

上海交通大学

网络系统与控制

课程报告

项目名称：网络控制系统的仿真与稳定性分析

姓 名：陈思哲 张沛东 谢敬鱼

班 级：F1603203

学 号：516021910038 516021910178 516021910125

电子邮箱：729020210@qq.com 2744817421@qq.com m13262935410@163.com

2018-2019 学年第 2 学期

2019 年 6 月 4 日

实验目的

了解网络控制系统的组成和运行以及控制原理

熟悉利用 MATLAB 和 TrueTime 进行网络控制系统的仿真的方法

研究丢包率对无线网络传输的网络控制系统稳定性的影响

为系统设计控制器是系统稳定

题目描述

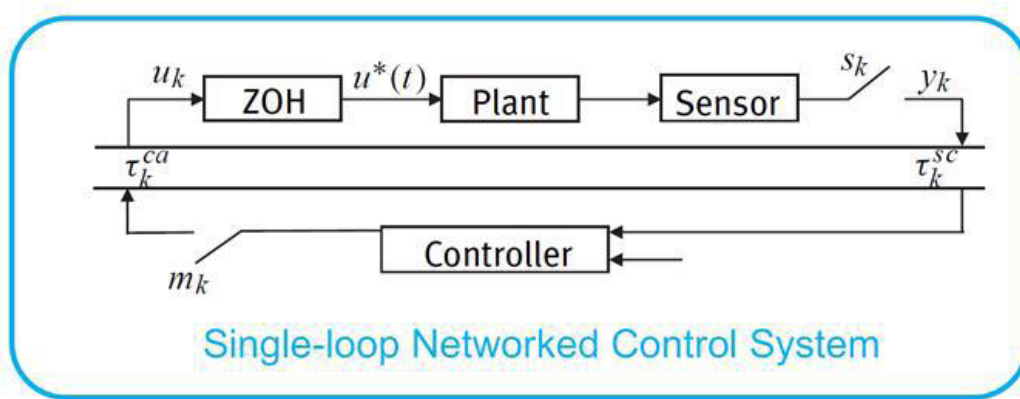


图 2.1 实验系统

考虑如图 2.1 所示的网络化控制系统。 y_k 和 u_k 都通过无线网络输出。 y_k 在传输过程中存在丢包，存在随机丢包，丢包过程是一个 Bernoulli 过程，记为 s_k ： $s_k = 0$ 表示没有丢包； $s_k = 1$ 表示发生了丢包。 u_k 在传输过程中不存在丢包。

已知系统的状态方程为：

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases}$$

其中， $A = \begin{bmatrix} -0.2656 & -2.2023 \\ -1.1878 & 0.9863 \end{bmatrix}$ ， $B = \begin{bmatrix} -0.5186 \\ 0.3274 \end{bmatrix}$ ， $C = [0.2341 \ 0.0215]$ 。

$$\text{定义 } y_c = \begin{cases} y(k), & s_k = 0 \\ y(k-1), & s_k = 1 \end{cases}$$

为以上网络化控制系统，设计动态输出反馈控制器

$$\begin{cases} x_c(k+1) = A_c x(k) + B_c y_c(k) \\ u(k) = C_c x_c(k) \end{cases}$$

- (1) 在不同丢包率 p 下，使用 LMI 技术分析分别求出两种情形下的动态输出反馈控制器，使得闭环系统均方稳定；
- (2) 利用 Truetime 仿真工具，分别给出两种丢包率条件下的系统状态变量 $x(k)$ 、输出变量 $y(k)$ 、动态输出反馈控制器状态 $x_c(k)$ 、系统的控制输入 $u(k)$ 随时间的变化曲线；给出在随机丢包概率条件下的 $y_c(k)$ 。

仿真平台简介

TureTime 具有以下几个用途: 1.探索时间不确定性对于控制系统的影响 2.设计能动态调整控制器的补偿机制 3.为动态调度提供灵活的实验手段 4.仿真事件驱动的系统。

Turetime 仿真软件主要包括两个基本模块：内核模块（TrueTime Kernel）和网络模块（TrueTime Network），如图 3.1 所示。Truetime2.0 工具箱主要包括六大模块：TrueTime Kernel (内核模块)、TrueTime Network (网络模块)、TrueTime Wireless Network (无线网络模块)、TrueTime Battery（电池模块）、TrueTimeSend（发送模块）、TrueTime Receive（接受模块）。

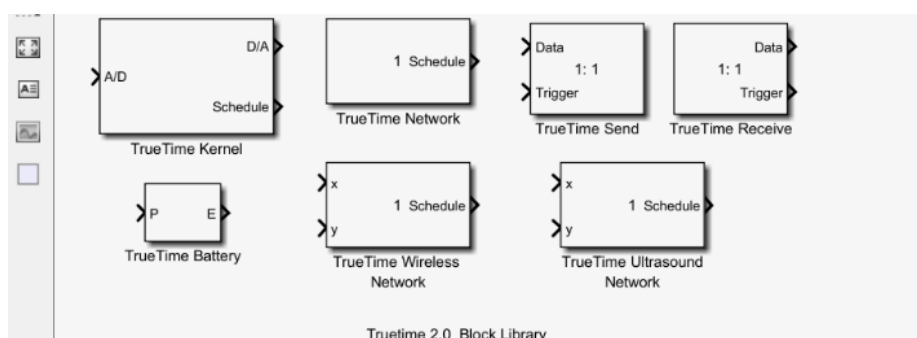


图 3.1 TrueTime 基本模块介绍

仿真平台构建

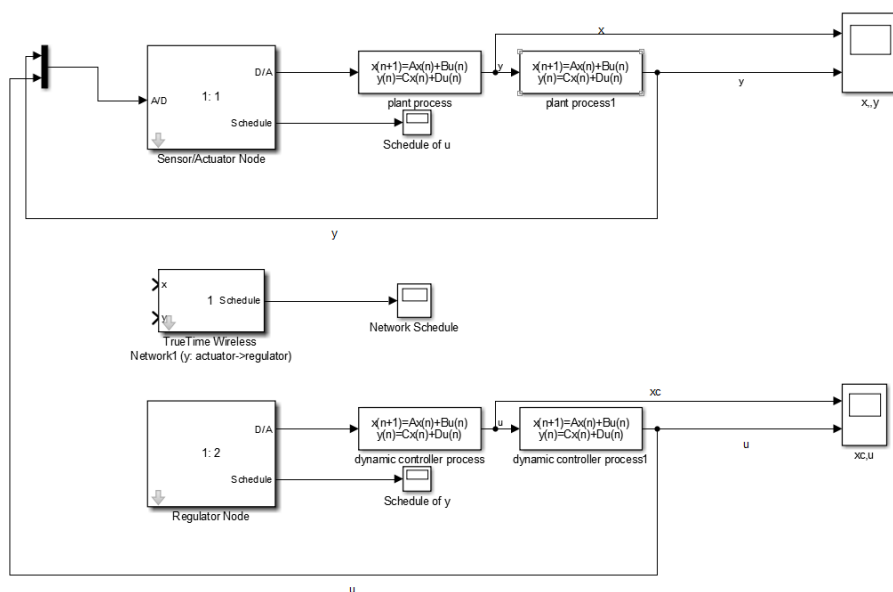


图 4.1 系统框图

仿真网络中使用了两个 Truetime Kernal，分别命名为 Regulator Node 和 Sensor/Actuator Node。由于题目给定的条件是 y_k 丢包而 u_k 不丢包，所以网络图中仅使用一个无线节点，用于 Regulator Node 向 Sensor/Actuator Node 发送 y 数据。 u 通过有线从 Regulator Node 传送至 Sensor/Actuator Node。

Regulator Node 以事件驱动方式向控制器状态空间对象输送 y ，一旦有新的 y 通过网络到达，就立即将 y 输送到状态空间对象当中。Sensor/Actuator Node 以事件驱动方式向被控对象状态空间输送 u 。Sensor/Actuator Node 以时钟驱动方式检测被控对象状态空间输出 y 的状态并通过无线网络发送给 Regulator Node，采样周期为 0.01s。

由于需要检测状态空间的状态量 x ，所以采取了两个状态空间模块串联的形式。第一个状态模块输出为等效状态空间的状态量 x ，第二个状态空间模块输出为等效状态空间的输出量 y 。两个状态空间模块的参数定义为： $A_1 = A, B_1 = B, C_1 = I, D_1 = 0$ ； $A_2 = 0, B_2 = 0, C_2 = 0, D_2 = C$ 。

理论推导控制器设计

定义增广的状态向量：

$$\bar{x}(k) = [x^T(k) \ y_c^T(k-1)]^T$$

假设不会发生连续丢包，即 k 时刻丢包的话， $k-1$ 时刻不会丢包，用公式表达为 $y_c(k-1) = Cx(k-1)$, $s_k = 1$ 。那么有：

$$y_c(k) = \begin{cases} y(k) = Cx(k), & s_k = 0 \\ y(k-1) = y_c(k-1), & s_k = 1 \end{cases}$$

整理可得：

$$y_c(k) = [\theta(k)C \ (1-\theta(k))I] \begin{bmatrix} x(k) \\ y_c^T(k-1) \end{bmatrix} = \bar{C}_{\theta(k)} \bar{x}(k) \quad (1)$$

(1)式再结合系统状态方程 $x(k+1) = Ax(k) + Bu(k)$ 和不难得到新的系统状态方程：

$$\bar{x}(k+1) = \begin{bmatrix} x(k+1) \\ y_c^T(k) \end{bmatrix} = \begin{bmatrix} A & 0 \\ \theta(k)C & (1-\theta(k))I \end{bmatrix} \bar{x}(k) + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) \quad (2)$$

将 $\bar{x}(k)$ 和 $u(k)$ 前面的系数矩阵分别记为 $\bar{A}_{\theta(k)}$ 和 \bar{B} ，它们的具体形式如下：

$$\bar{A}_{\theta(k)} = \begin{bmatrix} A & 0 \\ \theta(k)C & (1-\theta(k))I \end{bmatrix} \quad \bar{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

注意 $\bar{A}_{\theta(k)}$ 是一个与随机事件 $\theta(k)$ 的取值相关的矩阵。在本题目中， $\theta(k)$ 有两个取值，即 0 或 1。因此 $\bar{A}_{\theta(k)}$ 也有两个取值， $\bar{A}_0 = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}$, $\bar{A}_1 = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}$ 。在后文的描述中，将 $\bar{A}_{\theta(k)}$ 和 $\bar{C}_{\theta(k)}$ 简记为 \bar{A}_i 和 \bar{C}_i 。

假设控制器的状态空间系数与 $\theta(k)$ 有关，即 $\begin{cases} x_c(k+1) = A_{ci}x_c(k) + B_{ci}u(k) \\ u(k) = C_{ci}x_c(k) \end{cases}$ ，

结合(1)(2)式子，将状态向量再次增广得到：

$$\xi(k+1) = \begin{bmatrix} \bar{x}(k+1) \\ x_c(k+1) \end{bmatrix} = \begin{bmatrix} \bar{A}_i & \bar{B} \\ B_{ci}\bar{C}_i & A_{ci} \end{bmatrix} \begin{bmatrix} \bar{x}(k) \\ x_c(k) \end{bmatrix} \quad (3)$$

将 $\begin{bmatrix} \bar{A}_i & \bar{B} \\ B_c \bar{C}_i & A_c \end{bmatrix}$ 记为 $A_{cl,i}$ 。 $A_{cl,1} = \begin{bmatrix} \bar{A}_1 & \bar{B} \\ B_c \bar{C}_1 & A_c \end{bmatrix}$, $A_{cl,2} = \begin{bmatrix} \bar{A}_2 & \bar{B} \\ B_c \bar{C}_2 & A_c \end{bmatrix}$ 。(3)式刻画

了一个 Markov 跳变系统。我们需要做的是设计动态输出反馈控制器

$$\begin{cases} x_c(k+1) = A_{ci}x(k) + B_{ci}y_c(k) \\ u(k) = C_{ci}x_c(k) \end{cases}$$

使得(3)描述的系统能够镇定。

定理一：马尔可夫跳变系统是均方稳定 (MSS) 的, 当且仅当能找到矩阵 $G > 0$,

使得 $G - \sum_{j=0}^1 p_{cl,j} A_{cl,j}^T G A_{cl,j} > 0$ 。

在本题目中, (3)式所描述的系统均方稳定等价于: 存在正定矩阵 G , 使得 $G - p A_{cl,1}^T G A_{cl,1} - (1-p) A_{cl,0}^T G A_{cl,0} > 0$ 。

将 $G - p A_{cl,1}^T G A_{cl,1} - (1-p) A_{cl,0}^T G A_{cl,0} > 0$ 转化为矩阵形式,

$$\begin{bmatrix} \sqrt{1-p} A_{cl,0}^T & \sqrt{p} A_{cl,1}^T \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & G \end{bmatrix} \begin{bmatrix} \sqrt{1-p} A_{cl,0} \\ \sqrt{p} A_{cl,1} \end{bmatrix} - G < 0 \quad (4)$$

引入 Schur 补引理:

定理二： $A^T P A - Q < 0$ 与矩阵不等式 $\begin{bmatrix} -P^{-1} & A \\ A^T & -Q \end{bmatrix} < 0$ 等价。

利用 Schur 补引理将(4)式转化为如下形式:

$$\begin{bmatrix} -G & * & * \\ \sqrt{p} A_{cl,0} & -G^{-1} & * \\ \sqrt{1-p} A_{cl,1} & 0 & -G^{-1} \end{bmatrix} < 0 \quad (5)$$

*代表矩阵的对称项。将(5)式子分别左乘、右乘 $\begin{bmatrix} G^{-1} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$, 令 $Z = G^{-1}$, 得到

(6)式:

$$\begin{bmatrix} Z & * & * \\ \sqrt{p} A_{cl,0} Z & Z & * \\ \sqrt{1-p} A_{cl,1} Z & 0 & Z \end{bmatrix} < 0 \quad (6)$$

为了能利用 matlab LMI toolbox 直接求解, 需对(6)式进一步转化, 引入定理:

定理三： 存在 $Z > 0$, A_{ci} , B_{ci} , 和 C_{ci} 使得(6)成立等价于存在 $Y = Y^T$, $X = X^T$, L_i , F_i 和 W_i 使得下面的矩阵不等式成立

$$\begin{bmatrix} \begin{bmatrix} Y & I \\ I & X \end{bmatrix} & & * & * \\ \sqrt{p} \begin{bmatrix} Y\bar{A}_0 + L_0\bar{C}_0 & W_0 \\ \bar{A}_0 & \bar{A}_0X + \bar{B}F_0 \end{bmatrix} & \begin{bmatrix} Y & I \\ I & X \end{bmatrix} & & * \\ \sqrt{1-p} \begin{bmatrix} Y\bar{A}_1 + L_1\bar{C}_1 & W_1 \\ \bar{A}_1 & \bar{A}_1X + \bar{B}F_1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} Y & I \\ I & X \end{bmatrix} \end{bmatrix} > 0 \quad (7)$$

这里仅给出由式(7)求解 A_{ci} , B_{ci} , 和 C_{ci} 的方程：

$$B_{ci} = Y^{-1}L_i \quad (8)$$

$$C_{ci} = F_i(Y^{-1} - X)^{-1} \quad (9)$$

$$A_{ci} = -Y^{-1}[Y\bar{A}_iX + Y\bar{B}_iF_i + L_i\bar{C}_iX - W_i](Y^{-1} - X)^{-1} \quad (10)$$

这样我们就得到了本题目要求的控制器的设计方法：在 matlab LMI toolbox 中求解(7)中给出的线性矩阵不等式，再利用(8)(9)(10)给出的关系求解控制器参数 A_{ci} , B_{ci} , 和 C_{ci} 。

线性矩阵不等式求解

在本作业求解的过程中我们熟悉了 Matlab 自带的 LMI 工具箱接口的使用，进一步加深了线性矩阵不等式的概念的理解。我们主要使用了以下函数：

setlmis([])：初始化一个 LMI 系统

lmivar(type,struct)：设定矩阵变量

lmiterm(termID,A,B,flag)：设定矩阵的项

getlmis：获得 LMI 系统的内部表示

feasp(lmisys,options,target)：调用 feasp 求解器，查看有没有可行解

dec2mat(lmi,feas,Q)：把决策变量转化为矩阵形式

对 LMI

$$\begin{bmatrix} \begin{bmatrix} Y & I \\ I & X \end{bmatrix} & * & * \\ \sqrt{p} \begin{bmatrix} Y\bar{A}_0 + L_0\bar{C}_0 & W_0 \\ \bar{A}_0 & \bar{A}_0X + \bar{B}F_0 \end{bmatrix} & \begin{bmatrix} Y & I \\ I & X \end{bmatrix} & * \\ \sqrt{1-p} \begin{bmatrix} Y\bar{A}_1 + L_1\bar{C}_1 & W_1 \\ \bar{A}_1 & \bar{A}_1X + \bar{B}F_1 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} Y & I \\ I & X \end{bmatrix} \end{bmatrix} > 0$$

获得动态输出反馈控制器的参数如下：

丢包概率为 0.12 时，

$$A_{c0} = \begin{bmatrix} -0.2211 & -0.0640 & 0.0307 \\ -1.2153 & -0.3525 & -0.0041 \\ 0.0000 & 0.0000 & -0.0629 \end{bmatrix} B_{c0} = \begin{bmatrix} -0.0319 \\ -0.0032 \\ 1.0628 \end{bmatrix} C_{c0} = \begin{bmatrix} 0.0868 \\ 4.1218 \\ 0.0025 \end{bmatrix}^T$$

$$A_{c1} = \begin{bmatrix} -0.8956 & -0.1034 & 0.0027 \\ 0.4277 & -0.2157 & 0.0017 \\ 0.0020 & 0.0000 & 0.0000 \end{bmatrix} B_{c1} = \begin{bmatrix} -2.8526 \\ 7.0021 \\ -0.9916 \end{bmatrix} C_{c1} = \begin{bmatrix} 0.0732 \\ 4.1653 \\ 0.0051 \end{bmatrix}$$

丢包概率为 0.05 时，

$$A_{c0} = \begin{bmatrix} -0.2154 & -0.0632 & 0.0307 \\ -1.2188 & -0.3531 & -0.0023 \\ 0.0000 & 0.0000 & 0.0716 \end{bmatrix} B_{c0} = \begin{bmatrix} -0.0264 \\ -0.0013 \\ 1.0714 \end{bmatrix} C_{c0} = \begin{bmatrix} 0.0977 \\ 4.1234 \\ 0.0033 \end{bmatrix}$$

$$A_{c1} = \begin{bmatrix} -0.9711 & -0.1125 & 0.0023 \\ 0.4645 & -0.2109 & 0.0014 \\ 0.0025 & 0.0000 & 0.0000 \end{bmatrix} B_{c1} = \begin{bmatrix} -3.2018 \\ 7.1760 \\ 0.9891 \end{bmatrix} C_{c1} = \begin{bmatrix} 0.0851 \\ 4.1623 \\ 0.0044 \end{bmatrix}$$

实验结果讨论

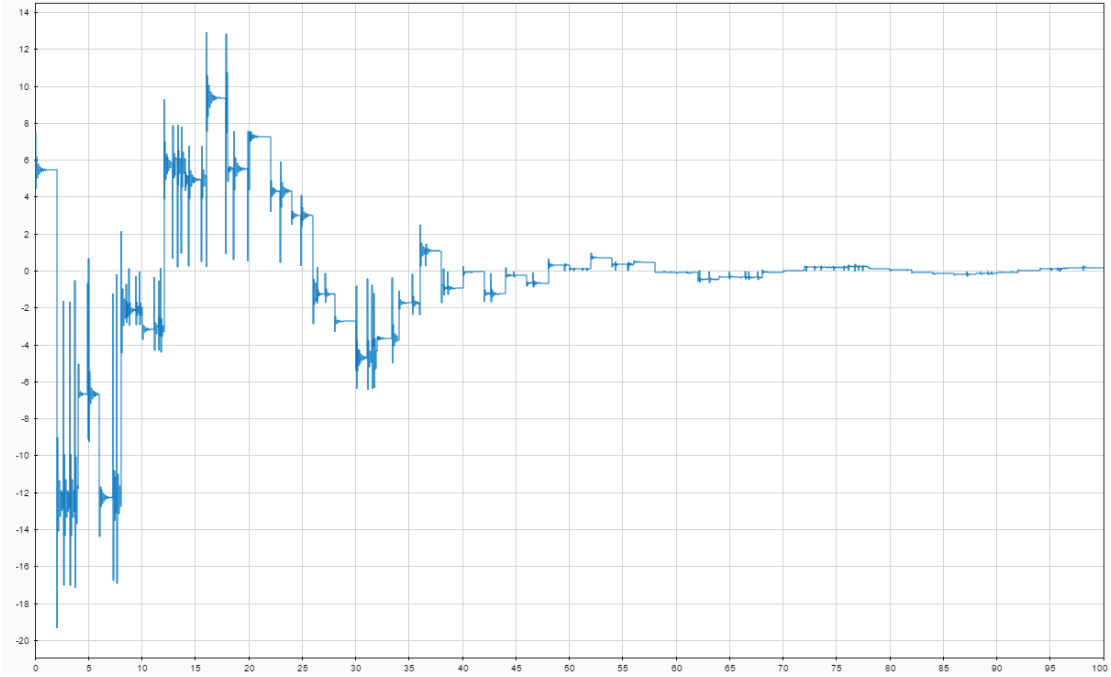


图 7.1 丢包率为 0.05 时的 u_k 曲线

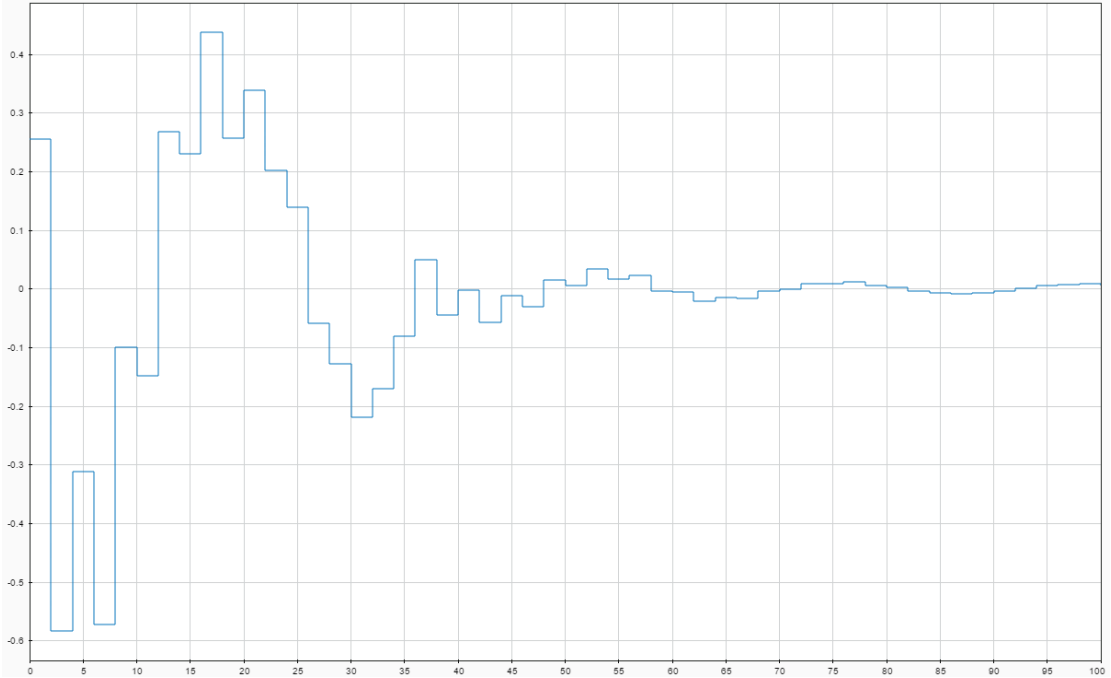


图 7.2 丢包率为 0.05 时的 y_k 曲线

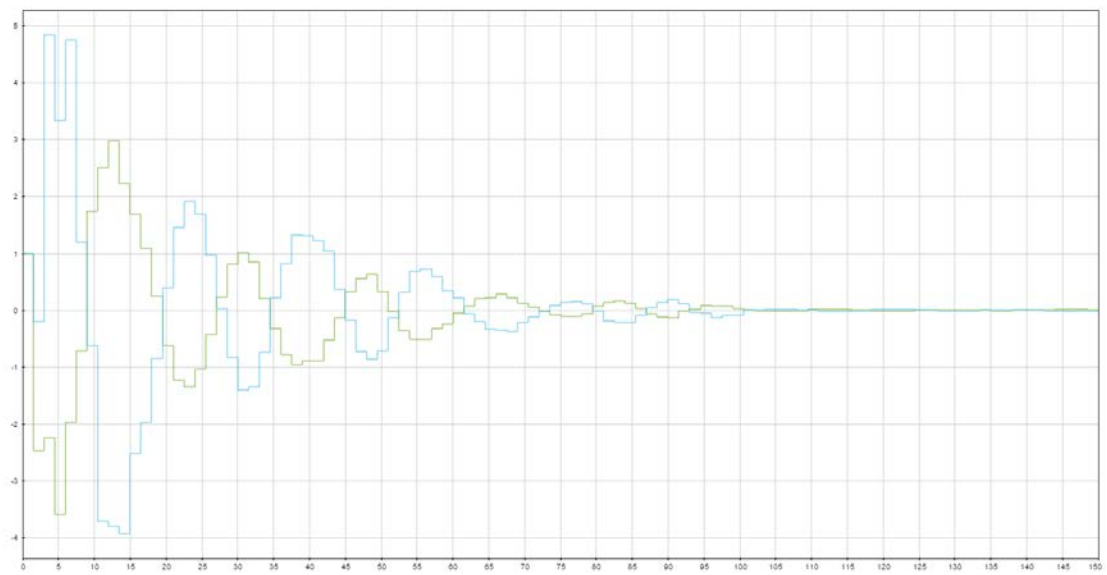


图 7.3 丢包率为 0.05 时的 x 曲线

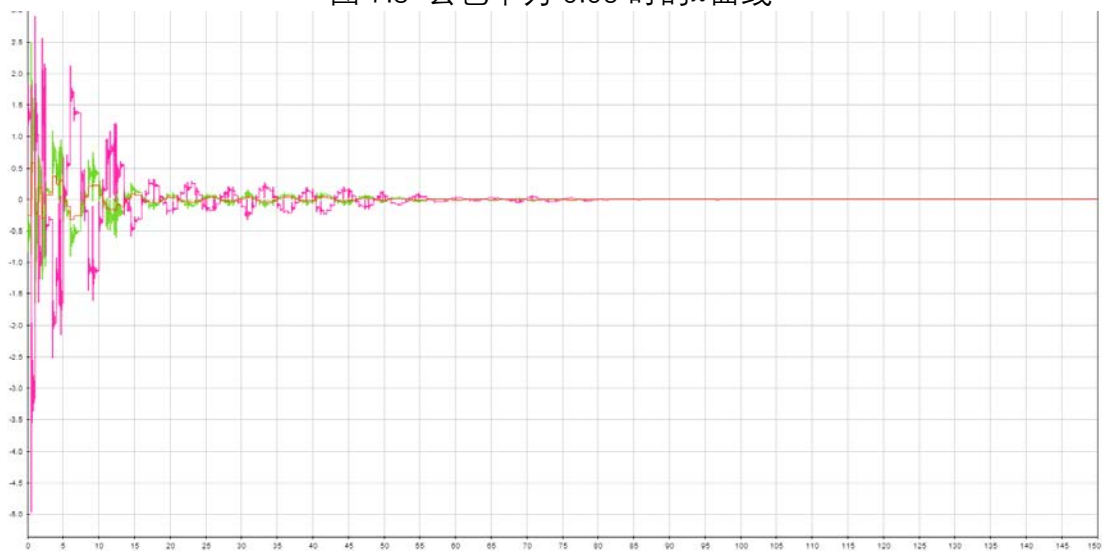


图 7.4 丢包率为 0.05 时的 x_c 曲线

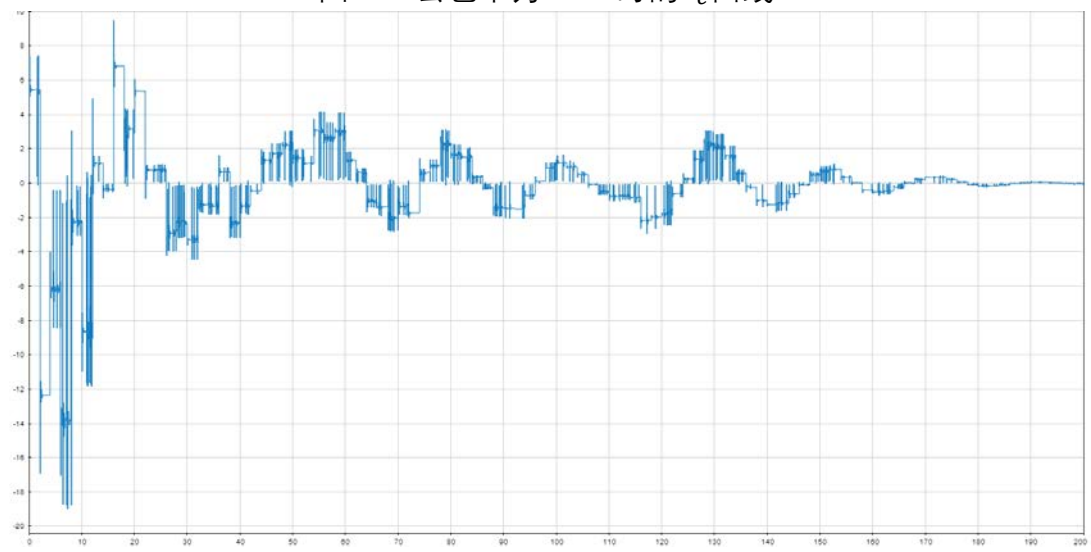


图 7.5 丢包率为 0.12 时的 u_k 曲线

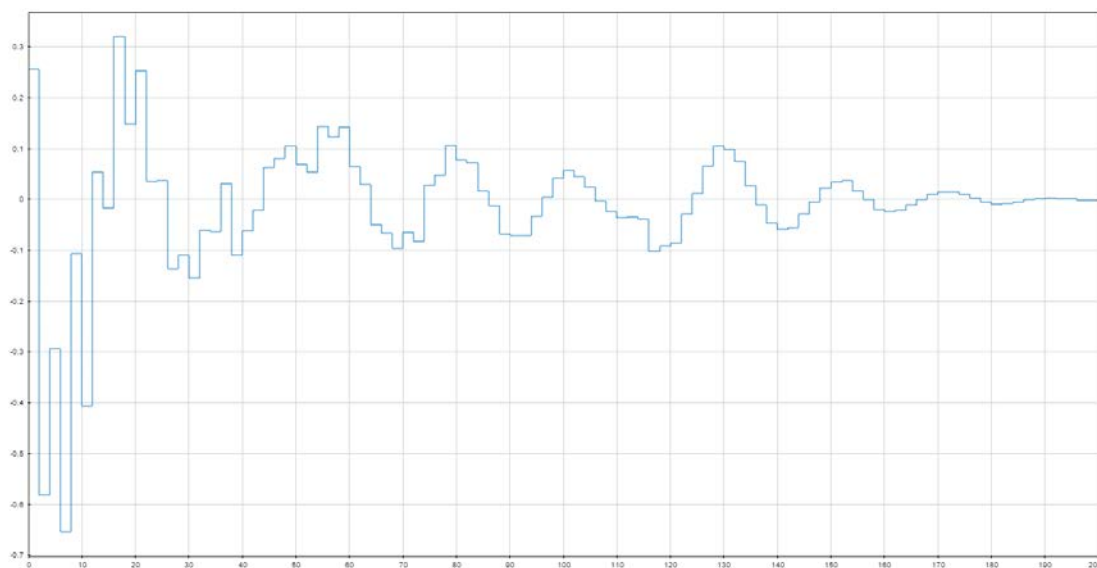


图 7.6 丢包率为 0.12 时的 y_k 曲线

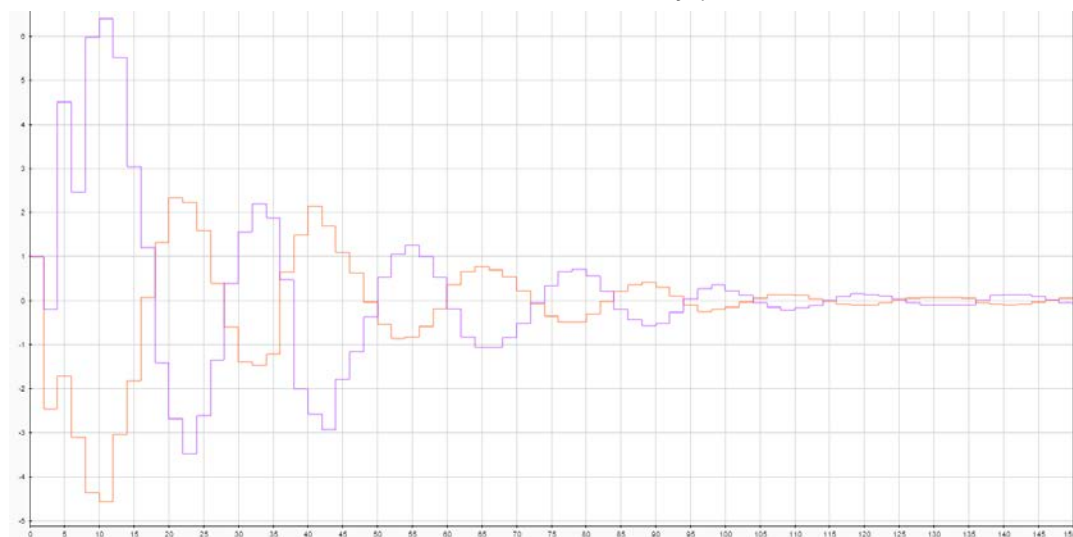


图 7.7 丢包率为 0.12 时的 x 曲线

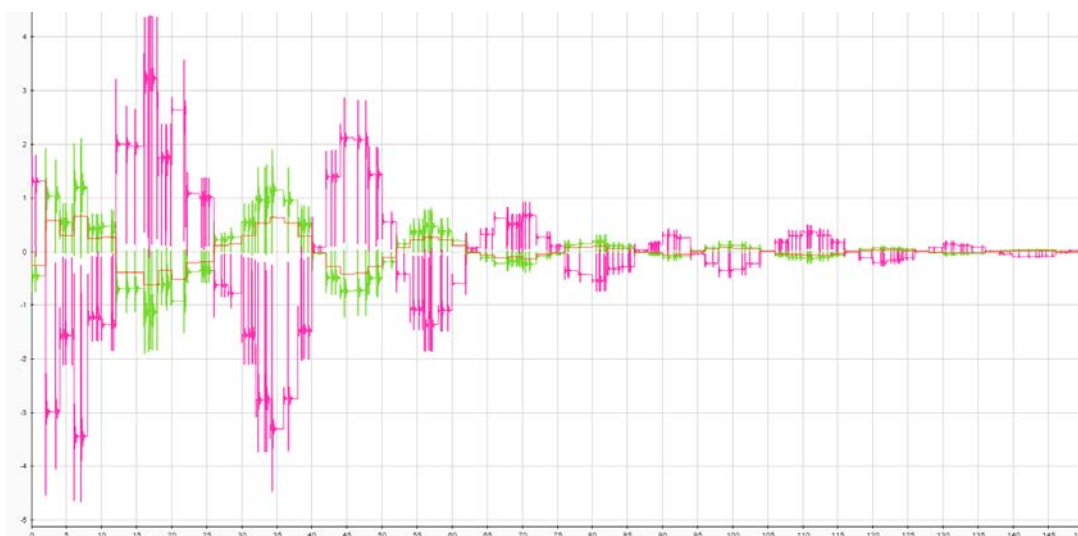


图 7.8 丢包率为 0.12 时的 x_c 曲线

实验结果分析

从图 7.1~7.4 和图 7.5~7.8 的对比中我们不难看出，伴随着丢包率的增加，即使设计的到的控制器参数能使得系统均方稳定，控制效果仍会变差，表现为抖动次数增加、超调幅度增大，过渡时间增长，不稳定的几率增大。比如，从 u_k 来看，丢包率为 0.12 时，系统的在 190s 之后才过渡到平稳状态，且抖动次数较大，抖动幅度较大；而丢包率为 0.05 时稳定时间缩短到 70s 左右，抖动次数减小、抖动幅度也减小。另外地，实验过程中我们发现丢包率为 0.12 时，有更大可能出现系统控制长时间出现小幅震荡的情况。这与理论是相符的，丢包率越大系统越不容易稳定。

此外，系统能否稳定也与传感器节点采样频率以及控制器节点状态空间模块的更新频率有着密切的关系。实验中我们发现，传感器节点采样频率高，被控对象输出值 y 一变化，立刻就能反映到控制器的状态空间表达式里；控制器节点状态空间模块的更新快，在 y_c 这个外界强迫量作用下，控制器节点状态空间能在一个小时间尺度内达到稳态，得到稳定的输出。这可以极大地改善系统的稳定性。

实验体会与感悟

陈思哲：在这次大作业里我主要负责 Truetime 仿真程序的搭建。通过实验，Truetime 工具包的操作，对线性矩阵不等式、Schur 补引理等概念有了更加深刻的理解。更重要的是，从知识体系上来说，通过将囫圇吞枣得来的知识加以运用，网络控制的核心概念已经逐渐连贯，脉络也渐渐清晰。课堂上我将这门课程“学厚”，这次作业则是将它“学薄”了。

张沛东：我主要负责资料查找和公式推导，将问题建模和转化为可解的形式。由于数学艰深，自己的状态空间等的概念基础又薄弱，在公式推导的时候几次心态爆炸。我要感谢陈彩莲老师，她在外出开会的时候仍抽出时间回答同学的问题；我还要感谢张景龙学长，他在实验操作上给出了很专业的指导；我还要感谢我的队友陈思哲、谢敬鱼，我们在实验过程中相互交流，相互鼓励，最终较为圆满地完成了任务。

谢敬鱼：这次大作业我们分工合作，我主要负责编写求解 LMI 的代码，在同组同学理论推导的基础上，使用 LMI toolbox 的求解器 feasp。本次大作业，较上次实验的 LMI 编写更加复杂，这使得我对于该求解器有了更加全面深刻的认知，也能更加熟练的掌握其使用方法。本次作业我们分工合作各司其职，整个过程效率非常高，但也深刻体会到了自己在这方面知识的薄弱。十分感谢陈彩莲老师、张景龙助教、同组两位同学以及同班同学在过程中给予我的帮助，正是这些帮助我们才得以完成本次大作业。