

# 多新息优化 MFAC 控制律在网络控制系统中的应用

陈晓青, 秦品乐, 刘毛毛, 常江

(中北大学, 太原 030051)

**摘要:** 网络控制系统建模难度大或代价高, 并且网络环境受时延、丢包、其他节点干扰的影响时刻都在变化, 使用传统的控制方法已经难以满足控制要求, 在网络控制系统中引入仅利用当前在线数据的无模型自适应控制, 从而避开了系统建模问题, 同时为了更好地适应网络环境变化, 将多新息理论应用于无模型方法的控制律计算, 加快其收敛速度, 达到准确快速控制的目的; 利用 TrueTime 仿真工具箱, 搭建了一个以太网网络控制系统的仿真平台; 仿真结果证实了无模型自适应控制算法在非线性系统下的控制效果优于 PID, 并且证实了使用多新息理论改进无模型控制律算法的有效性。

**关键词:** MFAC; 多新息; 网络控制系统

## MFAC Control Law Optimization Using Multi-Innovation and its Application in Network Control System

Chen Xiaogin, Qin Pinle, Liu Maomao, Chang Jiang

(North China University, Taiyuan 030051, China)

**Abstract:** Networked control system modeling is difficult with large cost, and the network environment is changeable because of delays, package loss and interference from other nodes. The traditional control method can't meet the network control system demand. Model-free adaptive control algorithm is introduced to network control system which only use system's online data in calculation, to avoid modeling problem. And in order to adapt the network environment faster and more accurately, using multi-innovation theory to improve MFAC control law, to accelerate the convergence rate. We use truetime toolbox to build an ethernet network control system simulation platform. And simulation results confirm the model-free adaptive control algorithm results in a nonlinear system is superior to PID, and confirmed the validity of the MFAC control law improved multi-innovation theory.

**Keywords:** MFAC; multi-innovation; NCS

### 0 引言

网络控制系统 (networked control system, NCS) 通过公用或专用的通信网络来代替传统控制系统中的点对点结构, 与传统点对点结构的控制系统相比, 网络控制系统具有成本低、功耗小、安装与维护简便、可实现资源共享、能进行远程操作等优点<sup>[1]</sup>。网络控制系统的出现在一定程度上为控制系统的发展带来了好处, 但网络控制系统也具有网络特性和结构复杂性, 网络的引入带来了传输不稳定的问题, 如丢包、时延等, 目前已经有不少学者研究网络因素对网络控制系统的影响<sup>[2]</sup>。不少学者们在研究网络因素对 NCS 的影响时, 都需要事先对系统进行建模, 然而, 由于网络控制系统的结构趋于复杂, 规模较大, 大大增加了系统建模的难度, 为了方便研究, 往往需要简化受控模型或花费大量的工作在 NCS 的建模上, 给研究网络控制系统增加了额外的负担。在传统的控制理论中, 通常会使用无模型控制算法避开系统建模问题<sup>[3]</sup>, 目前使用最为广泛的无模型控制算法为 PID 控制算法, 但 PID 控制一般只适用于单线性、非时变或动态特性只有很小变化的对象, 在处理

具有强非线性、时变性和周期性扰动的系统控制问题时其控制效果不甚理想, 另外, 由于丢包、网络时延、节点竞争等一系列随机的网络因素的存在, 使得网络控制系统的通信环境变得难以预测, 若继续使用传统的控制方法将无法适应网络控制系统的变化, 从而造成系统失控。

由侯忠生教授<sup>[4]</sup>提出无模型自适应控制理论 (model-free adaptive control, MFAC) 也是一种无模型控制方法, 仅使用系统当前的在线输入输出数据对系统进行控制, 目前已经应用到多个领域并且取得了良好的控制效果。相比于常用的 PID, MFAC 具有如下几个优点使其更加适合应用到 NCS 中, 第一, 网络控制系统的规模和结构往往较为复杂, 并且网络环境随时都在变化, MFAC 只依赖于当前的在线数据对系统进行控制, 可以很好地适应系统的变化。第二, MFAC 的计算仅仅只依赖于当前数据, 也就意味着可以单独开发 MFAC 控制器, 降低成本。第三, MFAC 的控制原理简单易用, 计算量小, 并且鲁棒性强, 针对网络时延大的 NCS 更具有适用性。

将 MFAC 的应用到网络控制系统虽然避免了系统的建模问题, 但在时延较大的 NCS 中, 如何准确地对系统进行控制, 快速适应当前系统的变化变得极为重要。MFAC 控制算法在计算控制信号的时仅使用到了系统当前的数据, 并未对系统历史数据加以利用, 学者秦品乐<sup>[5]</sup>提出了使用多新息理论优化无模型控制律, 从而加快了无模型控制方法中控制律的收敛速

收稿日期: 2014-05-09; 修回日期: 2014-06-23。

基金项目: 山西省青年基金(2010021016-3)。

作者简介: 陈晓青(1988-), 女, 福建龙岩人, 硕士生, 主要从事网络控制系统控制算法等方向的研究。

度,达到了对系统快速准确地控制的目的。本文使用 truetype 仿真工具箱,搭建了基于以太网的 NCS 的仿真平台,通过对比 MFAC、PID 和 MI-MFAC3 种控制算法,在该平台上对算法进行了验证本文提出方法的有效性和可行性。

## 1 MFAC 及其改进

### 1.1 MFAC

无模型控制 (model free adaptive control, MFAC)<sup>[6]</sup>理论与方法是指:控制器的设计仅利用受控系统的 I/O 数据,控制器中不包含受控过程数学模型的任何信息的控制理论与方法。其基本思想是利用一个新引入的伪梯度向量和伪阶数的概念,在受控系统轨线附近用一系列的动态线性时变模型一般离散时间非线性系统。

$$y(k) = f(y(k), \dots, y(k - n_y), u(k), \dots, u(k - n_u))$$

其中:  $y(k), u(k)$  分别表示系统的输入与输出,  $n_y, n_u$  分别表示系统阶数。系统在满足一定条件下,可以表示成泛模型:

$$\Delta y(k+1) = \varphi^T(k) \Delta u(k) \quad (1)$$

其中:  $\varphi(k)$  为伪梯度,基于紧格式线性化的 MFAC 算法为:

$$\hat{\varphi}(k) = \hat{\varphi}(k-1) + \frac{\eta \Delta u(k-1)}{\mu + |\Delta u(k-1)|^2} [\Delta y(k) - \hat{\varphi}(k-1) \Delta u(k-1)]$$

若  $\hat{\varphi}(k) < \epsilon (\epsilon > 0)$  或  $|\Delta u(k-1)| \leq \epsilon$ , 则  $\hat{\varphi}(k) = \hat{\varphi}(k-1)$ 。

$$u(k) = u(k-1) + \frac{\hat{\alpha} \varphi(k)}{\lambda + |\hat{\varphi}(k)|^2} [y(k+1)^* - y(k)] \quad (2)$$

从式 (2) 可以看出, MFAC 的计算仅仅使用了当前  $k$  时刻的数据,历史数据并未参与计算,没有充分使用历史数据中隐含的信息。

### 1.2 多新息理论改进 MFAC 控制律

多新息 (multi-innovation, MI) 理论最初是用于系统辨识中对参数的估计,许多辨识算法,如最小二乘类和随机梯度类算法的一个共同特点:都是利用单新息修正技术的单新息辨识方法。

即对于标量系统<sup>[7]</sup>:

$$y(t) = \varphi^T(t) \theta + v(t) \quad (3)$$

其中,  $y(t) \in \mathbf{R}$  为系统的输出,  $\varphi^T(t)$  为系统输入输出数据构成的信息向量,  $\theta$  为待辨识的参数向量,  $v(t)$  为系统噪声。使用最小二乘辨识算法或随机梯度等辨识算法估计  $\theta$  有下列形式:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \mathbf{L}(t) [y(t) - \varphi^T(t) \hat{\theta}(t-1)]$$

其中  $e(t) = y(t) - \varphi^T(t) \hat{\theta}(t-1) \in \mathbf{R}$ , 为标量新息,即单新息。  $\mathbf{L}(t) \in \mathbf{R}^n$  为算法增益向量。顾名思义,多新息算法就是将新息加以推广。为了有效利用了历史新息所包含的信息,因此将新息扩展成一个多新息向量。

对标量系统而言,将算法中的标量新息  $e(t) \in \mathbf{R}$  推广为新息向量,  $E(p, t) = [e(t), e(t-1), \dots, e(t-p+1)]^T \in \mathbf{R}^p, p \geq 1$  为新息长度。

因此,对应将式 (3) 改为式 (4):

$$Y(p, k) = \varphi^T(p, k) \theta + v(p, k) \quad (4)$$

其中:

$$Y(p, k) = [y(k), y(k-1), \dots, y(k-p+1)]^T \in \mathbf{R}^p$$

$$\varphi(p, k) = [\varphi(k), \varphi(k-1), \dots, \varphi(k-p+1)] \in \mathbf{R}^{n \times p}$$

$$v(p, k) = [v(k), v(k-1), \dots, v(k-p+1)]^T \in \mathbf{R}^p$$

参考多信息系统的辨识模型 (4), 可将无模型的泛模型式 (1) 修改为如下形式<sup>[8]</sup>:

$$\Delta Y(p, k+1) = \Phi^T(p, k) \Delta u(k) + \Delta E(p, k) \quad (5)$$

其中:

$$\Delta Y(p, k+1) = [\Delta y(k), \Delta y(k-1), \dots,$$

$$\Delta y(k-p+2)]^T \in \mathbf{R}^p$$

$$\Phi(p, k) = [\varphi(k), \varphi(k-1), \dots, \varphi(k-p+1)] \in \mathbf{R}^{1 \times p}$$

$$\Delta E(p, k) = [\Delta e(k), \Delta e(k-1), \dots, \Delta e(k-p+1)]^T \in \mathbf{R}^p$$

考虑如下带有控制输入变化量平方惩罚项的平方控制误差准则函数

$$J(u(k)) = E[\|Y^*(p, k+1) - Y(p, k+1)\|^2 + \lambda \|u(k) - u(k-1)\|^2] + E[\|Y^*(p, k+1) - Y(p, k) - (Y(p, k+1) - Y(p, k))\|^2 + \lambda \|u(k) - u(k-1)\|^2] + E[\|Y^*(p, k+1) - Y(p, k) - \Delta Y(p, k+1)\|^2 + \lambda \|u(k) - u(k-1)\|^2] \quad (6)$$

其中:  $Y^*(p, k+1)$  表示新息长度为  $p$  的系统的期望输出,  $\lambda$  的大小对控制输入量的变化起到了惩罚作用,适当选取  $\lambda$  的值,可以避免由于控制输入量过大而引起的不稳定。

将式 (5) 代入式 (6), 得

$$J(u(k)) = E[\|Y^*(p, k+1) - Y(p, k) - \Phi^T(p, k) \Delta u - \Delta E(p, k)\|^2 + \lambda \|u(k) - u(k-1)\|^2] \quad (7)$$

求  $J(u(k))$  对  $u(k)$  的极小值, 即

$$\frac{\partial J(u(k))}{\partial u(k)} = 0$$

可以得到

$$E\{-\Phi(p, k) [Y^*(p, k+1) - Y(p, k) - \Phi^T(p, k) \Delta u(k)] + \lambda \Delta u(k)\} = 0$$

利用 Monro-Robbins 随机逼近算法,可以得到式

$$\Delta u(k) = \psi \hat{\Phi}(p, k) (Y^*(p, k+1) - Y(p, k)) \quad (8)$$

令

$$\psi = \frac{\rho}{\lambda + \|\hat{\Phi}(p, k)\|^2}$$

将  $\psi$  代入式 (8) 可以得到就多信息理论的无模型控制律式 (9):

$$u(k) = u(k-1) + \frac{\rho \hat{\Phi}(p, k)}{\lambda + \|\hat{\Phi}(p, k)\|^2} [Y^*(p, k+1) - Y(p, k)] \quad (9)$$

至此,基于多新息无模型控制律的无模型控制方法 (MI-MFAC) 可表示为 (10):

$$\hat{\varphi}(k) = \hat{\varphi}(k-1) + \frac{\eta \Delta u(k-1)}{\mu + \|\Delta u(k-1)\|^2} [\Delta y(k) - \hat{\varphi}(k-1) \Delta u(k-1)] \quad (10)$$

若  $\hat{\varphi}(k) < \epsilon (\epsilon > 0)$  或  $|\Delta u(k-1)| \leq \epsilon$ , 则  $\hat{\varphi}(k) = \hat{\varphi}(k-1)$

$$u(k) = u(k-1) + \frac{\rho \hat{\Phi}(p, k)}{\lambda + \|\hat{\Phi}(p, k)\|^2} [Y^*(p, k+1) - Y(p, k)]$$

其中:

$$\Delta Y(p, k) = [\Delta y(k), \Delta y(k-1), \dots, \Delta y(k-p+1)]^T \in \mathbf{R}^{1 \times p}$$

$$\Phi(p, k) = [\varphi(k), \varphi(k-1), \dots, \varphi(k-p+1)] \in \mathbf{R}^{1 \times p}$$

式中可以看出,当  $p=1$  时,改进的 MFAC 就变成了标准的 MFAC,在式中,可以发现改进的 MFAC 使用了历史数据参与计算,充分利用了历史数据中隐含的可用信息。加快了控

制律的收敛速度。

### 2 仿真结果与分析

网络控制系统的设计如图 1 所示, 但不同于传统控制系统的是, 各个节点是通过网络环境进行传输的, 这个网络可能是局域网也可以是互联网。本文使用 TrueTime 仿真工具箱对基于 MFAC 的网络控制系统进行仿真, 为了加入前面提出的改进 MFAC 算法, 本文需要事先搭建出一个 NCS 的仿真平台, 本仿真实例主要用到 TrueTime Kernel 模块和 TrueTime Network 模块, 其中传感器、控制器以及执行器等网络节点使用计算机模块仿真, 通过定义数据结构、编写代码处理相应功能完成各个节点的任务。这些节点之间的信息使用有线网络模块进行传输。Truetime 模块的使用情况如图 2 所示。

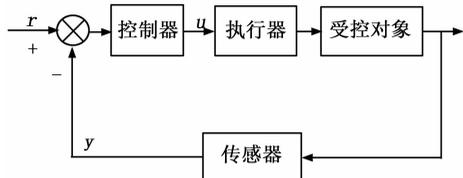


图 1 网络控制系统结构图

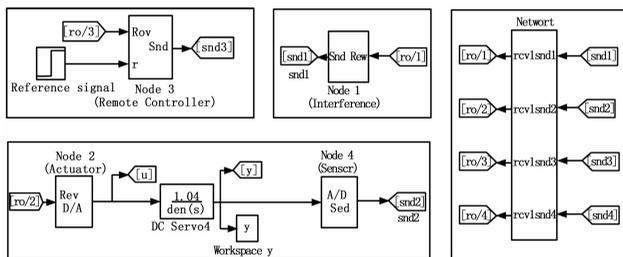


图 2 Truetime 仿真模块组成图

Truetime 模块当中的配置说明:

节点 1 为干扰节点, 通过定义周期性任务 interf\_task, 以概率 BWshare=0.2 给自己发送 100bit 的信息, 模拟网络中其他节点的干扰。其中 BWshare 可以用于模拟网络通信中带宽的占用率, 表示该节点占用信道的概率。

节点 2 为执行器和节点 4 为传感器, 节点 2 定义了事件触发任务, 当收到来自节点 3 (控制器) 发送的控制信号时, 则进行相应的执行操作, 进行 D/A 转换。节点 4 定义了周期性任务对受控系统进行采样, 通过 A/D 之后再发送给控制器。

节点 3 为控制器, 控制器的结构设计如图 3 所示, 在接收到传感器发送过来的采样信号之后, 进入控制信号的计算, 在控制器的初始化文件中定义中断, 当控制器接收到信号时, 进入中断, 并执行控制算法产生控制信号 u。2 号接收通道用于输入参考信号 r, 并进行 AD 转换, 1 号接收通道用于接收从网络发过来的实际输出信号 y, 计算产生的控制信号 u 通过 1 号发送通道给网络, 再由网络转发给执行器。

网络模块的设置: 网络的类型选择 CSMA/CD (Ethernet), 模拟当网络通信环境为以太网时的 NCS。本仿真只是用一个网络模块, 故 network number 设置成 1, 由于节点 1~4 都需要通过网络模块通信, 故通信节点设为 4 个, 数据传输速率设置为 10 Mbits/s, 最小帧长设置为 512 bit (以太网的最小帧长), 丢包率为 0.2%。

#### 2.1 线性系统

考虑如下线性系统仿真对象:

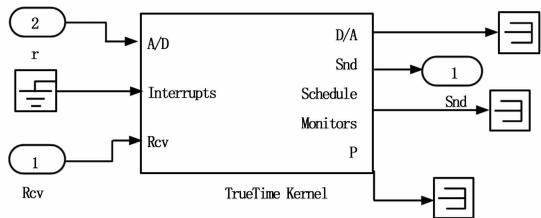


图 3 控制器节点的设计

$$G(s) = \frac{1000}{50s + 225}$$

当采用 PID 控制器时, 参照文献 [9] 各参数的取值如下:

$$K = 0.15; K_i = 2; K_d = 0.025$$

结合公式 (2) 将 MFAC 控制算法应用到 NCS 中, 参考文献 [10] 各参数的取值如下:

$$\epsilon = 10^{-6}; \mu = 1; \eta = 1; \lambda = 70; \rho = 1; \varphi(1) = 0.1;$$

两种控制器的仿真结果如图 4 所示。

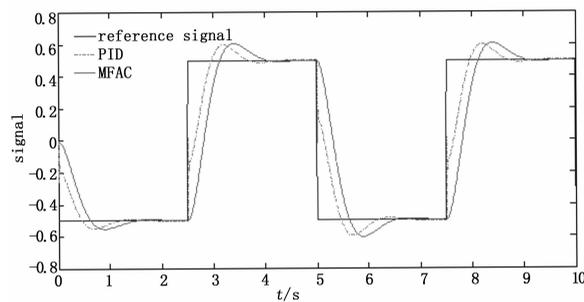


图 4 线性系统的仿真结果

由图 4 可以看出, 在线性系统下, PID 控制器要略优于 MFAC 控制器, 但是 MFAC 控制器仍然能保证网络控制系统的性能, 证实了 MFAC 控制器在网络控制系统中的可行性。由于在线性系统下, PID 的控制效果良好, 故在线性系统下, 不再对 MI-MFAC 进行仿真。

#### 2.2 非线性系统

考虑如下非线性系统仿真对象:

$$G(s) = \frac{1.04}{0.000675s^2 + 0.0225s + 0.1352}$$

当采用 PID 控制器时, 各参数的取值如下:

$$K = 0.01; K_i = 0.25; K_d = 0.0025$$

当采用 MFAC 控制器时, 各参数的取值如下:

$$\epsilon = 10^{-6}; \mu = 1; \eta = 0.1; \lambda = 1; \rho = 0.0225; \varphi(1) = 0.1$$

结合公式 (10) 将改进的 MFAC 控制算法应用到 NCS 中, 其中 p=3, 其余参数的设置可以参照 MFAC 控制算法下的取值, 并对各参数进行微调之后, 得到各参数的取值如下:

$$\epsilon = 10^{-6}; \mu = 1; \eta = 0.01; \lambda = 3; \rho = 0.0225;$$

$$\varphi(1) = 0.1; \varphi(1) = 0.2; y(1) = 0; y(2) = 0.1;$$

$$u(1) = u(2) = 0;$$

3 种控制器的仿真结果如图 5 所示。

由图 5 可以看出, 在非线性的系统下, PID 控制器的效果不是很突出, 调整时间过长, 而 MFAC 控制器具有良好效果, 调整时间相对 PID 控制器要较短, 在非线性的系统下的控制效果良好。当采用多新息改进 k 控制律的 MFAC 控制器时, 上