

文章编号:1674-2869(2014)011-0070-05

# 矩阵实验室的仿真工具TrueTime设计

何 莹,孙 剑

泰州学院计算机科学与技术学院,江苏 泰州 225300

**摘 要:**为了便于研究网络控制系统,使用了基于矩阵实验室的功能强大的 TrueTime(TT)网络仿真工具箱. 它包括网络和计算机两个接口模块,网络模块支持对六种网络的配置;计算机模块具有模拟/数字的转换器,实时内核,网络接口以及外部通道. 研究了在矩阵实验室编译环境下利用 TT 工具箱进行网络控制仿真的方法和流程,主要包括 TT 工具箱的调用、初始化、编译、控制算法 S 函数的调用和网络配置方法等. 最后给出了两个利用 TT 工具箱进行网络控制仿真的实例. 通过 TT 进行网络控制仿真能够较好地再现网络控制系统的特征,具有结果可信程度高、易于实现等特点.

**关键词:**网络控制系统;时延;TrueTime 工具箱

**中图分类号:**TP311

**文献标识码:**A

**doi:**10.3969/j.issn.1674-2869.2014.011.013

## 0 引 言

仿真开始前,要初始化各个内核模块以及网络模块,创建一些计时器、任务、事件和监控器等<sup>[1]</sup>. 初始化代码和仿真时需要执行的代码可写成 C++ 函数,也可以写成矩阵实验室(Matrix Laboratory,以下简称:Matlab)中的.m 文件,前者的效率较高,后者更方便.

## 1 TrueTime 的代码书写

通过一个比例控制器(PI-controller)的实现来说明如何书写 Matlab 或 C++ 代码.

### 1.1 书写 Matlab 代码

比例控制的 Matlab 代码实现如下:

```
Function[exectime,data]=pcontroller(segment,data)
Switch segment,
    Case 1,
        r=ttAnalogIn(1);
        y=ttAnalogIn(2);
        data.u=data.K*(r-y);
        exectime=0.002;
```

Case 2,

```
    ttAnalogOut(1,data,u);
    exectime=-1;%finished
```

end

变量 segment 用来决定执行哪一部分,data 是用户定义的数据结构,该数据结构是在执行 tt-

CreateTask 或 ttCreatePeriodicTask 时创建的,在这个函数里被更新并返回. ttAnalogIn 表示读,ttAnalogOut 表示写. 第一段代码的执行时间是 2 s,这表示该任务的输入到输出的延时至少为 2 s. 第二段代码返回一个负的执行时间,用以表示代码执行已经结束,即没有其他的代码段需要执行了.

### 1.2 书写 C++ 代码

比例控制的 C++ 代码实现如下所示,其中数据结构 Task\_Data 已知,包含了控制信号  $u$  和控制器放大倍数  $K$ .

```
Double Pcontroller(int segment,void * data){
    Task_Data * d=(Task_Data *) data;
    Swich(segment) {
        Case 1:
            double r=ttAnalogIn(1);
            double y=ttAnalogIn(2);
            d->u=d->K*(r-y);
            return 0.002;
        Case 2:
            ttAnalogOut(1,d->u);
            return FINISHED;//end of execution } }
```

### 1.3 调用仿真模块框图

不论是通过 C++ 实现还是通过 Matlab 实现,都可以在代码里调用仿真模块框图,这是实现控制器的一个简便方法,其代码实现如下:

```
Function[exectime,data]=Picode(segment,data)
```

收稿日期:2014-05-27

作者简介:何 莹(1981-),女,江苏泰州人,讲师,硕士. 研究方向:网络技术与信息安全.

```
Switch (segment),
    Case 1,
        inp(1)=ttAnalogIn(1);
        inp(2)=ttAnalogIn(2);
        outp=ttCallBlockSystem(2,inp,'PI_Controller');
        data.u=outp(1);
        exectime=outp(2);
    Case 2,
        ttAnalogOut(1,data,u);
        exectime= -1;%finished
end
```

2 初始化 TrueTime

TrueTime 内核模块初始化有<sup>[2]</sup>:定义调度策略;指定输入和输出的个数;创建任务、事件、监测器等. 这些都是由初始化脚本函数来完成的. 它的内核模块带有一个参数,不论是在 Matlab 还是在 C++ 里,此参数都是初始化脚本的文件名,例如下面例子中的参数就是:example\_init.

一个 TT 仿真在初始化时需要的最少代码如下所示,通过 ttInitKernel 函数定义输入、输出个数和调度政策;通过如下所示的函数创建一个周期任务,此任务调用比例控制 Matlab 代码实现程序中所给的 Pcontroller 函数.

```
Function example_init
TtInitKernel(2,1,'PrioFP');
data.u=0;
data.K=2;
offset=0;
period=0.005;
prio=5;
TtCreatePeriodicTask('ctrl',offset,period,prio,
'Pcontroll',data);
```

初始化脚本的 C++ 实现下面程序所示,文件 ttkernel.cpp 包含了用以实现 TrueTime 内核的仿真回调函数,这个初始化脚本实际上是一个完整的 Matlab S-函数. 变量 filename 是指源文件的 名字,即如果源文件是 example\_init.cpp,那么就把 S\_FUNCTION\_NAME 定义为 example\_init.

```
#define S_FUNCTION_NAME filename
#include "ttkernel.cpp"
//insert your code functions here

void init()
//perform the initialization }
```

```
void cleanup() {
//free dynamic memory allocated in this script
}
```

Init()函数完成所有的初始化工作,由它动态分配的内存都可以通过 cleanup()函数收回<sup>[3]</sup>. TT 初始化脚本的 Matlab 实现的 C++ 版本如下所示.

```
#define S_FUNCTION_NAME example_init
#include "ttkernel.cpp"
#include "Pcontroller.cpp" //P-controller
code function
class Task_Data { public: double u; double K; };
Task_Data * data;//pointer to local memory
for the task
void init() { ttInitKernel(2,1,FP);
data=new Task_Data; data->u=0.0;
data->K=2.0; double offset=0.0;
double period=0.005; double prio=5.0;
ttCreatePeriodicTask("ctrl",offset,period,
prio,Pcontroller,data); }
void cleanup() {
delete data; }
```

3 TrueTime 的编译

在 Matlab 中编译只要执行 make\_truetypeime.m 就行了,这个脚本会编译 TT 的内核以及网络 S-函数,还有 MEX 文件. 对于 C++ 就不同了,初始化脚本(比如 example\_init.cpp)需要先编译产生一个 Matlab mex 文件,一般采用下面这个命令:>>ttmex example\_init.cpp 每次改变代码函数或者初始化脚本时,都需要重新编译一次. 注意:ttmex 命令与普通的 mex 命令具有相同的功能,只是前者会自动找到内核文件的路径. 在 Matlab 情况下,即使对代码函数或者初始化脚本作了改动,但是仿真结果却没有任何变化,此时用下指令>>clean function 来迫使 Matlab 在仿真前重新加载所有的函数.

4 TrueTime 中的网络设置

TrueTime 的网络模块模拟了在一个局域网中的介质访问和分组传输过程<sup>[4]</sup>. 当一个节点需要发送消息(使用 ttSendMsg)时,一个触发信号会被送到相应的输入通道的网络模块中;当这个信息的仿真传输过程完成时,网络模块就会通过相应的输出通道发送一个新的触发信号到目的节点;最后这个被传输的信息放在了目的计算机的

缓冲区里. 此消息包含: 用户数据(例如控制信号以及测量信号)、发送以及接收节点的信息、信息的长度、优先级以及最终期限.

TrueTime 支持 6 种网络: CSMA/CD(e. g. Ethernet)、CSMA/AMP(e. g. CAN)、Round Robin(e. g. 令牌总线网)、FDMA、TDMA(e. g. TTP)和分组 Ethernet. TT 中忽略了传播时延, 因为在 LAN 中这类时延很小. 可以通过 block-mask 对话框来设置网络模块<sup>[5]</sup>. 以 Ethernet 为例来说明如何设置这个对话框. 首先介绍一些涉及到的网络参数: Network number, 表示网络模块的数量, 有上限; Number of nodes, 表示与网络连接的节点个数, 这个数字将决定 Snd、Rcv 的大小以及输入、输出量的个数; Data rate (bits/s) 表示网络速率; Minimum frame size (bytes), 对于不同的协议, 这个数值是不一样的, 在 Ethernet 中是 64 bytes, 它包括 14-byte 的头信息以及 4-byte 的 CRC; Pre-processing delay(s), 是指数据在发送端由于网络接口而造成的延迟; Post-processing delay(s): 是指数据在接收端由于网络接口而造成的延迟; 丢包率(0-1), 是指数据在传输过程中可能丢失的概率, 丢失的数据会占用网络带宽, 但永远到不了目的节点. Ethernet 采用的 MAC 机制是 CSMA/CD, 如果网络忙, 发送端会等待一段时间间隔后重发. 一旦出现数据发送冲突, 发送端将会等待一段时间  $t_{\text{backoff}}$ , 定义如下:  $t_{\text{backoff}} = \text{frame size}_{\min} / \text{data rate} \times R$ , 其中  $R = \text{rand}(0, 2^K - 1)$ ,  $K$  是发生冲突的节点数. 等待一段时间后, 发送端会尝试复发. 用一个例子说明如下, 有两个节点正在等待第三个节点的传输结束, 它们之间发生冲突的概率为  $1, 1/2(K=1), 1/4(K=2), \dots$ . 实现网络模块的 S-函数在 \$DIR/true-time/kernel 下面, 这个文件只要编译一次就可以了, 其编译指令为:  $\gg \text{mex ttnetwork. cpp}$ .

## 5 TrueTime 应用示例

笔者主要给出 2 个应用示例<sup>[6]</sup>, 例 1 是对直流伺服电机(DC-servo)实现简单的 PID 控制, 例 2 则通过网络实现对直流伺服电机的控制. 直流伺服电机的传递函数可以描述为:  $G(s) = \frac{1\,000}{s(s+1)}$ , PID 控制器的实现为:  $P(k) = K \cdot (r(k) - y(k))$ ,  $J(k+1) = I(k) + \frac{Kh}{T_d}(r(k) - y(k))$ ,  $D(k) = a_d D(k-1) + b_d(y(k-1) - y(k))$ ,  $u(k) = P(k) + I(k) + D(k)$ , 其中  $a_d = \frac{T_d}{Nh + T_d}$ ,  $\omega_c = 20 \text{ rad/s}$ . 选

择合适的控制器参数使闭环带宽为  $\omega_c = 20 \text{ rad/s}$ , 衰减率为  $\xi = 0.7$ .

[例 1] DC-servo 的实时控制 通过这个例子, 可以对 TrueTime 的仿真环境有个基本认识. 这个过程的控制任务是在一个 TT 内核模块里实现的, 既可以使用标准的 PID 实现方法, 也可以调用 Simulink 模块框图来计算每个采样时刻的控制信号. 下面仅给出几个主要的函数代码, 其他代码从略.

用 Matlab 代码实现控制器任务:

```
function[exectime,data]=picode(seg,data)
switch seg,
case 1,
    r=ttAnalogIn(data,rChan);
    y=ttAnalogIn(data,yChan);
    data=pidcalc(data,r,y);
    exectime=0.002;
Case 2,
    ttAnalogOut(data,uChan,data.u);
    exectime=-1;
end
```

初始化脚本如下:

```
Function singleservo_init
ttInitKernel(2,1,'prioFP');% nbrOfInputus,
nbrOfOutputs,FP
data.K=0.96; data.Ti=0.12;
data.Td=0.049; data.N=10;
data.h=0.006; data.u=0;
data.Iold=0; data.Dold=0;
data.yold=0; data.rChan=1;
data.yChan=2; data.uChan=1;
ttCreatePeriodicTask('pid_task',0.0,0.006,
2,'picode',data);
```

通过仿真模型, 可以很好地分析 DC-servo 的 PID 控制效果. 比如, 通过改变第一段代码的执行时间来仿真不同输入-输出延时的效果; 通过改变采样周期来研究采样周期对控制性能的影响.

[例 2] DC-servo 的网络控制 在这个实验里, 一共有 4 个计算机节点, 每一个都代表一个 TT 内核模块, 传感器采用时间驱动, 采样周期设为  $h$ , 通过网络将采样后的数据传输给控制节点, 计算控制信号是控制器的任务, 并将其传输给执行器节点, 执行器采用事件驱动方式, 所以一旦有控制信号到达, 便立即响应. 下面分别给出控制器、传感器和执行器的实现代码, 其他代码从略.

Matlab 代码实现传感器:

```

function[exectime,data]=sescode(seg,data)
switch seg,
    case 1,
        data.y=ttAnalogIn(1);
        exectime=0.0005;
    case 2,
        ttSendMsg(4,data.y,10);% Send message
to mode 4 (controller)
        exectime=0.0004;
    case 3,
        exectime= -1;%finished
end
Matlab 代码实现控制器:
function[exectime,data]=ctrlcode(seg,data);
switch seg,
    case 1,
        r=ttAnalogIn(1);%Read reference value;
        y=ttGetMsg;%Obtain sensor value;
        D=data.ad * data._Dold+data.bd * (data.
yold-y);
        P=data.K * (r-y);
        data.u=P+D; data.yold=y; data.Dold=
D; exectime=0.0005;
    case 2,
        ttSendMsg(2,data.u,10);%Send 10 bytes
to node2 (actuator)
        exectime= -1;%finished
end
Matlab 代码实现执行器:
function[exectime,data]=actcode(seg,data);
switch seg,
    case 1,
        data.u=ttGetMsg; exectime=0.0005;
    case 2,
        ttAnalogOut(1,data.u); exectime= -1;
%finished
end

```

## 6 结 语

应用 TrueTime 做网络控制体系的仿真能够较好的再现系统的特征,例如可以设置节点的驱动方式、采样周期,可以设定网络时延的特性,可以设置所采用的网络类型等等.因为 TrueTime 基于 Matlab-Simulink,所以控制算法的实现很方

便,既可以使用标准的传递函数方法,也可以调用系统自带的内核模块框图.

## 致 谢

感谢泰州学院提供的试验平台!

## 参考文献:

- [1] 何坚强,张焕春.基于 TrueTime 工具箱的网络控制系统仿真研究[J].微计算机信息,2004,20(1):33-34,98.  
HE Jian-qiang,ZHANG Huan-chun. Simulation research of networked control system based on True Time[J]. Control and Automation,2004,20(1):33-34,98. (in Chinese)
- [2] 于之训,蒋平,陈辉堂,等.具有传输延迟的网络控制系统中状态观测器的设计[J].信息与控制,2000,29(2):125-130.  
YU Zhi-xun,JIANG Ping,CHEN Hui-tang, et al. Design of statusobserver for network control system with transfer delay [J]. Information and Control, 2000,29(2): 125-130. (in Chinese)
- [3] 任长清,吴平东,王晓峰,等.基于互联网的液压远程控制系統延时预测算法研究[J].北京理工大学学报,2002,22(1):85-89.  
REN Chang-qing,WU Ping-dong,WANG Xiao-feng, et al. A study on the forecast arithmetic of hydraulic telecontrol system based on internet [J]. Journal of Beijing Institute of Technology,2002,22(1):85-89. (in Chinese)
- [4] 顾洪军,张佐,吴秋峰.网络控制系统中周期性通信的实时性充分条件[J].测控技术,2001,20(6):1-4.  
GU Hong-jun,ZHANG Zuo,WU Qiu-feng. Real-time sufficient condition of cyclic communication in networked control system [J]. Measurement and Control Technology,2001,20(6):1-4. (in Chinese)
- [5] 王晓峰,吴平东,任长清,等.基于 TCP/IP 网络的远程伺服控制系统中的动态模糊控制器[J].计算机应用,2002,22(11):83-85.
- [6] 于之训,陈辉堂,王月娟.基于  $H_{\infty}$  和  $\mu$  综合的闭环网络控制系统的设计[J].同济大学学报:自然科学版,2001,29(3):307-311.  
YU Zhi-xun,CHEN Hui-tang,WANG Yue-juan. Design of closed loop network control system based on  $H_{\infty}$  and- $\mu$  synthesis[J]. Journal of Tongji University: Natural Science,2001,29(3):307-311. (in Chinese)

## Design of matrix laboratory-based TrueTime toolbox

*HE Ying, SUN Jian*

College of Computer Science and Technology, Taizhou University, Taizhou 225300, China

**Abstract:** To facilitate the study of network control systems, a matrix laboratory-based TrueTime (TT) toolbox was adopted, which is a powerful network simulation toolbox. TT includes network block and computer blocks, the network module supporting six kinds of network configuration, and the computer module consisting of flexible real-time kernel, analog/digital and digital/analog converter, network interface and an external channel. We studied the method and process of network control simulation using TT toolbox in matrix laboratory-based compiler environment, mainly including TT toolbox call, initialization, compilation, control algorithm of S function call and network configuration method etc.. Finally we provided two simulation examples of network control using TT toolbox. The results show that the characteristics of network control system are simulated by TT toolbox, which has high reliability and easy realization.

**Key words:** network control systems; time-delay; TrueTime toolbox

本文编辑:苗 变