

基于全局状态观测器的网络控制系统节点同步控制方法^①

何 芳^{②*} 佟为明^{*} 王 强^{**}

(^{*} 哈尔滨工业大学电气工程及自动化学院 哈尔滨 150001)

(^{**} 济南大学机械工程学院 济南 250022)

摘 要 提出了一种采用全局状态观测器实现网络控制系统(NCS)节点同步控制的方法,即在控制器节点建立全局状态观测器,用以收集各网络节点的运行状态信息,计算出全局状态控制变量决策 NCS 的同步控制行为,实现 NCS 多个传感器节点同步采集或多个执行器节点同步输出控制。给出了全局状态观测器的设计,并依据观测器状态输出变量,采用两次同步激发控制使多个网络节点同时切入同步处理程序。该方法提供了一种可行的解决 NCS 节点同步控制问题应用程序设计思路,可以消除 NCS 节点同步控制误差,减小由于节点同步控制误差给系统控制性能带来的不良影响。

关键词 全局状态观测器,网络节点,同步控制,事件驱动方式

0 引 言

在很多复杂自动控制系统中,控制器的控制决策需要融合多个传感器的数据采集信息。在这类应用系统中,对各传感器数据采样的时间同步控制非常重要,如果各传感器数据采样的时间同步误差较大,则很难保证控制器决策的准确性,严重时会导致控制器做出错误判断而造成系统瘫痪。另外,在有些复杂自动控制系统中,被控对象需要由多个执行器进行协同控制,如机器人、多电机同步传动控制系统等^[1,2]。在这类应用系统中,对多个执行器的同步输出控制非常重要,如果各执行器的输出时间同步控制误差较大,则很难保证被控对象动作的协调一致,致使系统的控制精度降低。

如果采用传统的集中控制系统结构,系统中的传感器、控制器和执行器是通过点对点的专用线进行连接,系统的可靠性、抗干扰性和可拓展性都比较差,故障诊断和维护困难。特别是,当实际应用控制对象地域分散或系统的复杂度增加时,这些弊端就显得更加突出。将网络控制技术引入这些复杂同步控制系统可以很好地解决复杂接线带来的弊端。采用网络控制系统(networked control system, NCS)是指通过网络形成的反馈控制系统,可以使众多的传感

器、执行器和控制器等系统的主要功能部件通过网络相连接,相关的信号和数据通过通信网络进行传输和交换,从而可避免使用彼此间的专用连线,方便安装与维护,可增加系统的灵活性和可靠性,并可实现资源的共享、远程操作和控制。因为 NCS 具有多种优势,所以被广泛应用于电力、交通、机器人、航空航天等各种系统和远程控制领域。

网络的介入给控制系统带来各种优点的同时,也带来一些具有网络特征的新问题^[3,4]。其中,如何解决网络分布节点的同步控制问题是其中一个研究热点。特别是当对 NCS 节点的同步输入或同步输出性能要求很高时,采用恰当的同步控制策略就显得尤其重要^[5]。本文针对有高性能同步控制要求的多输入多输出 NCS,对 NCS 的同步控制及其性能评价指标进行了讨论,提出了一种通过建立网络节点的运行状态全局观测器决策 NCS 的同步控制行为的同步控制策略,以实现 NCS 多个传感器节点对被测信号的同步数据采集或多个执行器节点对控制信号的同步响应输出。

1 NCS 节点的同步控制及性能评价指标

1.1 NCS 节点的同步控制

NCS 的任务由网络中的传感器、控制器和执行

^① 国家自然科学基金(50875111)和山东省自然科学基金(ZR2009GM027, Y2006G25)资助项目。

^② 女,1964 年生,博士,教授;研究方向:工业网络控制系统,电气传动自动控制系统;E-mail:hefang7588@163.com
(收稿日期:2009-10-22)

器许多节点共同来完成。NCS 节点同步控制是指网络节点的动作时刻一致性的同步控制,包括网络节点的输入同步控制和输出同步控制。NCS 节点的输入同步控制主要关注实现对分布在不同物理位置的传感器测量信号的数据同时刻采样。更具体地说,它既包括对模拟量输入信号 A/D 采样转换启动时刻的同步控制,也包括对开关量传感器输入信号的同时刻提取。NCS 节点输出同步控制主要关注 NCS 的执行器节点的输出同步控制,是指多个执行器节点在接收到控制器的统一协同控制命令后,在同一时刻响应命令。

目前针对 NCS 的同步控制问题,主要包括控制系统和控制网络两方面的解决方案。在控制系统方面,可以借鉴传统控制系统的一些同步误差补偿控制方法;在控制网络方面,因为 NCS 的多个传感器节点的输入同步控制和多个执行器节点的输出同步控制的同步控制性能会受到节点时钟精度、网络时延、节点的驱动方式、节点的采样频率以及应用程序的任务调度等多方面因素的影响,因此采取的控制策略也各不相同。对这些影响因素,目前主要采用包括硬件和软件两方面的解决方案。在硬件方面,不同类型的工业网络有它自己的同步协议和硬件设计支持,各有优势又有其局限性。在软件方面,主要是通过一些控制算法来提高 NCS 同步控制性能,如节点时钟同步控制^[6]、时延的预测补偿控制^[7,8]、网络任务调度控制^[9]以及网络节点工作方式的合理设置等。软件设计实现方法在应用系统设计中具有更多的灵活性。

采用先进的控制算法在一定程度上可以减小由于网络特性造成的 NCS 节点同步误差,提高 NCS 的同步控制性能,但它们一般都有一定的限定条件,而且有些计算方法较为复杂,还只是仿真研究效果比较理想,真正应用于实时的实际应用系统设计中还比较困难。因此,采用先进的控制算法一方面要考虑提高系统的同步控制性能,另一方面也要考虑采用的算法应计算简便、可实施性强。本文提出的同步控制方法正是兼顾了这两方面要求。

1.2 同步控制性能评价指标

NCS 节点的同步控制性能评价指标包括节点的输入同步控制性能评价指标和节点的输出同步控制性能评价指标。

1.2.1 节点的输入同步控制性能评价指标

在 NCS 中,将需要进行同步数据采样的网络节点划分在同一个同步输入控制组中,用这些节点之

间的采样时刻相对时间同步误差评价 NCS 节点的输入同步控制性能。

定义 NCS 第 i 个传感器节点的采样输入同步误差 τ_{is} 为它的采样时刻 t_i 相对于基准时刻 t_n 的时间差,如式

$$\tau_{is} = t_i - t_n, \quad i = 1, 2, \dots, n \quad (1)$$

所示。其中 n 为需要进行同步采样的传感器节点的个数。 n 个传感器节点响应共同的同步控制命令。可以选用 n 个传感器节点中的任意一个节点开始采样的时刻作为基准时刻 t_n 。采用方差概念评价传感器节点采样同步输入控制性能,用 J_1 表示,取 n 个传感器节点的采样输入的同步误差 τ_{is} 的平均值作数学期望,如式

$$J_1 = \frac{1}{n} \sum_{i=1}^n (\tau_{is} - \frac{1}{n} \sum_{i=1}^n \tau_{is})^2 \quad (2)$$

所示。 J_1 越小则表示采样输入的同步控制性能越好。

1.2.2 节点的输出同步控制性能评价指标

由于控制器节点对各传感器节点的采样数据是分时处理的,所以它向每个执行器发送命令的时间是有先后的,由此造成各执行器节点接收和响应命令时刻存在差异,产生输出同步控制误差。

将需要实施输出同步控制的网络节点划分在同一个输出同步控制组中,定义 NCS 第 j 个网络节点的输出执行时刻 t_j 相对于基准时刻 t_m 的时间差为输出控制的相对同步误差 τ_{aj} , 如式

$$\tau_{aj} = t_j - t_m, \quad j = 1, 2, \dots, m \quad (3)$$

所示。其中 m 为需要同步输出的执行器节点的个数。确定其中一个执行器响应控制器同步输出控制命令的时刻为基准时间 t_m 。采用方差概念评价执行器节点输出的同步控制性能,用 J_2 表示,取 m 个执行器节点响应同一个同步控制命令的同步误差 τ_{aj} 的平均值作数学期望,如式

$$J_2 = \sum_{j=1}^m |t_j - t_m| = \frac{1}{m} \sum_{j=1}^m (\tau_{ja} - \frac{1}{m} \sum_{j=1}^m \tau_{ja})^2 \quad (4)$$

所示。 J_2 越小则表示执行器输出同步控制性能越好。

为减小 NCS 节点的输入同步误差或输出同步误差,获得更好的输入同步控制性能指标 J_1 或输出同步控制性能指标 J_2 ,主要是要考虑影响同步控制性能的各种因素,采用控制策略使多个节点的同步输入或同步输出的“启动”时刻一致,即传感器节点的采样时刻 t_i 或执行器节点的输出执行时刻 t_j 时刻

一致。

2 采用全局观测器同步控制策略

本文提出的采用全局观测器的 NCS 节点同步控制策略,从本质上说是依据建立的 NCS 全局状态观测器的运算结果作激发事件对网络节点采用节点事件驱动工作方式的一种同步控制策略,避开了许多影响网络时延的不确定因素造成的节点同步误差,可以简化同步控制算法。

2.1 基于全局观测器的同步控制基本思路

在分布式的 NCS 中,主从结构是一种最常见的结构。下面以主从结构 NCS 为研究对象讨论采用全局观测器的同步控制基本思路。NCS 的全局状态观测器设置在主站节点,它的作用是收集各网络节点的运行状态信息,计算输出决策网络节点同步控制的全局状态变量。

基于全局状态观测器的同步控制思想主要包括以下几点:

(1) 首先按照 NCS 实际系统的同步控制要求对 NCS 的网络节点进行分组,以便同步控制命令可以按组进行实施。

(2) 将每个从站节点自己的主循环程序分割成多个程序进程,设置节点的进程状态表。每段程序进程完成相对独立的功能。在每个进程的开始和结尾,节点的进程状态表被刷新。

(3) 从站节点的进程状态表要传送给主站,主站的全局状态观测器将运用节点的进程状态信息和同步处理状态信息进行计算,其计算输出将作为发布同步控制命令的判断条件。

(4) 从站节点在自己的主循环程序每个程序进程结束后,要检查是否有主站的同步执行要求。

(5) 采用控制器节点发送两次同步激发信号的具体实施方法,实现传感器从站节点同步输入控制组或执行器从站节点同步输出控制组的各节点的同步控制。其中,第 1 次同步激发信号是同步执行准备要求,在第 2 次同步激发信号后各网络节点同步切入它们各自的同步处理程序。

(6) 对每个同步控制节点,由设计者明确它们各自的同步条件和设定同步条件激活后的同步响应表,以便进入它们各自相应要求的同步处理。

2.2 同步控制节点分组

采用全局观测器实现 NCS 节点同步控制首先要对系统中的节点进行分组。对 NCS 同步控制节

点进行分组是将需要实现共同的同步控制目标的几个网络节点划分在一个组里。如果 NCS 中包含几个不同的同步控制目标,则可以在 NCS 中设置多个同步输入控制组和多个同步输出控制组。

假设一个 NCS 中包含 N 个不同的同步控制目标,可以设置 N 个同步控制组,则 NCS 同步控制节点的分组可表示为

$$Group = \{ Group_1, Group_2, \dots, Group_N \}$$

$$Group_i = \{ node_{i1}, node_{i2}, \dots, node_{in} \}$$

其中 $Group$ 表示所有同步控制节点组的集合, $Group_i$ 表示第 i 个同步控制组内网络节点的集合, $node_{ij}$ 代表第 i 个同步控制组内第 j 个网络节点, $i = 1, 2, \dots, N$; $j = 1, 2, \dots, n$ 。

NCS 节点的分组原则包括以下几点:

(1) 拥有共同的同步控制目标,例如同步输出或同步输入;

(2) 拥有相同的同步控制条件,例如同步开始控制条件或同步结束控制条件;

(3) 需要完成相互关联的同步响应任务,例如它们的输入或输出量值之间要求必须满足确定的函数关系。

对同步控制节点进行分组的意义主要表现在以下几点:

(1) 在一个 NCS 中,并不是对所有的节点都有高性能同步控制要求,因此可以只对有高性能同步控制要求的节点实施同步控制策略,而对那些同步控制性能要求不高的网络节点仍然可以采用一般周期性时间控制或非周期性事件控制。

(2) 在一个 NCS 中,需要实施同步控制的几个节点一般拥有一个共同的或密切关联的控制目标,当同步控制目标不同时,应采用不同的同步控制策略。因此,对 NCS 节点实施分组同步控制可以使同步控制策略的制定和实施具有更明确的针对性,有利于同步控制命令的组织和实施。

(3) NCS 同步控制性能的评价一定是针对确定的同步控制要求而言,即同步控制性能实质上只是关注那些实现一个共同的或密切相关同步控制要求的节点之间的输入或输出相对同步误差。

2.3 节点程序的进程分割及同步流程

根据提出的基于全局状态观测器的同步控制思想,每个从站节点的主循环程序应被分割成多个程序进程,在每个进程的开始和结尾节点的进程状态表被刷新。假设某一个网络节点主循环程序被分割成 p 个进程,则该节点的进程可表示为

$Process = \{ process_1, process_2, \dots, process_p \}$

NCS 节点的程序进程分割及同步流程图如图 1 所示。

采用这种应用程序设计方式的目的是:一方面因为在每个进程结束后都进行同步控制判断扫描,检查是否有同步执行要求,所以可使多个节点的同步控制要求可以尽快得到响应;另一方面保证每个程序进程在执行期间不会被同步控制中断。

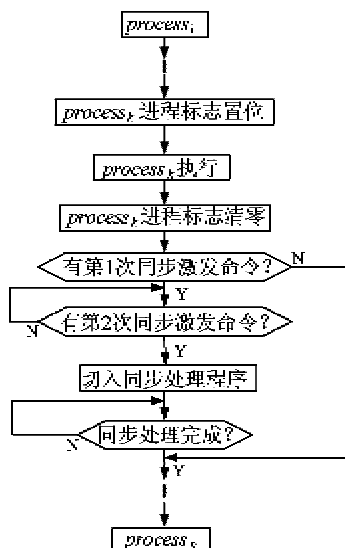


图1 NCS 节点的程序进程分割及同步流程图

2.4 全局状态观测器

NCS 的全局状态观测器是个多输入多输出控制器,如图 2 所示。对于其中的一个同步控制组,输入信号是该同步控制组中每个同步控制从站节点的同步控制状态信息;输出信号是控制组从站节点同步控制的决策信号。

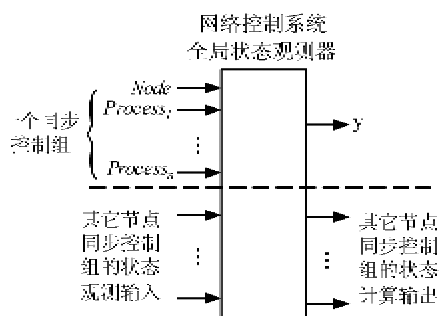


图2 NCS 节点同步控制全局观测器

假设一个主从 NCS 包含 N 个同步控制组,其中一个同步控制组的 n 个从站节点有共同的同步控制行为要求。全局观测器对这 n 个从站节点组成的同步控制组的状态观测与控制如图 2 所示,相对

应的输入、输出变量描述如下:

(1) 输入变量 $Node$

$Node$ 表示同步控制组的 n 个网络节点的同步处理状态,如式

$$Node = \{ node_1, node_2, \dots, node_n \}$$

所示。其中, $node_i = 0$ 或 $node_i = 1$, $i = 1, 2, \dots, n$ 。 $node_i = 0$ 表示第 i 个网络节点正在执行同步处理程序, $node_i = 1$ 表示该网络节点当前状态是非同步处理状态。 $node_i$ 状态值由从站节点反馈。

(2) 输入变量 $Process_i$

假设第 i 个网络节点具有 p 个进程, $Process_i$ 表示第 i 个节点的程序进程运行状态,则

$$Process_i = \{ process_{i1}, process_{i2}, \dots, process_{ip} \}$$

其中, $process_{ij} = 0$ 或 $process_{ij} = 1$, $i = 1, 2, \dots, n$ 且 $j = 1, 2, \dots, p$ 。 $process_{ij} = 0$ 表示第 i 个网络节点的第 j 个程序进程正在运行中。

(3) 输出变量 y

y 是全局观测器计算最终输出的同步控制全局变量。假设 y_i 表示第 i 个网络节点是否可以进入同步控制的状态输出变量, $y_i = 1$ 或 $y_i = 0$, $y_i = 1$ 表示第 i 个网络节点可以进入同步。因为只有同步控制组内的所有网络节点都允许同步执行后,才能切入同步控制处理,则 y 可通过 y_i 状态输出变量的逻辑交运算由式

$$y = y_1 \wedge y_2 \wedge \dots \wedge y_n \quad (5)$$

计算推出。如果运算结果为 $y = 1$ 输出,同步组节点将进入同步处理模式,否则不能进入同步处理。

式(5)中的 y_i 可以通过式

$$y_i = node_i \wedge (process_{i1} \wedge process_{i2} \wedge \dots \wedge process_{ip}) \quad (6)$$

推算出来。如果计算结果为 $y_i = 1$,则表示 i 个网络节点已准备好进入同步控制处理;如果计算结果为 $y_i = 0$,则表示 i 个网络节点还没准备好进入同步控制处理。

2.5 基于全局观测器的两次同步激发

为了实现从站节点的输入或输出的同步控制,以设置的 NCS 的全局状态观测器的运算结果为决策条件,本文提出采用发布两次激发信号来实现节点同步控制的实施方法。

主从结构 NCS 中,当主站要求从站同步组执行同步响应任务时,给同步组的各从站发送第 1 次同步激发信号。每个从站在当前进程结束后,刷新它的程序进程表,并检查是否有同步执行要求信号,没

有同步要求则进行下一个进程,如果有则进入等待状态。主站节点在发出第 1 次同步控制激发后,不断刷新各从站程序进程状态表,全局状态观测器进行运算,直至所有节点当前进程全部结束,全局状态观测器运算输出变量 y 变为 1,即刻发送第 2 次同步控制激发信号。在收到第 2 次同步控制激发信号以后,所有的同步控制节点进入同步处理操作,保证了每个节点的同步处理从同一时刻开始。从站在同步处理结束后将回到主循环流程。

以下用一个例子来描述采用两次同步激发信号实现节点同步控制的方法,并对前面提出的用于同步控制的 NCS 状态观测器进行计算验证。假定主从结构 NCS 包含 3 个从站节点,3 个节点的程序均被分为 6 个进程,两次同步激发信号的发送时刻分别是 t_1 和 t_2 ,如图 3 所示。

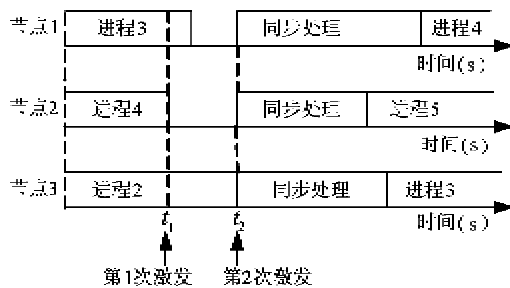


图 3 采用两次激发信号控制同步的示意图

在 t_1 时刻,主站控制器发送同步触发请求给同步节点组,即第 1 次同步激发信号。此时只有节点 2 完成了当前的进程,节点 1 的进程 3 还没有完成,节点 3 的进程 2 还没有完成,则 t_1 时刻有

$$Node = \{1, 1, 1\}$$

$$Process_1 = \{1, 1, 1, 0, 1, 1\}$$

$$Process_2 = \{1, 1, 1, 1, 1, 1\}$$

$$Process_3 = \{1, 1, 1, 1, 0, 1\}$$

根据式(5)和式(6),推出

$$y_1 = 1 \wedge (1 \wedge 1 \wedge 1 \wedge 0 \wedge 1 \wedge 1) = 0$$

$$y_2 = 1 \wedge (1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1) = 1$$

$$y_3 = 1 \wedge (1 \wedge 1 \wedge 1 \wedge 1 \wedge 0 \wedge 1) = 0$$

$$y = y_1 \wedge y_2 \wedge y_3 = 0 \wedge 1 \wedge 0 = 0$$

即观测器输出 $y = 0$,3 个节点没有进入同步处理。已经完成当前程序进程的节点进入等待状态。

在 t_2 时刻,3 个节点都完成当前的程序进程:

$$Node = \{1, 1, 1\}$$

$$Process_1 = \{1, 1, 1, 1, 1, 1\}$$

$$Process_2 = \{1, 1, 1, 1, 1, 1\}$$

$$Process_3 = \{1, 1, 1, 1, 1, 1\}$$

根据式(5)和式(6),推出

$$y_1 = 1 \wedge (1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1) = 1$$

$$y_2 = 1 \wedge (1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1) = 1$$

$$y_3 = 1 \wedge (1 \wedge 1 \wedge 1 \wedge 1 \wedge 1 \wedge 1) = 1$$

$$y = y_1 \wedge y_2 \wedge y_3 = 1 \wedge 1 \wedge 1 = 1$$

即观测器输出 $y = 1$,此时发送第 2 次同步激发信号使 3 个网络节点进入同步处理应用程序,实现了 3 个从站节点的同步控制。同步处理应用程序可以是传感器节点的同步采样输入程序或执行器节点的同步控制输出程序。

进入同步处理后,置 $Node = \{0, 0, 0\}$,节点不再响应其它同步控制激发信号。第 i 个节点的同步处理结束后, $node_i$ 又改变为 1。

3 结 论

通过以上讨论可以看出,本文提出的采用全局状态观测器实现 NCS 节点同步控制的方法需要通过设计者的应用编程实现。通过建立 NCS 全局观测器,并采用同步控制节点分组、节点程序进程分割以及两次同步激发控制等控制策略,可以使 NCS 多个网络节点同步切入数据采集程序或输出控制程序,从而实现分布式传感器节点数据采集的同步控制或执行器节点的输出同步控制。

这种采用全局观测器实现 NCS 节点同步控制的方法,实质上是采用了 NCS 节点事件驱动方式,重要的是通过全局观测器的计算输出得到网络节点的同步控制驱动事件,避免了许多影响网络时延的不确定因素造成的节点同步误差,可以简化同步控制算法,减小节点的同步控制误差,提高 NCS 的同步控制性能。

参考文献

- [1] 戴先中,刘国海.两变频调速电机系统的神经网络逆同步控制.自动化学报,2005,31(6):890-900
- [2] He F, Tong W M, Wang Q. Synchronization control strategy of multi-motor system based on profibus network. In: Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, 2007. 873-878
- [3] 杨丽曼,李运华.网络控制系统的时延和负载分析.控制与决策,2006,21(1):68-72
- [4] 孙海燕,侯朝桢.多输入多输出网络控制系统时延同步研究.弹箭与制导学报,2005,25(4):258-260

- [5] Fu L, Dai G Z. Delay characteristics and synchronization architecture of networked control system. In: Proceedings of the 1st International Symposium on Systems and Control in Aerospace and Astronautics, Harbin, China, 2006. 1056-1060
- [6] 张彦军, 刘靖宇, 郭廓. 分布式控制系统的时钟同步研究. 工业控制计算机, 2006, 19 (5): 20-24
- [7] 赵辉, 邓燕, 王红君等. 基于预测控制的网络时延补偿策略研究. 仪器仪表学报, 2008, 29(9): 1923-1927
- [8] Hassapis G. Implementation of predictive control algorithms on networked control systems. In: Proceedings of the American Control Conference, Minneapolis, Minnesota, USA, 2006. 1936-1941
- [9] 周黎辉, 王天堃, 徐大平. 网络控制系统调度与控制综合设计研究综述. 华北电力大学学报, 2008, 35(1): 31-35

Synchronous control of nodes of NCS based on global state observer

He Fang^{*}, Tong Weiming^{*}, Wang Qiang^{**}

(^{*} School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin 150001)

(^{**} School of Mechanical Engineering, University of Jinan, Jinan 250022)

Abstract

A method for synchronous control of nodes of a networked control system (NCS) by a global state observer is put forward in the paper. The method establishes a global state observer in the controller node of the NCS to collect the state information of every network node, to calculate the global state variable of the synchronous control, to determine the synchronous behavior of the NCS, and finally realize synchronous acquisition of multiple sensor nodes or synchronous output control of multiple actuator nodes. The design of the global observer is described. Based on the output variable of the global state observer, multiple network nodes are switched to their synchronous processes by adopting twice active controls. This method provides a feasible solution to the applied design of the NCS node synchronous control, which can eliminate synchronous control error among multiple network nodes, as well as decrease the negative influence of synchronous control error on control performance of NCS.

Key words: global state observer, node of NCS, synchronous control, event-driven mode