究相对较少。

有限状态机是描述离散动态的系统模型。通过状态精化的状态机可以对一个离散动态和连续动态结合起来的混合系统进行建模, 即动态行为建模。在混合系统中, 状态机可以适用于连续输入和输出, 还可以把离散动态和连续动态结合起来。混合系统是基于时间的模型和状态机模型的中间桥梁。这 2 种模型的结合, 为描述现实系统提供了一个完整的框架。动态行为建模理论是一种描述 CPS 系统计算实体模型的有效方法。

CPS 系统的核心技术是 3C(Computation, Communication, Control)的融合^[2]。计算、控制和通信, 每一个模块在系统中都扮演着不可替代的角色^[3]。本文以计算实体为核心, 给出一种新的 CPS 系统计算实体建模方法, 并运用该方法完成对智能车驱动模块计算实体的建模。

1 研究现状

CPS 系统分为: 计算实体、物理实体和控制实体。现在的 CPS 系统建模研究以计算实体和物理实体为主。其中计算实体建模以动态行为理论为基础, 而物理实体的建模和仿真则以 simulink/RTW 工具为主。计算实体主要描述系统的逻辑结构, 没有一个全局的时钟, 它是离散的基于事件驱动的。因此采用基于有限状态机的离散系统行为模型对 CPS 系统的计算实体模型进行构建是一种比较流行的建模方法^[4]。

目前对于 CPS 计算实体建模方法的研究比较广泛。文献[4]分析了构建计算实体模型的统一建模语言(UML)以及构建物理实体模型的 simulink/RTW 建模工具的可行性, 给出了基于 UML 框架的两种异质模型融合方法。其将 UML 语言应用到 CPS 系统计算实体建模的方法为本文提供了思路。

文献[5]以事件驱动为基础, 给出了不同类型事件之间的集成操作, 然后从集成事件中提取出环境的物理状态, 在物理状态的基础上给出了时空事件模型框架, 为连续时间系统和事件驱动的建模提供理论基础。文献[6]以一个具体的案例讨论了包括混合动力系统建模与仿真, 给出了特定领域的并行异构计算模型, 提高模块化和联合建模功能的实施架构。文献[7]建立了系统的静态结构图和动态行为图, 采用模型转换技术, 把 Simulink 建立的系统物理连续动态模型导入到 UML 模型, 基于实时性实现了 CPS 系统计算实体与物理实体融合的仿真建模。

计算学科一般将研究的重点放在逻辑关系上, 所以其在研究计算的时候往往容易忽略时间的因素, 直接将系统抽象为离散事件模型; 而在控制领域对物理世界的研究当中, 往往以时间为基础, 把系统抽象为连续的时间模型。在这个模型当中, 时间是其重要的坐标之一, 这将导致计算实体与物理实体在进行交互融合的时候非常容易产生冲突和无法预计的差错。

2 相关理论

统一建模语言 UML 是一种开放的方法, 用于说明、可视化、构建和编写一个正在开发的、面向对象的、软件密集系统的开放方法。UML 作为一种针对软件系统的建模语言, 提供了描述事务实体、性质、结构、功能、行为、状态和关系的建模元素, 并通过一组图描述由建模元素所构成的多种模型^[8]。

连续系统的建模, 可以使用 2 种截然不同的方法对系统进行描述。第 1 种是常微分方程, 是工程师必不可少的建模工具; 第 2 种就是参量模型建模方法, 是由软件建模与仿真工具驱动的较新的方法^[9]。这 2 种方法密切相关。一个系统可以利用多个模型进行建模, 而且这些模型与所建模的系统是有区别的。

嵌入式系统包括离散组件和连续组件。离散组件并不方便利用常微分方程进行建模, 因此, 需要引入状态机的概念。状态机是一个具有离散动态的

系统模型, 有限状态机(Finite-State Machine, FSM)是一个可能状态集 States 为有限的状态机。每个响应都是基于当前状态的从输入值到输出值的映射它反映从系统开始到现在时刻的输入变化。

混合系统是基于时间的模型和状态机模型的中间桥梁。这两类模型的结合, 为描述现实世界系统提供了丰富的框架。这其中包含 2 个关键思想: (1) 将离散事件嵌入到时机; (2) 系统要进行不同操作模式之间的离散转移时, 分层描述是非常有用的。如果将输入和连续状态结合在一起的判定式得到满足, 则发生状态转移。反过来, 与转移相关的动作设定目标模式的连续状态。混合系统可以适用于连续的输入和输出, 能够把离散系统和连续动态结合起来。

3 DBM-UML 建模方法

3.1 CPS 传统建模方法

CPS 系统建模一直是 CPS 系统研究的热点和难点。很多的业内人士也纷纷就不同的课题对 CPS 系统建模给出了自己的观点。简单的就单独的离散系统或连续系统进行建模或者仿真已经无法满足复杂的 CPS 系统建模仿真的需求。分别对 CPS 系统的 3 个实体分别进行建模并通过其他的手段将其连接已经是现在 CPS 系统建模的主要研究方向。

目前最广泛的一种建模趋势就是: 计算实体部分通过统一建模语言(UML)进行建模, 而物理实体部分则以 simulink/RTW 工具为主。CPS 系统计算实体没有统一的时钟, 是基于事件和逻辑关系的。UML 语言可以用面向对象图的方式来描述任何类型的系统, 能够对任何具有静态结构和动态行为的系统进行建模。所以通过 UML 语言对 CPS 系统的计算实体进行建模是非常恰当的选择。

3.2 DBM-UML 建模方法

本文给出了一种针对 CPS 系统计算实体部分的建模方法, DBM-UML(Dynamic behavior modeling-UML, 基于动态行为建模理论的 UML

建模方法)建模方法。即在对 CPS 系统动态行为建模的基础之上再进行 UML 建模。

具体的建模过程分 2 步:

(1) 动态行为建模

混合系统的行为可以利用模式转移的状态机分析工具以及精化系统的时序分析工具来理解, 混合系统的设计从两级进行: 状态机可以设计为得到模式转移的合理逻辑, 系统精化设计为实现每个模式下所需要的时序动作

对一个混合系统进行动态行为建模, 首先要抽象出混合系统的状态机。系统的离散值由状态机的各个状态表示, 系统的时序变化通过状态机的状态转化表示。系统的动态是时间上的演化和状态的改变。物理系统的模型要对产生的系统特性从某些方面进行描述, 所创建的物理模型需要具有可进行系统分析的数学特性。通过模型对系统特性进行模拟, 进而产生对系统的理解。

动态行为建模的关键是状态机的抽象。直观上, 一个系统的状态是指它在特定时间点的状况。状态会影响系统对输入做出怎样的响应。所以, 状态可以定义为对当前或下一个输入产生响应的先前输入的编码。状态间的转移决定状态机的离散动态以及从输入值到输出值的映射。

状态机提供的是一个系统行为建模的便捷方法。然而, 对于一个复杂的系统, 往往状态数目非常多, 可以看作是几个简单系统的组合。在对一个 CPS 系统进行动态行为建模时, 就需要进行状态机的组合。把建模初期罗列出的简单状态机按照逻辑顺序和时间顺序进行并发组合或者分层组合。组合后的状态机, 有些状态具有进一步精化的能力, 这些精化的状态自身就是状态机。把精化后的状态机和原来的状态进行组合, 得到最终的系统状态机。

(2) 创建状态机的 UML 模型

在动态行为建模完成之后, 再对已经建立的状态机模型进行 UML 建模。UML 语言是以能够描述任何类型的系统为目标的应用领域非常广泛的建模语言, 能够对任何静态系统或者动态系统进行

建模。UML 定义了 5 大类图，用来描述任何系统模型。DBM-UML 建模方法主要使用了用例图、类图和序列图 3 种图。这 3 种图能够把系统需要描述的结构和逻辑都描述到位。用例图用来描述系统的物理硬件参与者之间的逻辑关系和结构，类图用来描述系统的状态并细化到代码的层面使得后续的编程思路更加的清晰同时加强了模型的逻辑性，序列图用来描述系统的时序关系。这样整个系统的建模就完成了，在状态机模型的基础上的 UML 建模。图 1 展示了 DBM-UML 建模过程，即动态行为建模的基础上进行 UML 建模。

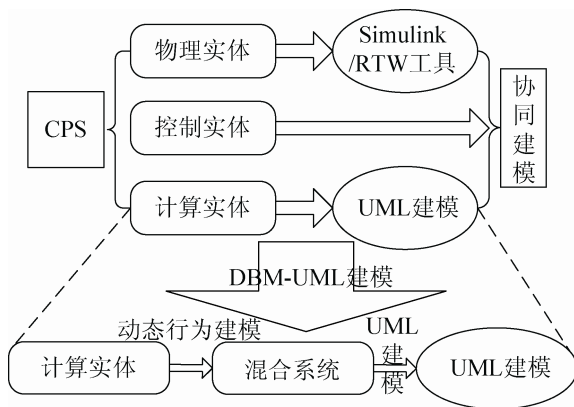


图 1 DBM-UML 建模图示

3.3 CPS 系统建模方法的对比

本文给出的 CPS 系统计算实体建模方法是在已有建模方法的基础上进行改进的。

文献[4]中计算实体建模采用有限状态机来刻画系统行为。但是，CPS 系统往往状态数目非常多，甚至无限。DBM-UML 建模方法的动态行为建模部分的核心也是状态机的抽象，和文献[4]不同的是，在解决复杂的混合系统中状态数目过多导致的有限状态机刻画系统行为吃力的缺点时，本文给出了状态机的组合方法，即把系统状态按照逻辑顺序和时间顺序进行并发组合或者分层组合，然后把具有进一步精化能力的状态进行状态精化。

文献[7]中对 CPS 系统计算实体仿真建模采用了 UML 语言。DBM-UML 建模方法与之不同的是首先对 CPS 系统进行动态行为建模，之后，在系

统的状态机模型基础上再进行 UML 建模。状态机是在原始系统上做了一阶抽象，把系统的逻辑关系和时序关系已经表达出来，在此基础上进行 UML 建模会更简单方便，而且系统结构更加清晰完整。

经过对比之后，DBM-UML 建模方法的优点如下：

(1) 动态行为建模能够有效的把连续系统和离散系统结合起来，实现基于时间的系统和基于逻辑的系统统一，可以很清晰的把系统的时间顺序和逻辑关系抽象出来。

(2) 先抽象出系统的状态机不但能够描述系统的结构特点和事件驱动时序，而且能够通过混合系统状态机的分析对系统进行改进以求更高的性能。加入状态机组合，解决由于状态过多导致的有限状态机建模吃力的缺点。

(3) 在清楚表述了系统逻辑关系的状态机的基础上进行 UML 建模，CPS 系统计算实体建模过程将变得更加清晰和完整，而且所建立的系统也是经过优化过的，能够达到更好的建模效果。

4 基于 DBM-UML 的智能车驱动模块建模实例

4.1 驱动模块设计

全国大学生智能汽车竞赛(飞思卡尔智能车大赛)是以智能车为研究对象的创意性科技竞赛，本节以智能车的驱动模块建模为例阐述 DBM-UML 建模方法的应用。

电动机采用 PWM 方波进行控制，由于 PWM 控制容易产生高次谐波使能量不能高效地转化成机械能，因此需要注意调节 PWM 频率，使电能得到充分利用。光电编码器的作用就是反馈当前智能车的速度，其返回值为脉冲个数。

电动机调速的最终目的是使智能车能够按照稳定的目标速度前行，并在必要的时候，能够及时做到弯道减速和直道加速，在速度控制的过程中利用 PID 算法进行控制。图 2 是智能车驱动模块闭环控制方法。

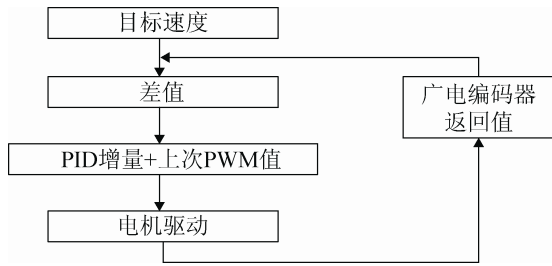


图 2 闭环控制

PID 控制中输入信号和输出信号的关系如公式(1)所示:

$$u(t) = K_p(e(t) + 1/T \int e(\tau) d\tau + T_D de(t)/dt) \quad (1)$$

在智能车驱动模块中, 设 T 为光电编码器采样周期; k 为采样序号, $k=0, 1, 2, \dots$; $e(k)$ 为系统在第 k 次采样时刻的偏差值; $e(k-1)$ 为系统在第 $k-1$ 次采样时刻的偏差值。当采样时间很短时, 可用一阶差分代替一阶微分, 用累加代替积分, 所以, 将连续时间离散化后得到的第 k 次采样输出 $u(k)$ 如公式(2)所示:

$$u(k) = K_p \{ e(k) + T / T_i \sum_{i=0}^k e(k) + T_D / T [e(k) - e(k-1)] \} + u_0 \quad (2)$$

设 $\Delta u(k)$ 为 PID 输出的增量差值。

$$\Delta u(k) = u(k) - u(k-1) \quad (3)$$

由公式(2)和(3)得出 PID 增量输出 $\Delta u(k)$ 的值如公式(4)所示:

$$\Delta u(k) = K_p [e(k) - e(k-1)] + K_i e(k) + K_D [e(k) - 2e(k-1) + e(k-2)] \quad (k \geq 2) \quad (4)$$

4.2 驱动模块的状态机模型

根据驱动模块设计中的闭环控制和离散 PID 公式, 通过动态行为建模方法, 把驱动模块的有限状态机抽象出来。

首先, 引入两个参数设置。 $f(k)$ 为第 k 次采样中光电编码器返回的 PWM 值, $f(m)$ 为目标 PWM 值。图 3 为驱动模块的状态机模型。这是一种状态机的串行组合, 把 PID 计算、PWM 控制和编码器采样 3 个状态机按照既定的顺序进行级联, 前者的输出作为后者的输入。整个级联的输入和输出以及状态的自转移构成了 PID 闭环控制的状态机。

inputs: $f(k), f(m)$

output: $f(k+1)$

根据图 2 中的驱动模块闭环控制设计, 构建单一模式状态机模型, 输入第 k 次采样的 PWM 值 $f(k)$ 和目标 PWM 值 $f(m)$, 输出第 $k+1$ 次 PWM 采样值 $f(k+1)$ 。如果两者不相等则进入 PID 闭环控制, 对 PWM 值进行调整, 直到光电编码器返回的下一次的 PWM 采样值和目标值相等, 不再对电动机的 PWM 微调。其中, 公式(4)中的第 k 次采样偏差值 $e(k) = \text{abs}[f(m) - f(k)]$, PID 增量为 $\Delta u(k)$, 上一次的 PWM 值为 $f(k-1)$ 。

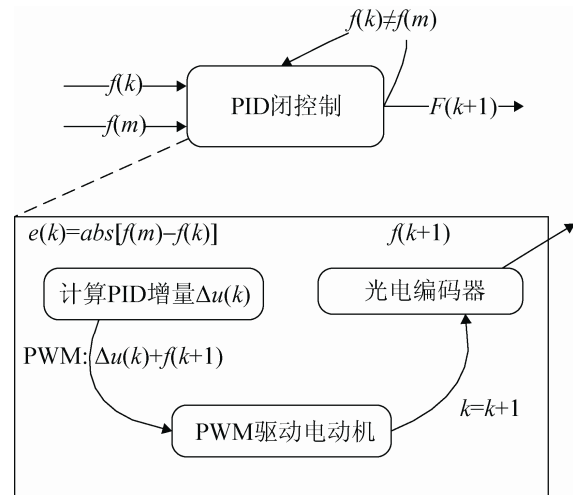


图 3 驱动模块状态机建模

4.3 驱动模块的 UML 建模

(1) 驱动模块用例设计

根据驱动模块的状态机模型, 驱动模块的 UML 建模创建 3 个用例, 分别是 PID 控制、电动机和光电编码器。其中 PID 控制是核心模块, 用来计算 PID 增量并将其转换成驱动电动机的 PWM 值。电动机是驱动模块的物理核心, 根据 PID 控制得到的 PWM 值驱动电动机。光电编码器则是对电动机进行定时采样, 将采样值反馈给 PID 控制, 根据差值做出微调。

(2) 驱动模块类图设计

在驱动模块类图设计中, 一共设计了 5 个类。分别是时间基类 Timebase、采样序号类 SampleNO、

编码器反馈类 *Sampler*，PID 控制类 *PIDcontrol* 以及电动机驱动类 *Motor*。图 4 即为驱动模块 UML 建模中的类图设计。

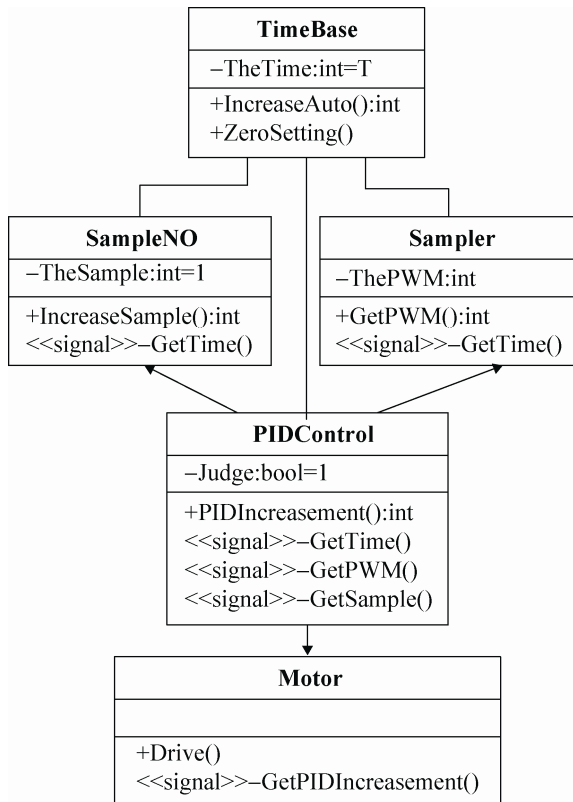


图 4 驱动模块类图设计

在类图设计中 *Timebase* 类提供一个全局时钟，设常数 T 为一个周期，即为采样周期。函数 *IncreaseAuto()* 为系统时间自增，当时间值为 T 时，通过函数 *ZeroSetting()* 归零。*SampleNO* 类增加和记录系统采样序列，*TheSample* 类记录当前采样序列号，当获取的系统时间为一个采样周期时，通过 *IncreaseSample()* 函数增加一次采样序号。*Sampler* 类获取光电编码器的采样值，当获取的当前系统时间为一个采样周期时，通过函数 *GetPWM()* 获取当前电动机的转速，并转换成 PWM 值，存储到 *ThePWM* 中。*PIDControl* 类是驱动模块的核心类，首先获取系统时间、上次采样值以及本次采样值，通过 *Judge()* 函数判断当前电动机的 PWM 值是否和设定的值相等，如果不相等则进行 PID 微调。通过 *PIDIncrement()* 函数计算 PID 增量并计算出相

应的 PWM 值。*Motor* 类则是电动机的驱动函数，根据 *PIDControl* 类获得的 PWM 值驱动电动机，通过驱动函数 *Drive()* 函数使电动机按照指定的速度转动。

(3) 驱动模块序列图设计

驱动模块序列图设计如图 5 所示。驱动模块的物理核心是电动机，首先从 *Timebase* 中获取系统时间，在一次采样周期到来时，通过 *SampleNO* 增加并记录一次采样序号，从 *Sampler*

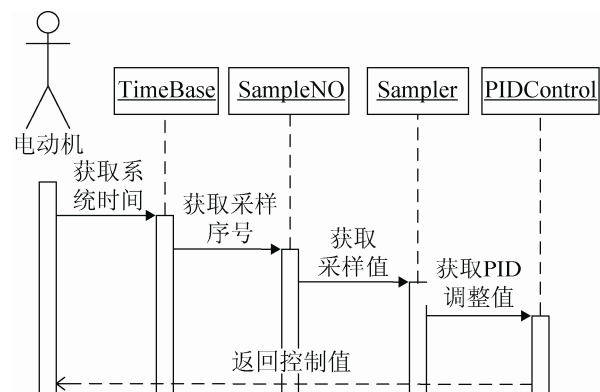


图 5 驱动模块序列图

获取光电编码器采集到的本次采样的电动机 PWM 值。将本次采样值传递给 *PIDControl*，通过判断计算出需要调整的 PID 增量，返回给电动机驱动程序调整后的 PWM 值。

5 结论

本文的主要研究是 CPS 系统计算实体部分的建模方法。在简单介绍 CPS 系统和 CPS 系统建模相关研究的基础上，依据动态行为建模理论，给出了一种新的 CPS 系统计算实体建模方法 DBM-UML 建模方法。该方法的核心思想是以动态行为建模理论为基础，抽象出 CPS 系统计算实体相应的混合系统状态机，这种模型能够把离散系统和连续系统结合起来，解决了计算实体建模只依据系统逻辑而缺乏统一时间基的问题以及有限状态机无法描述系统状态过多的问题，从而，完整的把系统计算实体的逻辑关系和时序关系描述出来。在状态机的基础上，对系统进行 UML 建模，建模过

(下转第 1016 页)