



实验指导书（自控专业卷）

国家级电力生产过程  
虚拟仿真实验教学中心

国家级电力生产过程  
虚拟仿真实验教学中心

## 自动控制专业实验指导书



## 目录

实验一	学习使用远程实验平台进行自动控制原理实验.....	4
实验二	风扇控制系统闭环控制实验.....	13
实验三	使用 PI 算法进行风扇控制实验.....	18
实验四	编写带有积分抗饱和的 PI 控制 S 函数 .....	23
实验五	使用 C 语言编写 S 函数控制风扇远程实验平台 .....	28
实验六	虚拟双容水箱水位远程 PI 控制实验.....	34
实验七	实体双容水箱水位远程 PI 控制实验.....	43
实验八	虚拟球杆系统的 LQR 控制 .....	51
实验九	虚拟球杆系统的 PD 控制 .....	59
实验十	虚拟线性一阶倒立摆的 LQR 控制 .....	67
实验十一	虚拟线性二阶倒立摆的 LQR 控制 .....	75
实验十二	风扇系统数学模型的辨识（综合性实验） .....	84



实验指导书（自控专业卷）

国家级电力生产过程  
虚拟仿真实验教学中心



学习使用远程实验平台进行自动控制原理实验

国家级电力生产过程  
虚拟仿真实验教学中心

## 实验一 学习使用远程实验平台进行自动控制原理实验

### 一 实验目的

- 1、学习远程实验平台 NCSLab，掌握使用此平台进行自动控制原理实验的方法
- 2、通过远程实验开环风扇速度控制系统，从感性上认识开环控制的优点和局限性。

### 二 实验内容

- 1、了解 NCSLab 远程实验平台
- 2、在 NCSLab ([www.powersim.whu.edu.cn/ncslab](http://www.powersim.whu.edu.cn/ncslab)) 的网站上注册自己的账号
- 3、使用 NCSLab 进行风扇控制系统的开环控制实验

### 三 实验原理

#### 1、NCSLab 远程实验平台

网络化控制系统实验室(NCSLab)于 2006 年 11 月 5 日由英国南威尔士大学(University of South Wales)创建。它基于因特网，集合了位于世界各地的控制系统实验平台及设备。在网络化控制系统实验室中，多种经典控制、现代控制、先进控制等实时控制实验都可以通过因特网远程进行。

学生可以在寝室或者家里，通过使用浏览器远程控制实验室内的设备，不需要前往实验室。各个学校的实验设备也可以通过远程实验平台进行共享。

具体信息参见 <http://www.powersim.whu.edu.cn/ncslab>

#### 2、风扇控制系统原理

风扇控制系统放置在主教 917 电力生产过程虚拟仿真实验教学中心的实验室中，如图 1 所示。它已经连入 NCSLab 网络控制平台，允许学生使用 Internet 进行远程访问。

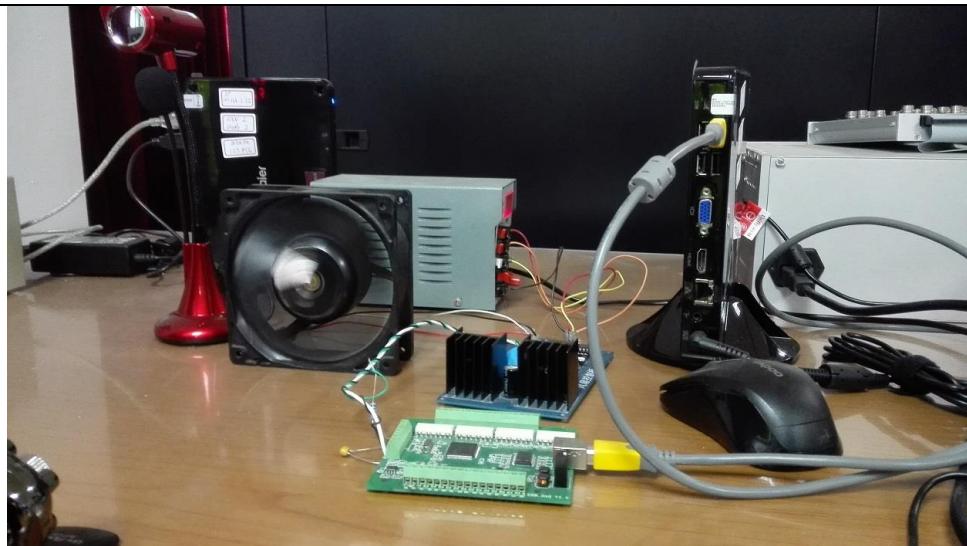


图 1. 实体的风扇控制系统

它的原理框图如图 2 所示。使用控制器在风扇上施加电压，就能驱动风扇转动。通过测量风扇叶片转动的频率，就可以得到风扇的转速。在控制器上使用反馈控制算法，就可以将风扇调节在我们需要的转速上。



图 2. 风扇系统的原理

### 3、风扇控制系统的开环控制实验

通过在风扇上驱动固定的电压，使得风扇旋转。在相同的工况下，各个的电压值对应各自的转速。原理框图如图 3 所示。

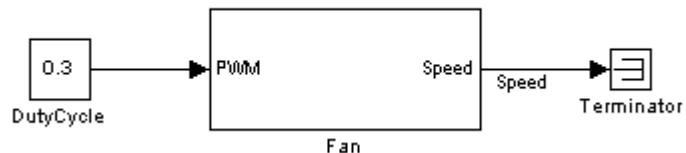


图 3. 风扇系统开环控制的原理

## 四 实验步骤

- 1、登陆 NCSLab 远程实验平台的网站，注册获得自己的实验账号。



学习使用远程实验平台进行自动控制原理实验

国家级电力生产过程  
虚拟仿真实验教学中心

(a) 进入 NCSLab 的网站。打开电脑浏览器 (IE11, Firefox, 或者 Chrome)，登陆 <http://www.powersim.whu.edu.cn/ncslab>，得到如图 4 所示的登陆界面。

The screenshot shows the NCS-3DRS-Lab homepage. At the top left, there is a circular callout bubble pointing to the '注册' (Register) button in the login form. The page features a blue header with the university logo and navigation links like '主页', '实验室', '实验设备', 'NetCon系统', '研究团队', '关于我们', '实验日程', and '课程建设'. Below the header, there's a main content area with sections for '系统入口' (System Entry), '公告' (Announcements), '新闻' (News), '系统介绍' (System Introduction), '图片展示' (Image Display), and '友情链接' (Friendship Links). The '系统介绍' section contains a network diagram showing various components like '总定值', '执行器', '网络化控制器', 'Internet', and '网站服务器'. The '图片展示' section shows images of a '网络化控制器' and '实验设备'.

图 4. NCSLab 的主界面

(b) 获得自己的 NCSLab 账号。在主界面上点击注册按钮，进入注册界面（如图 6 所示）。依次输入自己的个人信息，点击注册完成注册过程。注意：登陆 ID 设定使用自己的学号 (wz+16+学号，如自己学号 109，则登陆 ID 为 wd16109)，验证码使用 whu2015，如图 5 所示。



学习使用远程实验平台进行自动控制原理实验

国家级电力生产过程  
虚拟仿真实验教学中心

NCS-3DRS-Lab  
网络化控制系统 3D 实时仿真实验室

登录 首页 我的NCS-3DRS-Lab English

注册NCS-3DRS-Lab

注册ID使用WD+自己的学号

设定登录ID: wd500 2-12字符英文字母或数字,以字母开头

设定密码: \*\*\*\*\* 密码2-12字符由组成

确认密码: \*\*\*\*\*

电子邮件地址: aa@whu.edu.cn E-mail信箱用于激活帐号和找回密码

真实姓名: 张三 真实姓名必须输入

证件号码:

单位/部门: 动力系

验证码使用whu2015

联系地址: 5舍5517

联系电话: 1345566 联系电话必须输入

输入验证码: whu2015|  Wl CQHZ 内网注册需输入指定验证码

注册

Powered by © ncslab.net.

图 5. 注册自己的 NCSLab 账号

## 2、使用 NCSLab 平台, 进行开环风扇控制系统的远程实验

(a) 在 NCSLab 主界面上输入自己的账号和密码, 进入 NCSLab 的实验界面, 如图 6 所示。在风扇控制实验室内, 一共有两台风扇控制系统可供使用, 可以选择其中任何一台进行实验。

(b) 选中风扇控制系统之后, 进入如图 7 所示的界面。如果想要进行实验, 首先要获得系统的控制权。只有获得系统控制权, 才可以选择远程下载实验算法或者远程调节实验参数。如果没有系统控制权, 则只能观察实验数据。

(c) 点击申请控制权按钮 (如图 7 所示), 获得 30 分钟系统的控制权, 这样就可以继续进行实验了。如果此时有其他用户正在进行实验, 则需要等待, 系统会提示预计等待时间。



学习使用远程实验平台进行自动控制原理实验

国家级电力生产过程  
虚拟仿真实验教学中心

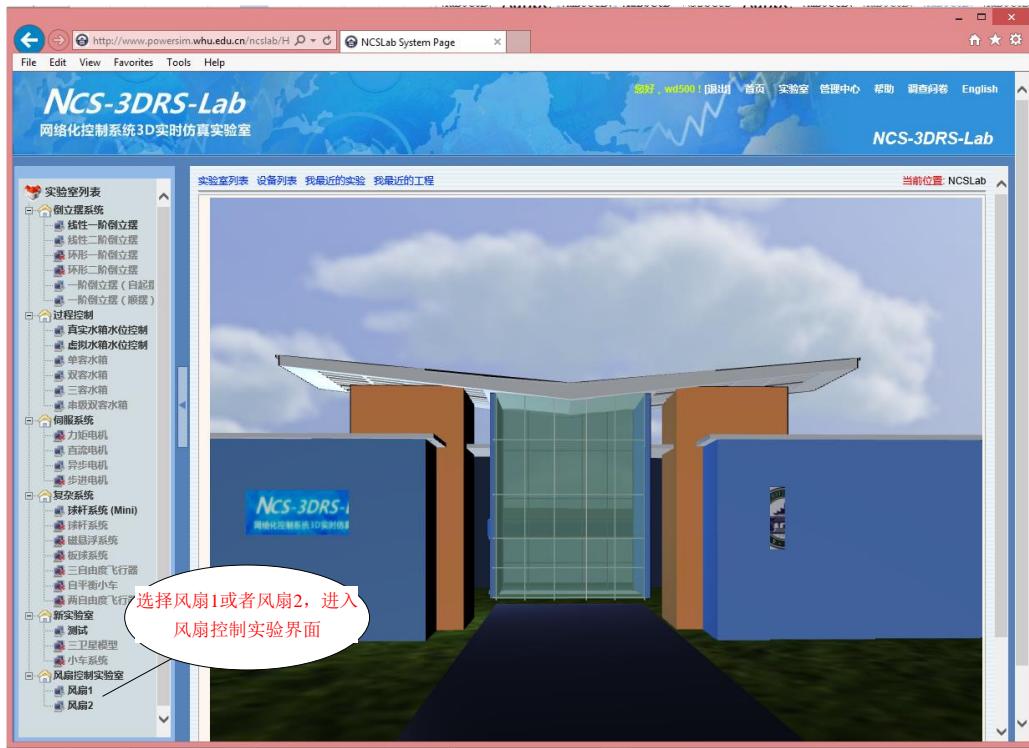


图 6. NCSLab 主界面



图 7. 风扇控制实验主界面

(d) 在获得系统控制权后，点击实验算法，进入实验算法选择界面，如图



学习使用远程实验平台进行自动控制原理实验

国家级电力生产过程  
虚拟仿真实验教学中心

8 所示。一共有两种实验算法可供选择，一种是闭环的 PI 控制（PI Control），另  
一种是开环控制（OpenLoop）。



图 8. 控制算法选择界面

(e) 点击 OpenLoop 实验算法的“开始实验”连接，将开环控制程序远程  
下载到实验装置上。远程实验正式开始，现在可以远程观察实验数据了。

(f) 点击“新建监控组态”链接（如图 9 所示），在弹出窗口中建立组态界  
面，远程观察实验状态，修改实验参数。

(g) 弹出的窗口如图 10 所示。在工具栏中点选“数字输入框”，“趋势图”  
和“实时视频”按钮，获得这三种控件。并将他们拉到合适的位置。



学习使用远程实验平台进行自动控制原理实验

国家级电力生产过程  
虚拟仿真实验教学中心



图 9. 新建组态界面



图 10. 实验组态界面

(h) 双击各个控件，为各个控件选择相关联的参数或信号。“数字输入”控件中选择“DutyCycle”这个参数。“趋势图”控件中选择“Speed”这个信号。



学习使用远程实验平台进行自动控制原理实验

国家级电力生产过程  
虚拟仿真实验教学中心

如图 11 所示。

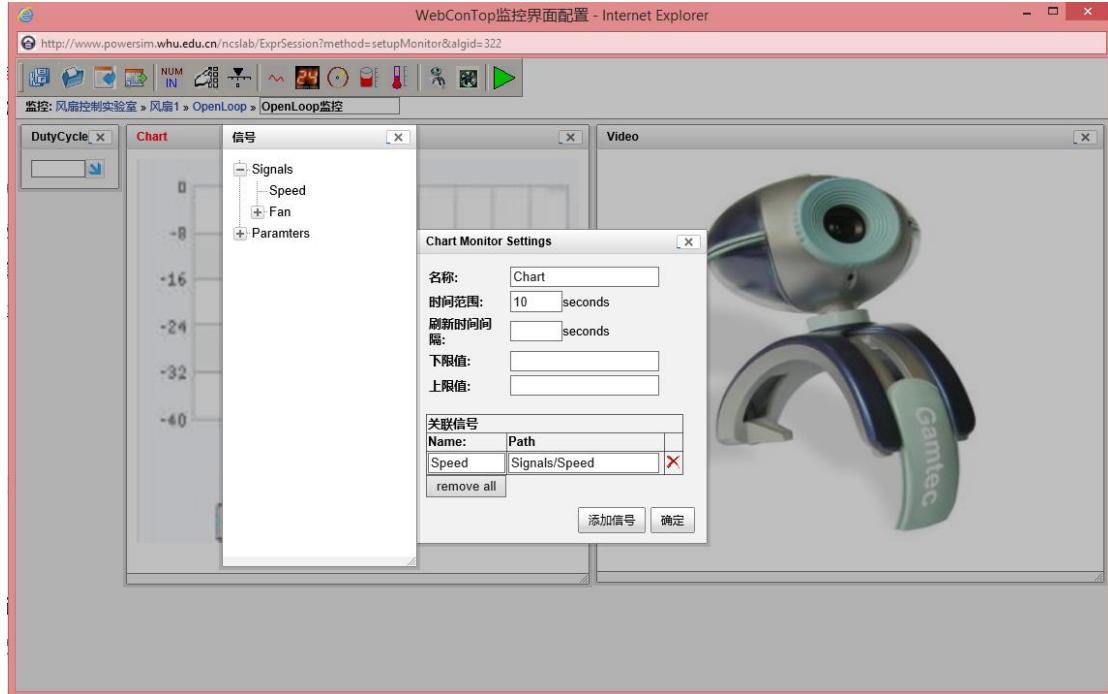


图 11. 为各个控件选择参数或信号

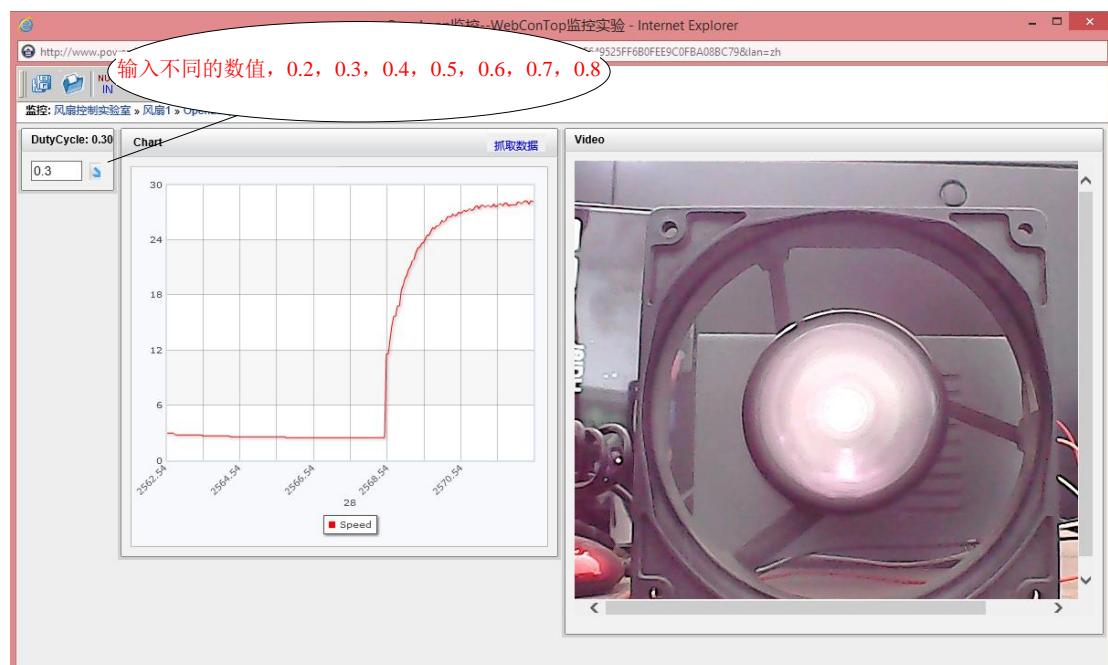


图 12. 远程实验监控

- (i) 点击“运行”按钮，开始远程监控实验进行。如图 12 所示。
- (j) 在“数字输入”控件中分别输入 0.2,0.3,0.4,0.5,0.6,0.7,0.8 的占空比，观



察“趋势图”控件中的系统输出，将稳态值记录下来。

占空比	风扇转速
0.2	
0.3	
...	
0.8	

## 五 思考题

(1) 如果我们想要把风扇转速调整到 50，我们应该怎样使用开环控制实现这个目标？

(2) 通过这次实验，认识到开环控制有什么局限性？

## 六 实验报告

(1) 需要将实验过程中的每一个关键步骤抓图，粘贴到实验报告上。

(2) 每一位同学对应一个实验账号，完成一份实验报告，学号-实验报告-实验账号要保持一一对应关系。在实验报告中要标明自己的实验账号。

(3) 实验报告中要回答思考题提出的问题（没有标准答案，按照自己的理解回答）。



## 实验二 风扇控制系统闭环控制实验

### 一 实验目的

- 1、通过与开环控制的比较，了解闭环控制系统的优点
- 2、从感性上了解控制参数对控制效果的影响

### 二 实验内容

- 1、进一步了解 NCSLab 远程实验平台
- 2、使用 NCSLab 进行风扇控制系统的闭环控制实验

### 三 实验原理

#### 1、风扇控制系统原理

风扇控制系统放置在主教 917 电力生产过程虚拟仿真教学中心的实验室中，如图 1 所示。它已经连入 NCSLab 网络控制平台，允许学生使用 Internet 进行远程访问。

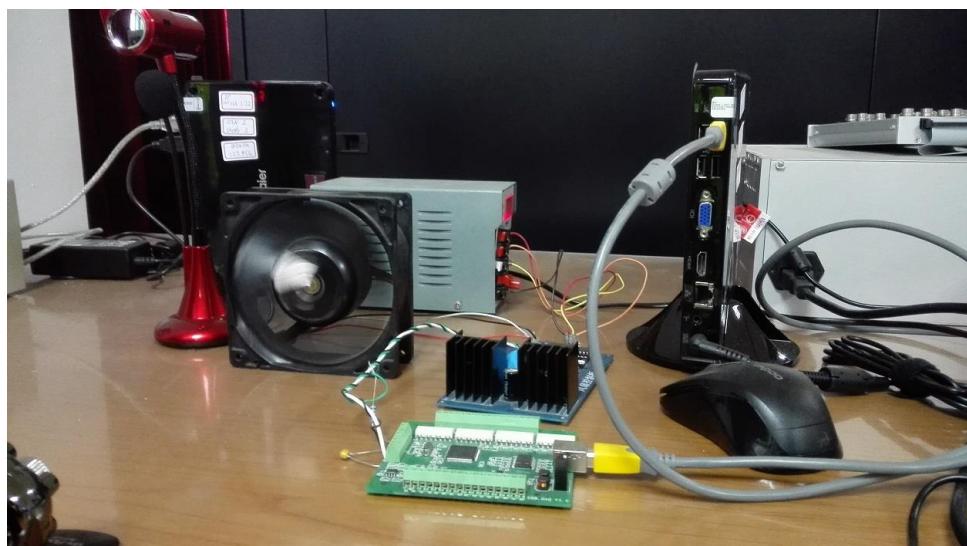


图 1. 实体的风扇控制系统

它的原理框图如图 2 所示。使用控制器在风扇上施加电压，就能驱动风扇转动。通过测量风扇叶片转动的频率，就可以得到风扇的转速。在控制器上使用反馈控制算法，就可以将风扇调节在我们需要的转速上。



图 2. 风扇系统的原理

## 2、风扇控制系统的闭环 PID 控制实验

控制算法比较风扇的实际转速与设定转速的差值，通过 PID 控制算法计算实时的控制量（电压或 PWM）。通过不断的比较求差，调整控制量的输出，指导风扇转速达到预期的设定值为止（或者误差小于一定范围）。控制框图如图 3 所示。

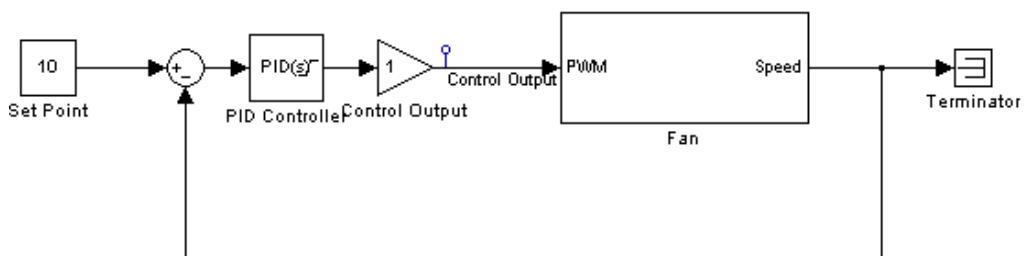


图 3. 风扇系统闭环控制的原理

## 四 实验步骤

1、使用自己的实验账号登陆 NCSLab 网站，找到并且进入风扇控制（风扇 1 或风扇 2）的页面。

2、在实验算法中找到“PI Control”算法，并且把这个算法下载到远程的控制器中。

3、点击“新建监控组态”链接，建立如图 4 的组态。拖动三个“数字输入框”控件（分别于 Paramters/Set\_Point，Paramters/PID\_Controller/Proportional\_Gain，Paramters/PID\_Controller/Integral\_Gain 关联），一个“液晶显示框”控件（与 Signals/Control\_Output 关联），一个“趋势图”控件（与 Paramters/Set\_Point 和 Signals/Speed 关联），一个“实时视频”控件。



## 风扇控制系统闭环控制实验

国家级电力生产过程  
虚拟仿真实验教学中心

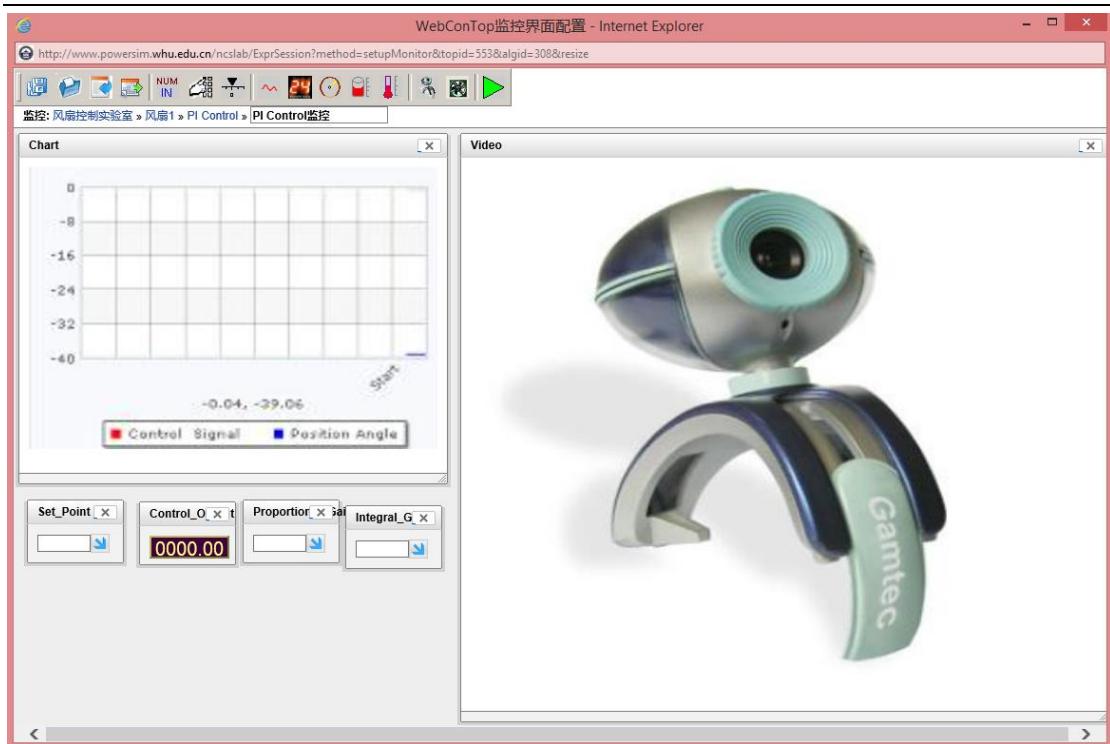


图 4. 闭环控制组态

- 4、点击“运行”按钮，开始实验监控，如图 5 所示。
- 5、将设定值“set\_point”从 10 改为 50，观察趋势图上的过渡过程，并且抓图（附在实验报告上）。
- 6、将积分参数 Integral\_Gain 从 0.1 改为 0，再次将设定值“set\_point”从 10 改为 50，观察趋势图上的过度过程，分析为什么会有静差（抓图）。

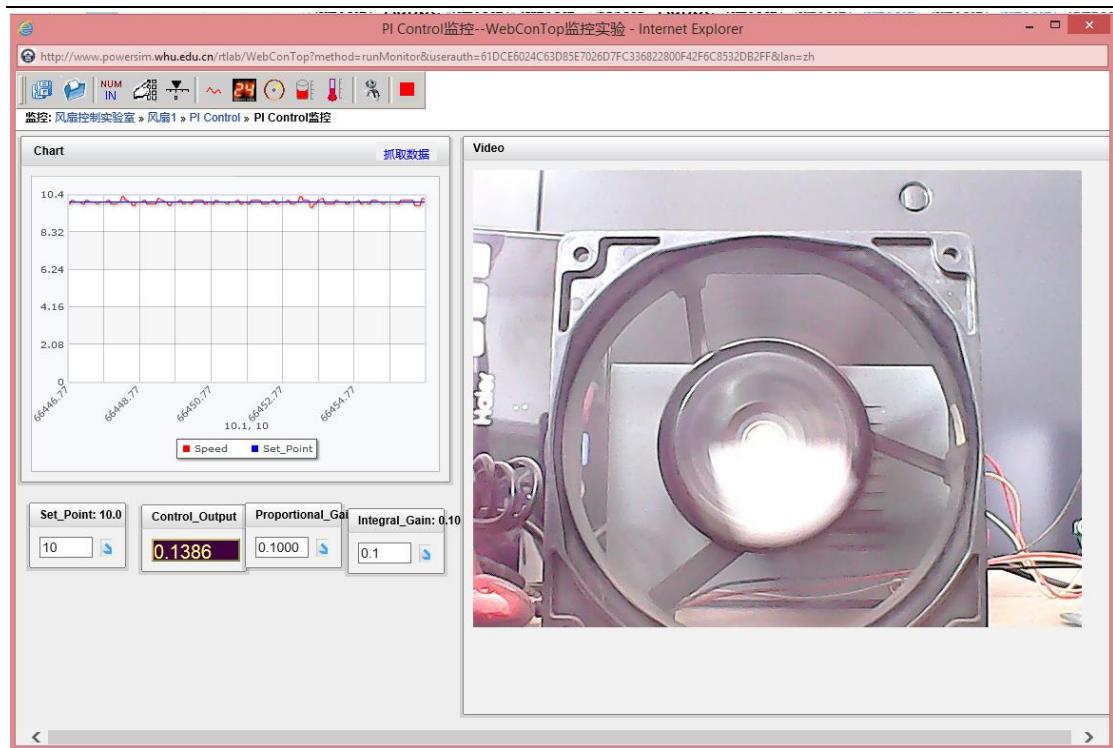


图 5. 闭环控制远程实验

7、将积分参数 Integral\_Gain 从 0 改为 1，第三次将设定值“set\_point”从 10 改为 50，观察趋势图上的过度过程，分析与前两次对比有什么区别（抓图）。

8、将比例参数 Proportional\_Gain 设为 0.01，积分参数 Integral\_Gain 保持最初的 0.1，第三次将设定值“set\_point”从 10 改为 50，观察趋势图上的过度过程，分析这次又什么特点。

9、从稳定性，准确性和快速性三个方面定性分析步骤 5-8 的四次阶跃实验的曲线，完成下表（可以主观发挥，用文字描述）。

	稳定性	准确性	快速性
5			
6			
7			
8			

10、自由修改控制参数，体会控制参数对控制效果的影响。

## 五 思考题

(1) 结合本次实验和前一次实验，比较与开环控制相比，闭环控制有什么优点。

(2) 结合本次实验，描述积分项对闭环控制效果的影响（从稳定性，准确性和快速性）三个方面阐述。



## 六 实验报告

- (1) 需要将实验过程中的每一个关键步骤抓图，粘贴到实验报告上。
- (2) 每一位同学对应一个实验账号，完成一份实验报告，学号-实验报告-实验账号要保持一一对应关系。在实验报告中要标明自己的实验账号。
- (3) 实验报告中要回答思考题提出的问题。



## 实验三 使用 PI 算法进行风扇控制实验

### 一 实验目的

- 1、通过实验了解 PI 算法的具体特性
- 2、了解积分抗饱和算法的原理和使用方法

### 二 实验内容

- 1、搭建软件平台，使用 Simulink 和 Real-Time Workshop 自动生成可执行算法
- 2、使用 Simulink 搭建 PI 和抗饱和 PI 控制算法的框图，生成对应的控制程序，并且使用远程控制平台 NCSLab 将它们下载到远程的风扇控制系统中运行。

### 三 实验原理

#### 1、风扇控制系统原理

风扇控制系统放置在主教 917 电力生产过程虚拟仿真教学中心的实验室中，如图 1 所示。它已经连入 NCSLab 网络控制平台，允许学生使用 Internet 进行远程访问。



图 1. 实体的风扇控制系统

它的原理框图如图 2 所示。使用控制器在风扇上施加电压，就能驱动风扇转动。通过测量风扇叶片转动的频率，就可以得到风扇的转速。在控制器上使用反馈控制算法，就可以将风扇调节在我们需要的转速上。



图 2. 风扇系统的原理

## 2、风扇控制系统的 PI 控制算法

风扇 PI 控制框图如图 3 所示。PI 控制器由比例和积分两部分组成，两部分的计算量通过一个加法器连接起来，共同构成最终的控制量。

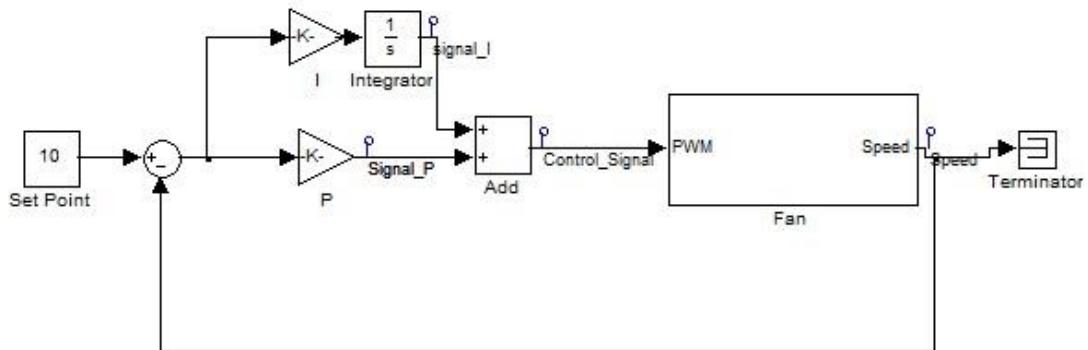


图 3. 风扇系统闭环 PI 控制的原理

## 3、风扇控制系统的抗饱和 PI 控制算法

风扇抗饱和 PI 控制算法如图 4 所示。当控制量的输出超出饱和范围的时候，就通过一个乘法器关闭系统的积分器，这样就实现了抗饱和的 PI 控制算法。

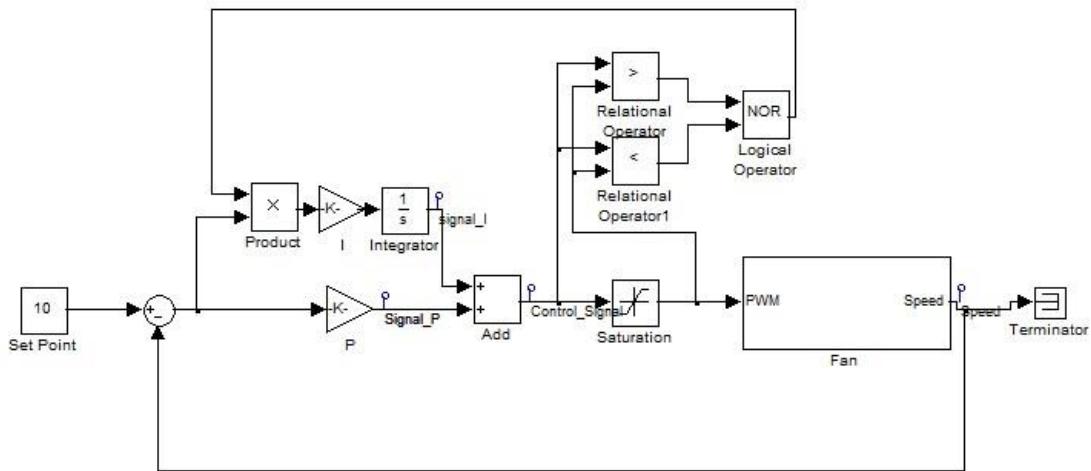


图 4. 风扇系统抗饱和 PI 控制的原理

#### 四 实验步骤

- 1、安装 WMWare 虚拟机软件
- 2、将带有 Matlab 和 Visual Studio 的 Win7 虚拟机导入到虚拟机软件中
- 3、在虚拟机软件中运行 Win7 系统
- 4、打开虚拟机中的 Matlab 软件，打开 Simulink
- 5、打开 FanTest.mdl，在此基础上修改，建立如图 3 所示的 PI 控制框图，其中 P 参数和 I 参数都设为 0.1
- 6、使用 Simulink 生成可执行算法，通过复制粘贴把可执行程序从虚拟机中拷贝出来
- 7、去掉可执行文件的.exe 后缀，使用自己的账号登陆 NCSLab，将算法文件上传到服务器上。
- 8、获得系统控制权，将上传算法下载到远程的风扇控制系统中去
- 9、建立组态，如图 5 所示。将风扇转速的设定值从 30 调整到 60，分别观察设定值与实际风扇速度反馈的关系（放在一张趋势图上），观察控制过程中比例项输出、积分项输出与总控制量之间的关系（放在一张图）。



使用 PI 算法进行风扇控制实验

国家级电力生产过程  
虚拟仿真实验教学中心

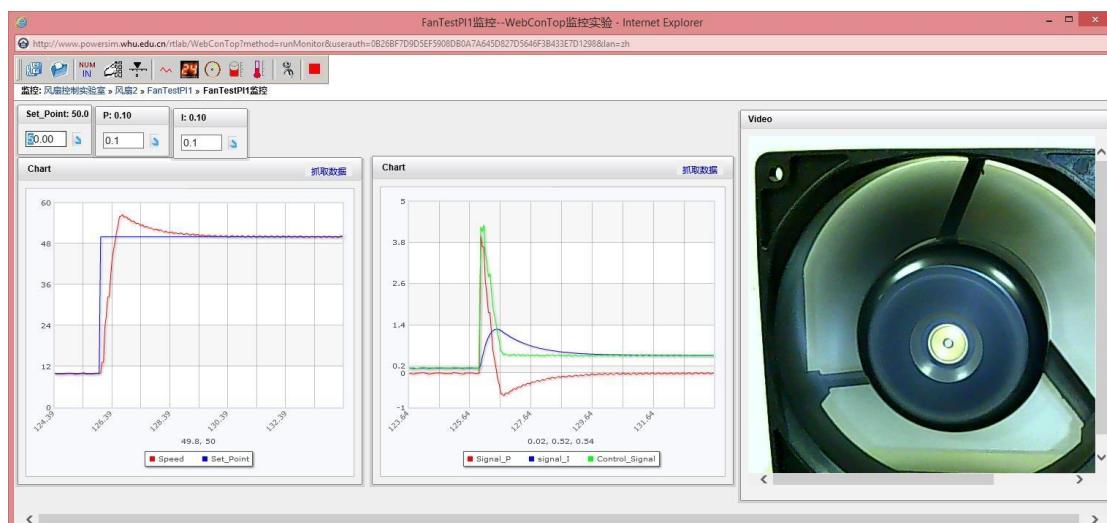


图 5. PI 控制组态

- 10、将设定值设为 200（饱和状态），观察积分项是如何饱和的。
  - 11、将设定值设定到 50，观察风扇的实际速度有什么反应，是否有滞后，为什么会有滞后。
  - 12、返回到虚拟机，建立如图 4 所示的抗饱和 PI 控制的框图，并且生成可执行文件，通过 NCSLab 下载到远程控制器中。
  - 13、建立和 5 相似的组态，将风扇转速的设定值从 30 调整到 60，观察各个控制量之间的关系。同没有抗积分饱和的算法相比，系统的响应有什么区别。
  - 14、将设定值设为 200（饱和状态），观察积分项是否饱和。
  - 15、将设定值设定到 50，观察风扇的实际速度有什么反应，是否滞后消失，为什么滞后会消失。
- ## 五 思考题
- (1) 结合本次实验的结果，描述图 3 的 PI 控制算法的原理
  - (2) 结合本次实验的结果，描述图 4 的抗饱和 PI 控制算法的原理，抗饱和是如何实现的（请结合实验曲线图阐述）。
  - (3) 同样是 P 和 I 均为 0.1 的参数，为什么使用普通 PI 控制和抗饱和 PI 控制算法，系统的响应是不一样的，请结合自己的实验数据阐述（没有标准答案）。
- ## 六 实验报告
- (1) 需要将实验过程中的每一个关键步骤抓图（实验监控画面和 Simulink 框图），粘贴到实验报告上。
  - (2) 每一位同学对应一个实验账号，完成一份实验报告，学号实验报告—实验账号要保持一一对应关系。在实验报告中要标明自己的实验账号。



使用 PI 算法进行风扇控制实验

国家级电力生产过程  
虚拟仿真实验教学中心

- 
- (3) 实验报告中要回答思考题提出的问题。
  - (4) 如果有兴趣，可以自行设计一些其他算法，并且将结果附在实验报告上（酌情加分）。



## 实验四 编写带有积分抗饱和的 PI 控制 S 函数

### 一 实验目的

- 1、了解 Simulink 环境下使用 M 语言编写 S 函数的方法
- 2、使用 S 函数编写带有积分抗饱和的 PI 控制模块

### 二 实验内容

- 1、使用 M 语言编写实现积分抗饱和 PI 控制 S 函数
- 2、使用编写的抗饱和 PI 控制器，在 Simulink 环境下建立仿真框图，实现风扇的速度控制

### 三 实验原理

- 1、风扇控制系统的数学模型

风扇系统的原理框图如图 1 所示。使用控制器在风扇上施加电压，就能驱动风扇转动。通过测量风扇叶片转动的频率，就可以得到风扇的转速。



图 1. 风扇系统的原理

风扇系统可以近似地看作为一个一阶惯性环节。它的传递函数可近似辨识为

$$G(s) = \frac{90}{0.8s + 1}$$

因为 PWM 占空比的输入范围是 0.1 至 0.9 之间，因此，在 Simulink 环境下，风扇系统的数学模型可以用图 2 搭建的框图来表示。其中饱和模块的上下限值分别为 0.9 和 0.1。

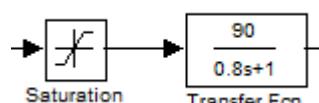


图 2. 风扇的数学模型

- 2、使用 S 函数实现风扇控制系统的 PI 控制

编写 S 函数 PIM.m，可以实现 PI 控制模块。搭建框图如图 3，则可以实现风扇的 PI 控制。

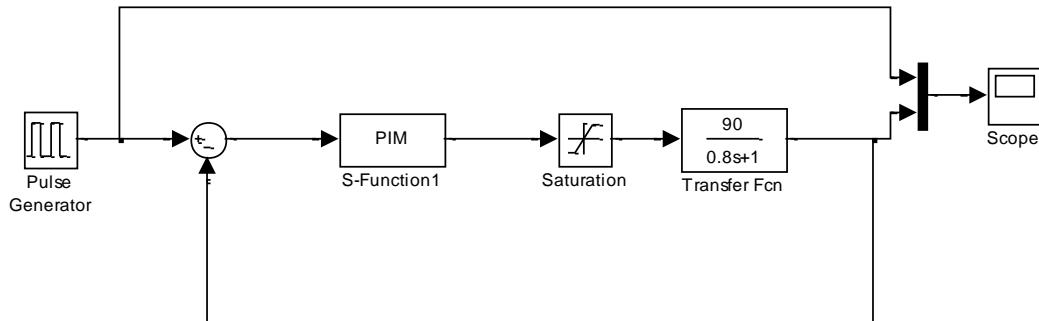


图 3. 使用 S 函数实现风扇速度 PI 控制

### 3、使用 S 函数实现风扇控制系统积分抗饱和 PI 控制

通过编写积分抗饱和 PI 控制的 S 函数 PIM1.m，搭建如图 4 的 Simulink 模型，可以实现带有积分抗饱和的 PI 风扇速度控制。

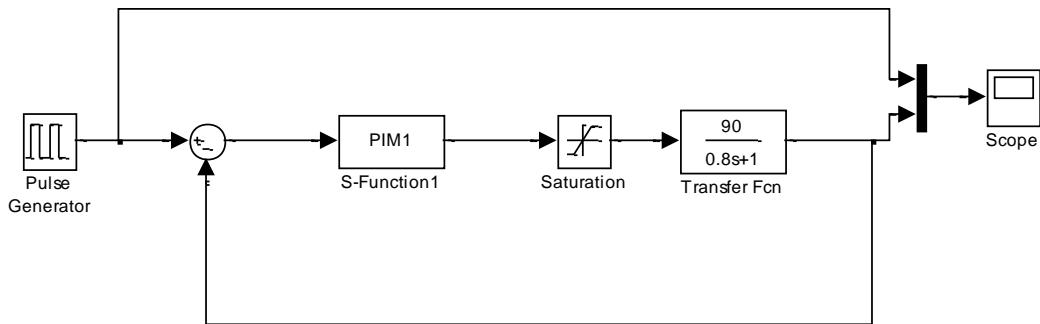


图 4. 使用 S 函数实现风扇速度积分抗饱和 PI 控制

## 四 实验步骤

- 1、编写 PI 控制的 S 函数 PIM.m
- 2、搭建如图 4 所示的风扇 PI 控制框图，其中 S 函数的参数分别输入 0.1,0.1，表示 P 和 I 参数均为 0.1
- 3、参考信号采用方波，周期为 10 秒，占空比 50%，幅值 100
- 4、进行仿真并观察响应曲线，是否出现了积分饱和的现象，将响应曲线抓图
- 5、编写带有积分抗饱和的 PI 控制 S 函数 PIM1.m，替换原 S 函数 PIM.m，S 函数的参数分别输入 0.1,0.1,0.1,0.9，表示 P 和 I 参数均为 0.1，积分饱和下限 0.1，积分饱和上限 0.9
- 6、进行仿真并观察相应曲线，积分饱和的现象是否消失，将响应曲线抓图

## 五 思考题



(1) 解释为什么 PIM1.m 能够实现积分抗饱和的 PI 控制，结合 S 函数的编写原理，解释关键代码。

(2) 将 PIM1.m 封装成一个带有四个参数的 Simulink 模块，四个参数分别为 P, I, Lower Limit, 和 Upper Limit。

## 六 实验报告

(1) 需要将实验过程中的每一个关键步骤抓图粘贴到实验报告上。

(2) 实验报告中要回答思考题提出的问题。

### 附录 1, PIM.m

```
function [sys, x0, str, ts] = PIM(t, x, u, flag, P, I)
    switch flag,
        case 0,
            [sys, x0, str, ts] = mdlInitializeSizes();
        case 1,
            sys = mdlDerivatives(t, x, u, P, I);
        case 2 ,
            sys = mdlUpdate(t, x, u);
        case 3,
            sys = mdlOutputs(t, x, u, P);
        case 4,
            sys = mdlGetTimeOfNextVarHit(t, x, u);
        case 9,
            sys = mdlTerminate (t , x , u);
        otherwise
            error ( [ 'Unhandled flag =' ,num2str ( flag ) ] );
    end
```

```
function [sys,x0,str,ts]=mdlInitializeSizes()
    sizes =simsizes;
    sizes.NumContStates=1;
    sizes.NumDiscStates=0;
    sizes.NumOutputs=1;
    sizes.NumInputs=1;
    sizes.DirFeedthrough=1;
    sizes.NumSampleTimes=1;
    sys=simsizes(sizes);
    x0=0;
    str=[ ];
```



---

ts=[0, 0];

```
function sys=mdlDerivatives(t,x,u,P,I)
    sys=I*u;

function sys=mdlUpdate(t,x,u)
    sys=[ ];

function sys=mdlOutputs(t,x,u,P)
    sys=P*u+x;

function sys=mdlGetTimeOfNextVarHit(t,x,u)
    sampleTime=1;% Example, set the next hit to be one second later
    sys=t+ sampleTime;

function sys=mdlTerminate(t,x,u)
    sys = [ ];
```

## 附录2， PIM1.m

```
function [sys, x0, str, ts] = PIM1(t, x, u, flag, P, I, lowerLimit, upperLimit)
    switch flag,
        case 0,
            [sys, x0, str, ts] = mdlInitializeSizes();
        case 1,
            sys = mdlDerivatives(t, x, u, P, I, lowerLimit, upperLimit);
        case 2 ,
            sys = mdlUpdate(t, x, u);
        case 3,
            sys = mdlOutputs(t, x, u, P);
        case 4,
            sys = mdlGetTimeOfNextVarHit(t, x, u);
        case 9,
            sys = mdlTerminate (t , x , u);
        otherwise
            error ( [ 'Unhandled flag =' ,num2str ( flag ) ] );
    end

    function [sys,x0,str,ts]=mdlInitializeSizes()
        sizes =simsizes;
```



---

```
sizes.NumContStates=1;
sizes.NumDiscStates=0;
sizes.NumOutputs=1;
sizes.NumInputs=1;
sizes.DirFeedthrough=1;
sizes.NumSampleTimes=1;
sys=simsizes(sizes);
x0=0;
str=[ ];
ts=[0, 0];

function sys=mdlDerivatives(t,x,u,P,I,lowerLimit, upperLimit)
if(P*u+x>lowerLimit&&P*u+x<upperLimit)
    sys=I*u;
else
    sys=0;
end

function sys=mdlUpdate(t,x,u)
sys=[ ];

function sys=mdlOutputs(t,x,u,P)
sys=P*u+x;

function sys=mdlGetTimeOfNextVarHit(t,x,u)
sampleTime=1;% Example, set the next hit to be one second later
sys=t+ sampleTime;

function sys=mdlTerminate(t,x,u)
sys = [ ];
```



## 实验五 使用 C 语言编写 S 函数控制风扇远程实验平台

### 一 实验目的

- 1、了解 Simulink 环境下使用 C 语言编写 S 函数的方法
- 2、了解控制算法从仿真到实体实验的设计方法

### 二 实验内容

- 1、使用 C 语言编写实现积分抗饱和 PI 控制 S 函数
- 2、使用编写的 S 函数，在 Simulink 环境下建立仿真框图，实现风扇的速度控制
- 3、使用编写的 S 函数建立风扇速度控制实体实验的框图，利用 Real Time Workshop 生成控制程序，并通过 NCSLab 平台实施在远程的实验平台上。

### 三 实验原理

- 1、风扇控制系统的数学模型

风扇系统的原理框图如图 1 所示。使用控制器在风扇上施加电压，就能驱动风扇转动。通过测量风扇叶片转动的频率，就可以得到风扇的转速。



图 1. 风扇系统的原理

风扇系统可以近似地看作为一个一阶惯性环节。它的传递函数可近似辨识为

$$G(s) = \frac{90}{0.8s + 1}$$

因为 PWM 占空比的输入范围是 0.1 至 0.9 之间，因此，在 Simulink 环境下，风扇系统的数学模型可以用图 2 搭建的框图来表示。其中饱和模块的上下限值分别为 0.9 和 0.1。

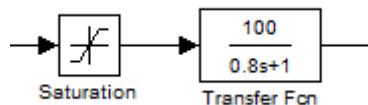


图 2. 风扇的数学模型

- 2、使用 S 函数实现积分抗饱和 PI 控制仿真



使用 C 语言编写积分抗饱和 PI 控制的 S 函数 PIM1, 搭建如图 3 的 Simulink 模型, 可以实现带有积分抗饱和的 PI 风扇速度控制的仿真。在仿真框图中, 被控对象是风扇的数学模型。

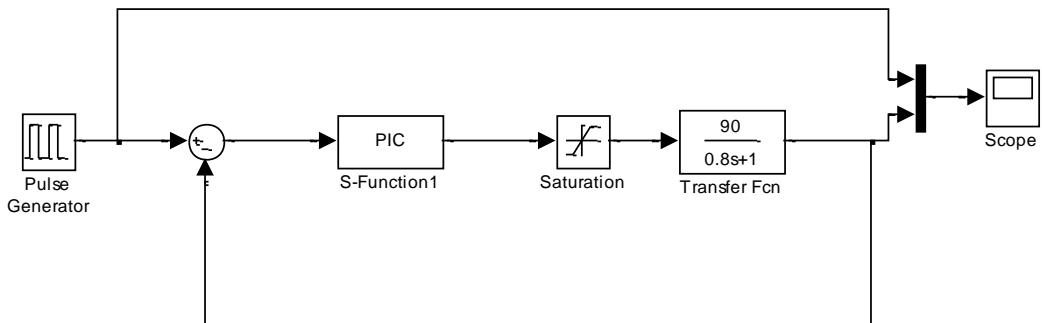


图 3. 仿真框图

### 3、使用 S 函数实现积分抗饱和 PI 控制的远程实体实验

用自己编写的 S 函数替换实验二控制框图中的 PI 控制器, 搭建风扇速度控制实体实验的 Simulink 框图, 如图 4 所示。生成可执行文件, 并且实施在远程实验平台上。

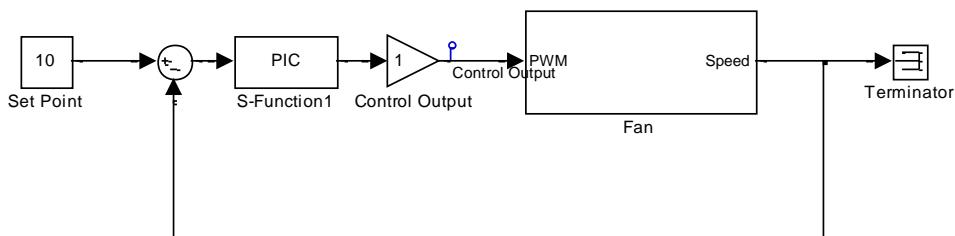


图 4. 实体实验框图

## 四 实验步骤

- 1、使用 C 语言编写带有积分抗饱和 PI 控制的 S 函数 PIC.c
- 2、搭建如图 3 所示的风扇 PI 控制仿真框图, 其中 S 函数的参数分别输入 0.1, 0.1, 0.1, 0.1, 0.1, 0.9, 表示 P 和 I 参数均为 0.1, 积分饱和下限 0.1, 积分饱和上线 0.9
- 3、参考信号采用方波, 周期为 10 秒, 占空比 50%, 幅值 100
- 4、进行仿真并观察响应曲线, 是否出现了积分饱和的现象, 将曲线抓图。
- 5、将方波信号的幅值改为 50, Delay 设为 3 秒。运行仿真, 观察速度从 10 阶跃到 50 的过渡过程曲线。将响应曲线抓图。



6、用自己编写的 S 函数替换实验三框图中的 PI 控制器，生成可执行程序，并上传到 NCSLab 的网站。

7、获得系统控制权，将上传算法下载到远程的风扇控制系统中去

8、建立组态，将风扇转速的设定值从 10 调整到 50，分别观察设定值与实际风扇速度反馈的关系（放在一张趋势图上），并且抓图。

9、用步骤 8 中的实体实验的结果，与步骤 5 中仿真的结果想比较，观察两者的契合度，仿真程序是否如实反映了实体实验的动态特性。

## 五 思考题

(1) 如何通过实验 1 的实验结果，估计出风扇系统的数学模型

(2) 解释为什么 PIC.c 能够实现积分抗饱和的 PI 控制，结合 S 函数的编写原理，解释关键代码。

## 六 实验报告

(1) 需要将实验过程中的每一个关键步骤抓图粘贴到实验报告上。

(2) 实验报告中要回答思考题提出的问题。

(3) 如果有兴趣，可以自行设计一些其他算法，并且将结果附在实验报告上（酌情加分，总成绩加 0-5 分）。

(4) 如果有兴趣，可以自行设计系统辨识的方法，采集数据，辨识出风扇系统更准确的数学模型。（酌情加分，总成绩加 0-10 分）。

附录 1，PIC.c

```
#define S_FUNCTION_NAME PIC
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"

#define U(element) (*uPtrs[element]) /* Pointer to Input Port0 */

#define P(S) (ssGetSFcnParam(S,0))
#define I(S) (ssGetSFcnParam(S,1))
#define LOWER(S) (ssGetSFcnParam(S,2))
#define UPPER(S) (ssGetSFcnParam(S,3))

/*=====
 * S-function methods
 =====*/
/* Function: mdlInitializeSizes
=====
=====
```



## \* Abstract:

\*     The sizes information is used by Simulink to determine the S-function  
\*     block's characteristics (number of inputs, outputs, states, etc.).  
\*/

```
static void mdlInitializeSizes(SimStruct *S)
{
    ssSetNumSFcnParams(S, 4); /* Number of expected parameters */
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
        return; /* Parameter mismatch will be reported by Simulink */
    }

    ssSetNumContStates(S, 1);
    ssSetNumDiscStates(S, 0);

    if (!ssSetNumInputPorts(S, 1)) return;
    ssSetInputPortWidth(S, 0, 1);
    ssSetInputPortDirectFeedThrough(S, 0, 1);

    if (!ssSetNumOutputPorts(S, 1)) return;
    ssSetOutputPortWidth(S, 0, 1);

    ssSetNumSampleTimes(S, 1);
    ssSetNumRWork(S, 0);
    ssSetNumIWork(S, 0);
    ssSetNumPWork(S, 0);
    ssSetNumModes(S, 0);
    ssSetNumNonsampledZCs(S, 0);
    ssSetSimStateCompliance(S, USE_DEFAULT_SIM_STATE);

    /* Take care when specifying exception free code - see sfunmpl_doc.c */
    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}
```

## /\* Function: mdlInitializeSampleTimes

## \* Abstract:

\*     Specify that we have a continuous sample time.  
\*/

```
static void mdlInitializeSampleTimes(SimStruct *S)
{
```



```
ssSetSampleTime(S, 0, CONTINUOUS_SAMPLE_TIME);
ssSetOffsetTime(S, 0, 0.0);
ssSetModelReferenceSampleTimeDefaultInheritance(S);
}
```

```
#define MDL_INITIALIZE_CONDITIONS
/* Function: mdlInitializeConditions
=====
* Abstract:
*   Initialize both continuous states to zero.
*/
static void mdlInitializeConditions(SimStruct *S)
{
    real_T *x0 = ssGetContStates(S);

    *x0=0;
}

/* Function: mdlOutputs
=====
* Abstract:
*    $y = Cx + Du$ 
*/
static void mdlOutputs(SimStruct *S, int_T tid)
{
    real_T          *y      = ssGetOutputPortRealSignal(S,0);
    real_T          *x      = ssGetContStates(S);
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);
    real_T P=(*mxGetPr(P(S)));

    UNUSED_ARG(tid); /* not used in single tasking mode */

    /*  $y=Cx+Du$  */
    y[0]=x[0]+P*U(0);
}

#define MDL_DERIVATIVES
/* Function: mdlDerivatives
=====
* Abstract:
*    $\dot{x} = Ax + Bu$ 

```



```
/*
static void mdlDerivatives(SimStruct *S)
{
    real_T             *dx    = ssGetdX(S);
    real_T             *x     = ssGetContStates(S);
    InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,0);

    real_T I=(*mxGetPr(I(S)));
    real_T P=(*mxGetPr(P(S)));
    real_T Lower=(*mxGetPr(LOWER(S)));
    real_T Upper=(*mxGetPr(UPPER(S)));

    /* xdot=Ax+Bu */
    if(P*U(0)+x[0]>Lower&&P*U(0)+x[0]<Upper)
    {
        dx[0]=I*U(0);
    }
    else
    {
        dx[0]=0;
    }
}
/* Function: mdlTerminate
=====
* Abstract:
*   No termination needed, but we are required to have this routine.
*/
static void mdlTerminate(SimStruct *S)
{
    UNUSED_ARG(S); /* unused input argument */
}

#ifndef MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```



## 实验六 虚拟双容水箱水位远程 PI 控制实验

### 一 实验目的

- 1、了解虚拟双容水箱的结构、原理和数学模型
- 2、了解虚拟双容水箱的 PI 控制算法

### 二 实验内容

- 1、了解双容水箱的原理、结构和
- 2、了解双容水箱数学模型的建立方法
- 3、使用远程实验的方式，操作远程的虚拟双容水箱实验平台，调节 PI 控制器参数，优化控制效果，为进一步的实体实验做准备。

### 三 实验原理

真实水箱控制算法。图 1 是为测试真实水箱系统特性而搭建的算法框图，其中限幅模块值从 0 到 inf 保证为正值。Analog Output 和 Analog Input 端口号(port number) 均为 0, Analog Input 端口号为 1, 采样时间为 0.04 秒。

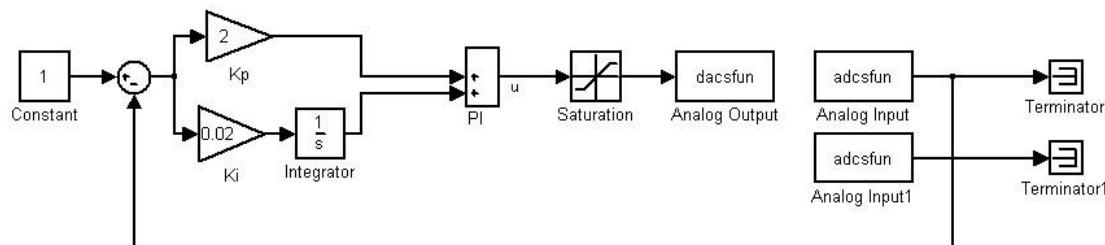


图 1 测试真实水箱系统特性算法框图

在图 1 的控制算法下测得数据如表 1，其中 constant 为给定值 left 和 right 都分为两栏，左栏为虚拟界面水位，右栏为真实水箱水位。经测定，只有当给定值 constant 的值大于 3.5，第二个水箱水位值才为正。当给定值为 7.5 时，第一个水箱水位刚好到达警戒值。因此，对给定值取值范围应该做一个限定：3.5-7.5。

表 1 真实水箱系统特性测试数据

constant	left(virtual/real)		right(virtual/real)	
3.5	3.5	10.5	1.7	0
4	4	14	2	2.5
4.5	4.5	18	2.4	5
5	5	21	2.8	7.5



5.5	5.5	24	3.2	9.8
6	6	27	3.5	12.2
6.5	6.5	30.5	3.9	14.6
7	7	34	4.2	17
7.5	7.5	37	4.6	19.4

根据表 1 数据, 经过处理和线性拟合, 可以得到真实水箱远程实验的算法框图, 如图 2 所示。

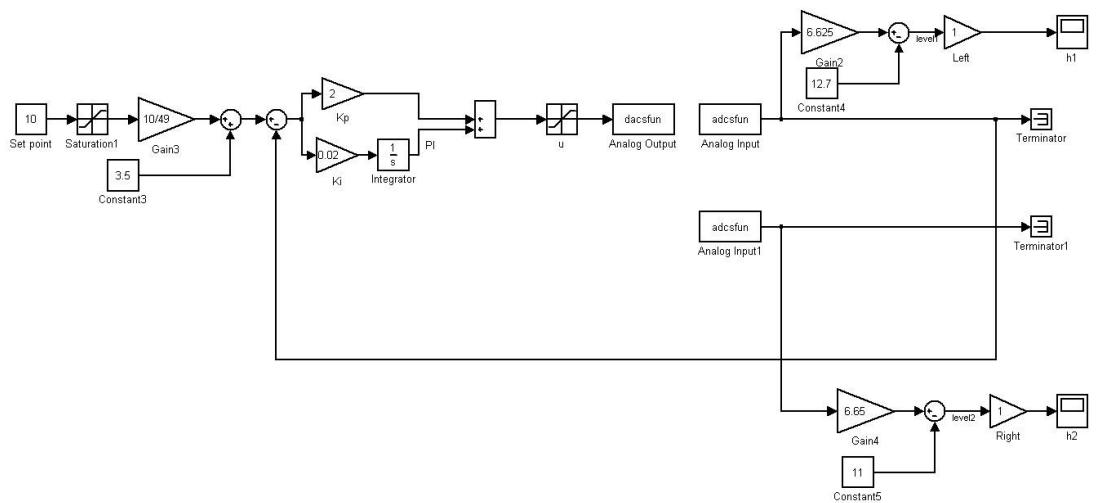


图 2 真实水箱远程实验算法

其中, 对给定值作了限幅和线性处理。处理过程和结果如下。

根据最终给定值范围为 3.5-7.5, 为了让给定值表示第二个水箱, 也就是被控液位的水箱水位一致, 将表 1 中的第一列和最后一列做线性处理。constant 为因变量, right(real) 为自变量, 其关系如图 3 所示。

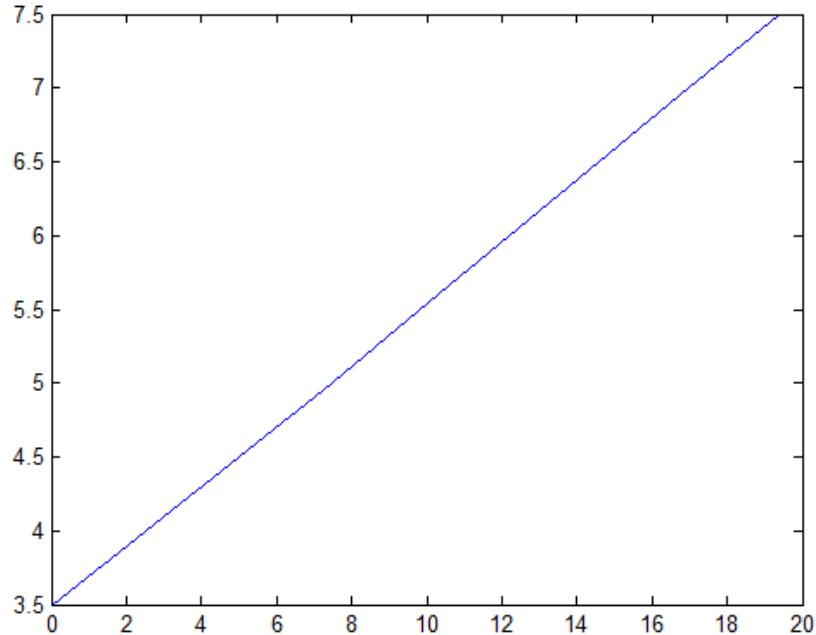


图 3 测试算法给定值与真实水箱被控水位关系

综合处理，得斜率为  $1/4.9$ ，截距为 3.5，其拟合关系和真实关系效果图如图 4 所示。

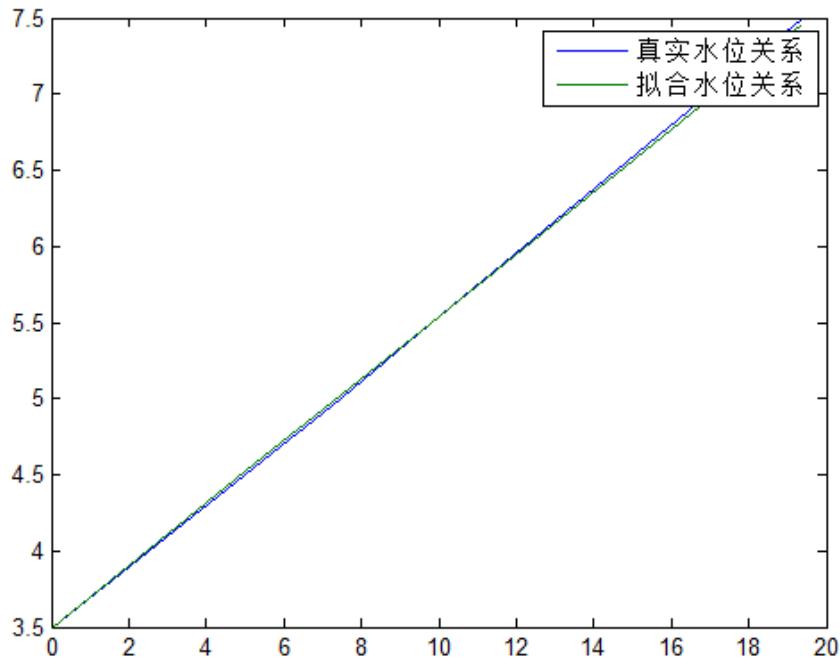




图 4 测试算法真实水位关系和拟合水位关系效果图

因此，在远程实验算法中，真实的给定值（图中的 Set point）可以变化范围限定在 0-19.4，这样既保证第二个水箱有水位值，同时保证第一个水箱不溢出，给定值还能与被控水箱的真实水位相联系，如图 5 所示。

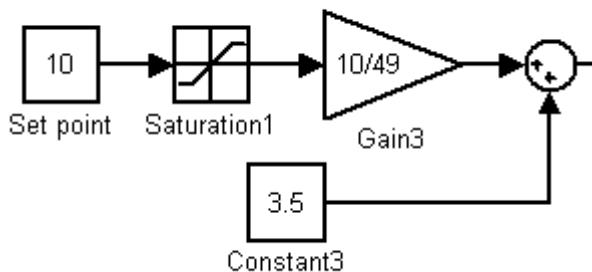


图 5 拟合处理后给定值表现方式

给定值和 PI 控制器组成了控制信号，将控制信息传给 Analog Output 模块，然后将反馈信号采集过来给终端，为了使三维监控界面水位与真实水位相一致，将表 1 中 2 和 3 以及 4 和 5 分别作为两组数据进行拟合，以虚拟水箱水位为自变量，真实水箱水位为因变量，第一第二个水箱效果分别如图 6、7 所示。

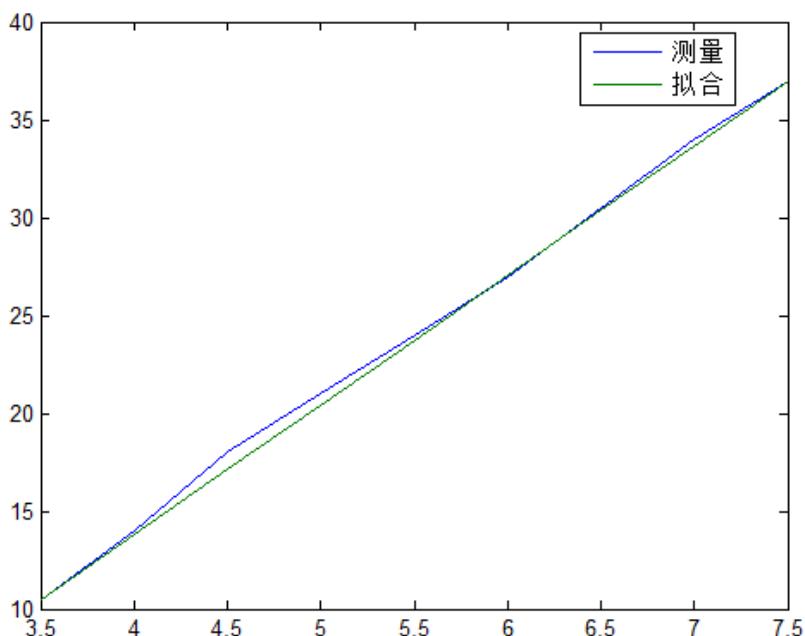


图 6 第一个水箱测量数据和拟合数据对比效果



图 2 中采用的控制算法为 PI 控制, 其中初始的 P 参数为 2, I 参数为 0.02(有很大的优化空间)。

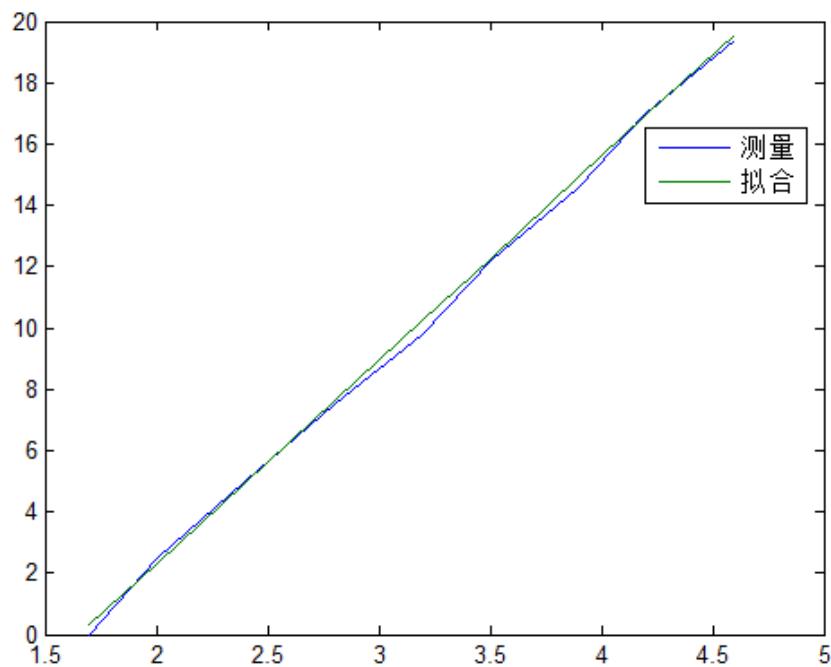


图 7 第二个水箱测量数据和拟合数据对比效果

则设计的三维虚拟界面显示水位的算法分别如图 8、9 所示。

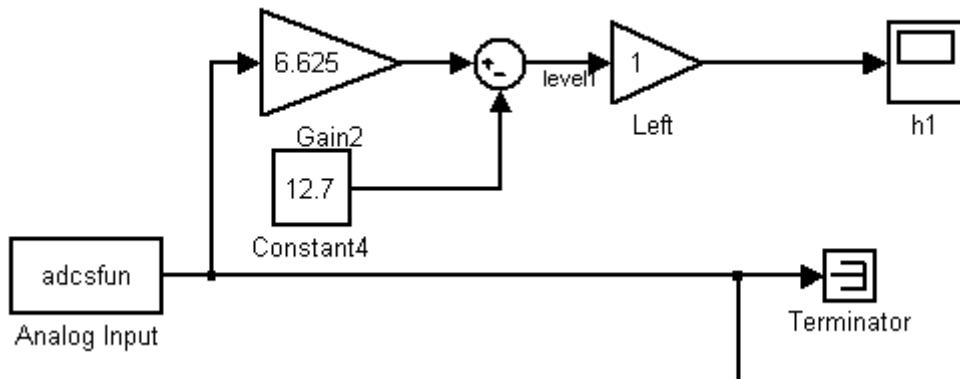


图 8 第一个水箱反馈信号与显示水位处理算法

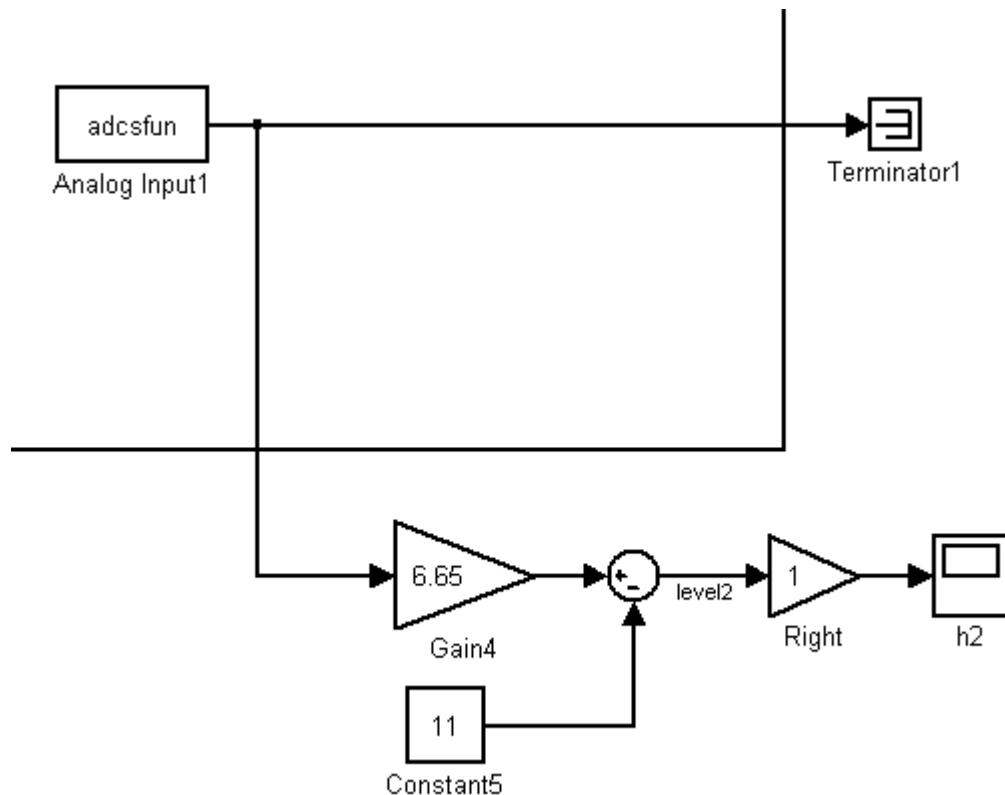


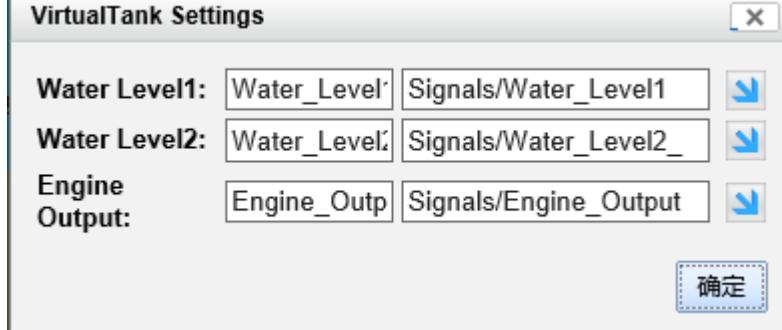
图 9 第一个水箱反馈信号与显示水位处理算法

至此，真实水箱远程实验的算法设计完成，如图 2 所示。

#### 四 实验步骤

1. 登录 NCSLab 3D 网站, [www.powersim.whu.edu.cn/ncslab](http://www.powersim.whu.edu.cn/ncslab)。
- 2、进入虚拟水箱水位控制的主界面，仔细阅读设备介绍和设备文档
- 3、在实验算法中选择 PI Control，下载到远程控制器中。
- 4、新建组态，如图 7 所示。建立如下的关联关系：

三维组件：



输入组件 1：



Numeric Input Settings

名称:	Set_point
下限值:	
上限值:	
数据精度:	3 个有效数字
参数:	Paramters/Set_point

输入组件 2:

Numeric Input Settings

名称:	Kp_
下限值:	
上限值:	
数据精度:	3 个有效数字
参数:	Paramters/Kp_

输入组件 3:

Numeric Input Settings

名称:	Kp_
下限值:	
上限值:	
数据精度:	3 个有效数字
参数:	Paramters/Kp_

趋势图组件:

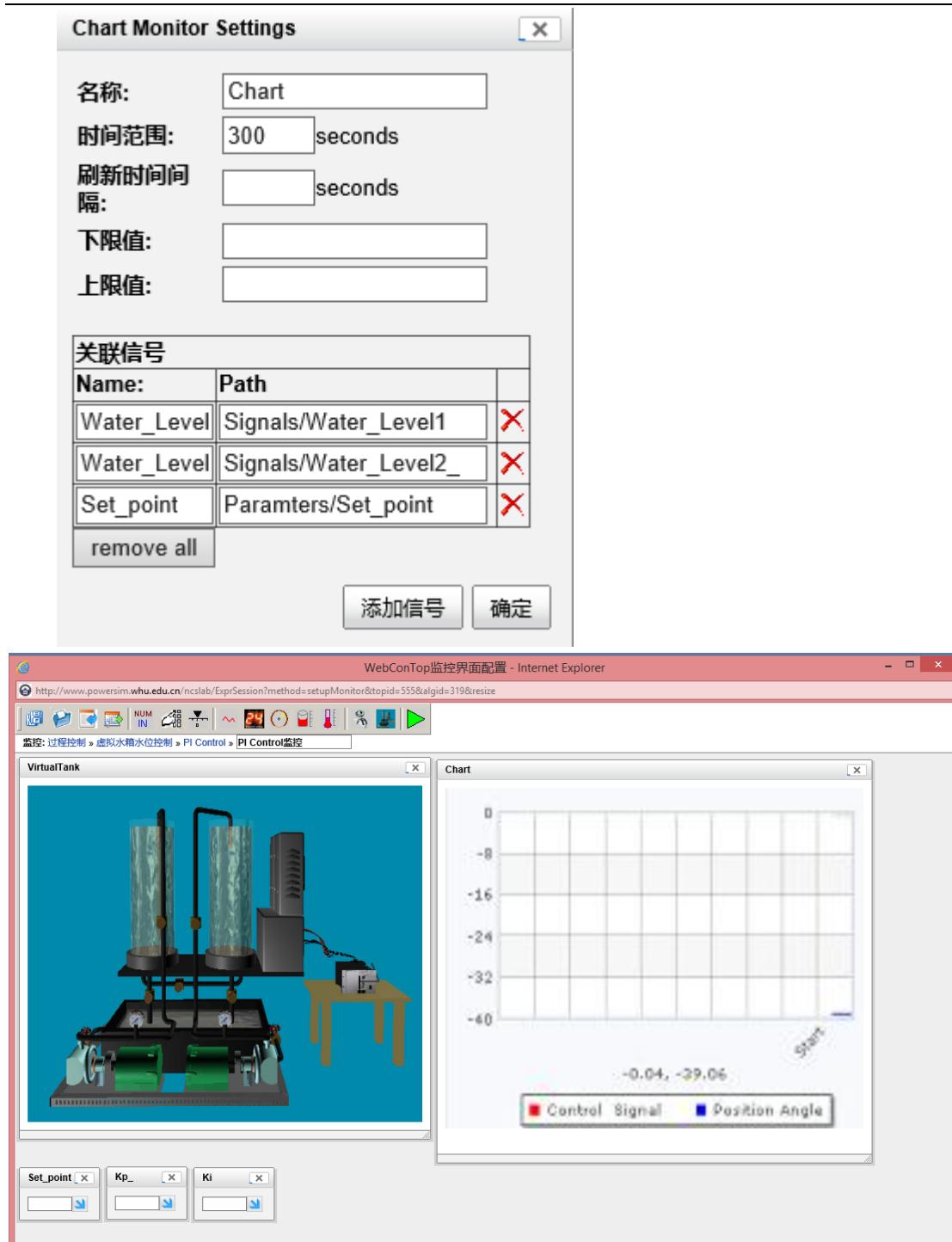


图 7. 组态图

- 5、点击运行按钮，开始进行虚拟实验。
- 6、改变水箱水位的设定值，观察 PI 控制器是如何将虚拟水箱的水位调整到新设定值的。如图 8 所示。



编写带有积分抗饱和的 PI 控制 S 函数

国家级电力生产过程  
虚拟仿真实验教学中心

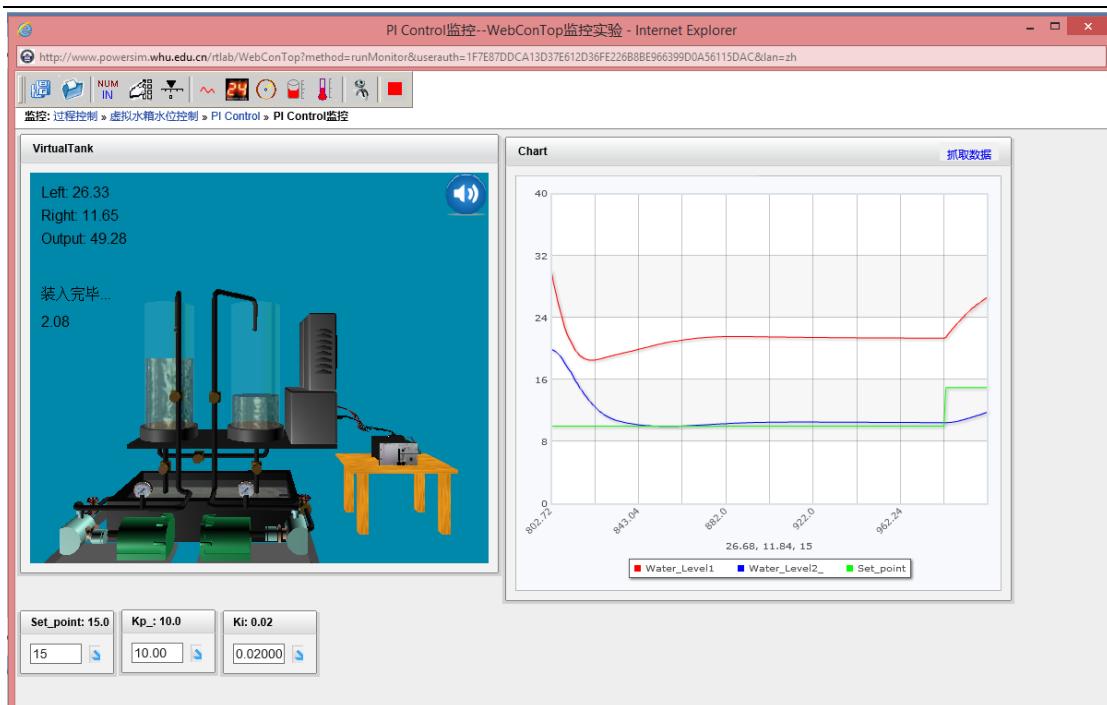


图 8. 虚拟水箱水位的控制过程

7. 通过修改 PI 参数，使得水位从 5 阶跃到 15 的调节时间最短
8. 将最佳的实验结果抓图，并量出调节时间。

## 五 思考题

- (1) 双容水箱水位控制的 PI 参数应该怎么选择，有什么特定的方法。
- (2) 双容水箱的数学模型是如何建立的。

## 六 实验报告

- (1) 需要将实验过程中的每一个关键步骤抓图粘贴到实验报告上。
- (2) 实验报告中要回答思考题提出的问题。



## 实验七 实体双容水箱水位远程 PI 控制实验

### 一 实验目的

- 1、了解双容水箱的结构、原理和数学模型
- 2、了解双容水箱的 PI 控制算法

### 二 实验内容

- 1、了解双容水箱的原理、结构和
- 2、了解双容水箱数学模型的建立方法
- 3、使用远程实验的方式，操作远程的双容水箱实验平台，调节 PI 控制器参数，优化控制效果。

### 三 实验原理

真实水箱控制算法。图 1 是实体水箱的照片。图 2 是为测试真实水箱系统特性而搭建的算法框图，其中限幅模块值从 0 到 inf 保证为正值。Analog Output 和 Analog Input 端口号（port number）均为 0，Analog Input 端口号为 1，采样时间为 0.04 秒。

图 2 中采用的控制算法为 PI 控制，其中初始的 P 参数为 2，I 参数为 0.02（有很大的优化空间）。

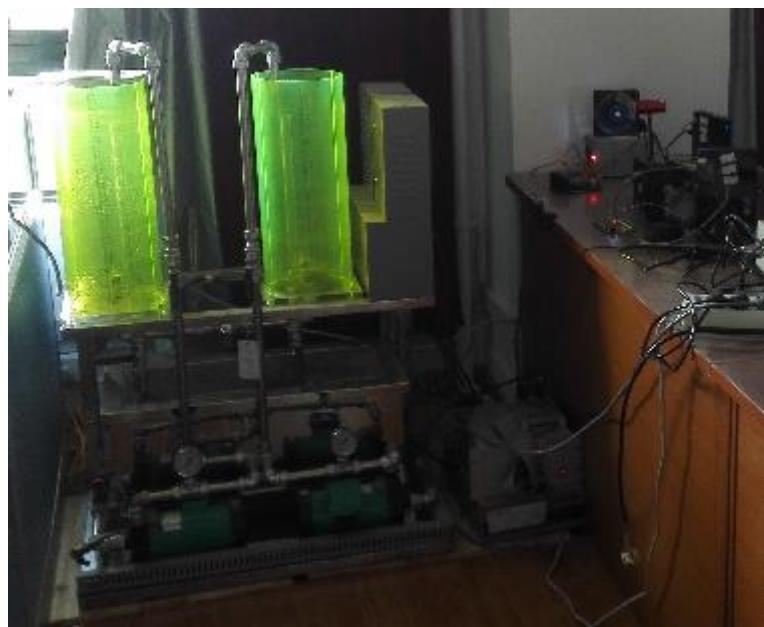


图 1 实体水箱

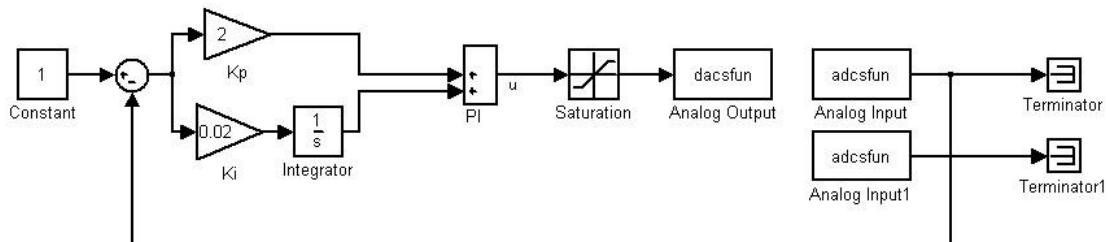


图 2 测试真实水箱系统特性算法框图

#### 四 实验步骤

1. 登录 NCSLab 3D 网站, [www.powersim.whu.edu.cn/ncslab/](http://www.powersim.whu.edu.cn/ncslab/), 其登录界面如图 3 所示, 输入用户名和密码即可登录。



图 3 NCSLab 3D 网站登录界面

2. 登录之后, 进入实验室列表界面, 网页左侧将一系列不同实验分门别类在不同分实验室下。选择过程控制, 然后选择里面的真实水箱水位控制实验。其显示界面如图 4 左图所示。点击“申请控制权”按钮, 效果如图 4 右图所示。



图 4 真实水箱水位控制初始界面

3. 申请控制权之后，就可以选择算法进行实验。点击“实验算法”链接，就可以进入到设备实验的可执行算法界面，如图 5 所示。



The screenshot shows the 'NCS-3DRS-Lab' software interface. On the left, there is a sidebar with a tree view of experimental algorithms, including 'RemoteControl1', 'RemoteControl2', 'RemoteControl3', 'RemoteControl4', 'Simulink', and 'WebConSim'. The main panel displays a table of executable algorithms:

ID	名称	类别	适用平台	算法步长	上次实验时间	操作
347	RemoteControl1	本地仿真控制	ARM9	0.04 秒	2015-04-11,11:06	
362	RemoteControl	本地仿真控制	ARM9	0.04 秒	2015-04-11,10:52	
360	RemoteControl3	本地仿真控制	ARM9	0.04 秒	2015-04-11,10:34	
358	RemoteControl2	本地仿真控制	ARM9	0.04 秒	2015-04-11,10:05	

At the bottom of the main panel, it says 'CopyRight© ncslab.net All Rights Reserved'.

图 5 设备实验的可执行算法界面

4. 选择好相应的算法后，点击“开始实验”按钮，就会自动下载算法，启动算法的实验监控组态列表，如图 6 所示。如果之前之前有人进行过实验，那么“我的监控列表”里面就会有记录，可以点击“我的监控”，直接进行实验的监控，也可以点击“新建监控组态”，来搭建自己的监控界面。



编写带有积分抗饱和的 PI 控制 S 函数

国家级电力生产过程  
虚拟仿真实验教学中心

图 6 实验监控组态列表

5. 点击“新建监控组态”链接后，就进入了 Webcontop 实验监控配置界面，如图 7 所示。点击工具栏的不同按钮，就可以选择不同的工具进行监控。本实验点

击实验设备对应的按钮 ，将出现页面缩放到适宜的大小和位置，如图 8 所示。



图 7 监控配置界面工具栏

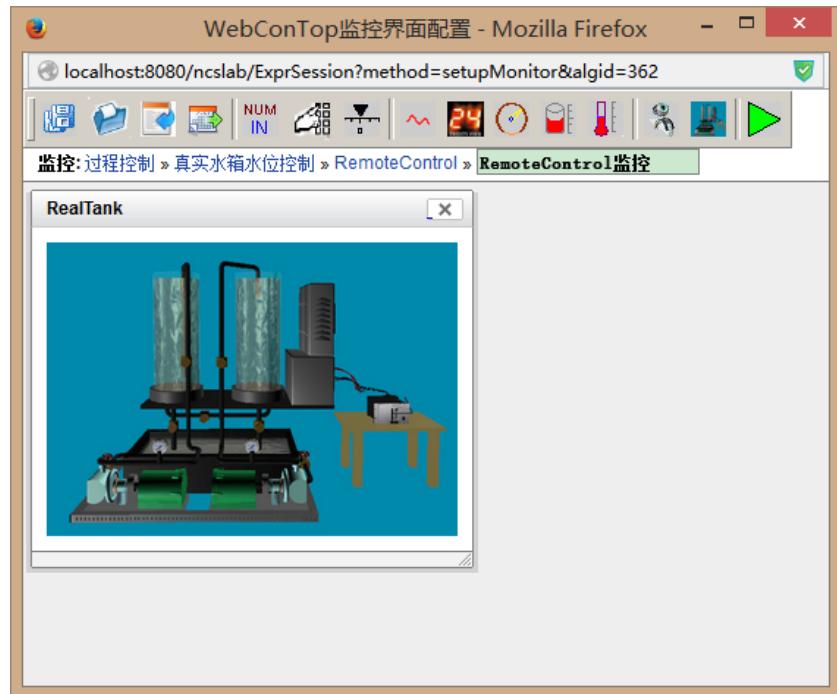


图 8 点击设备后配置界面

6. 鼠标左键双击界面，根据弹出的框选择好相应的信号和参数，信号用来监测和查看，参数用来设置和控制，双容水箱的信号和参数配置界面如图 9 所示。

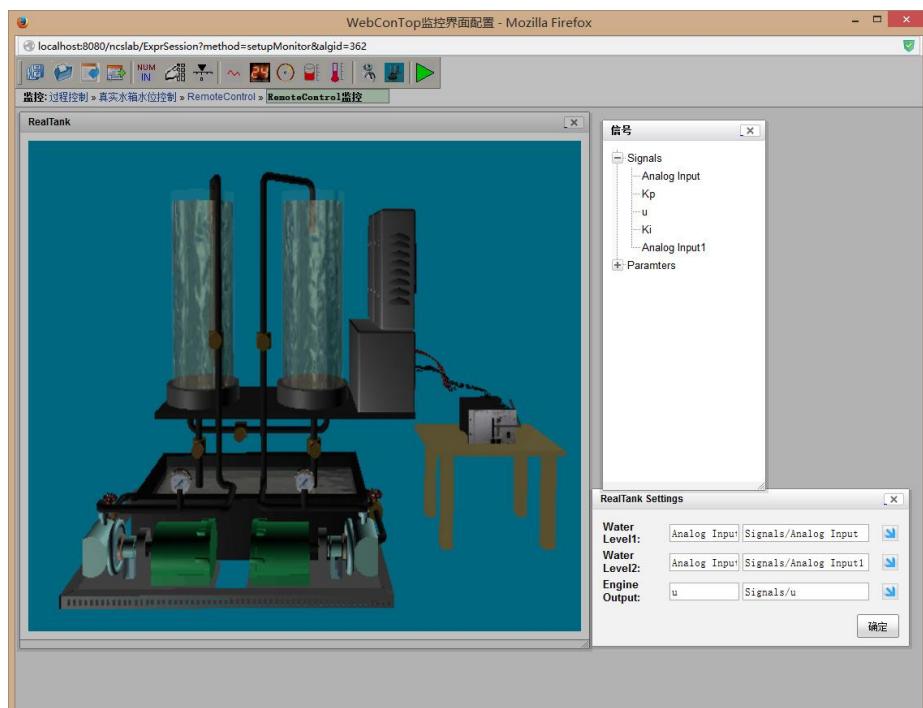


图 9 双容水箱信号和参数配置界面



7. 除了虚拟界面之外，实时视频，水位曲线的实时变化（点击 ），PI 参数、  
给定值的设定（点击 ）都需要在配置界面完成，并选择好对应的信号和参数，设置好相关的取值范围，如图 10 所示。

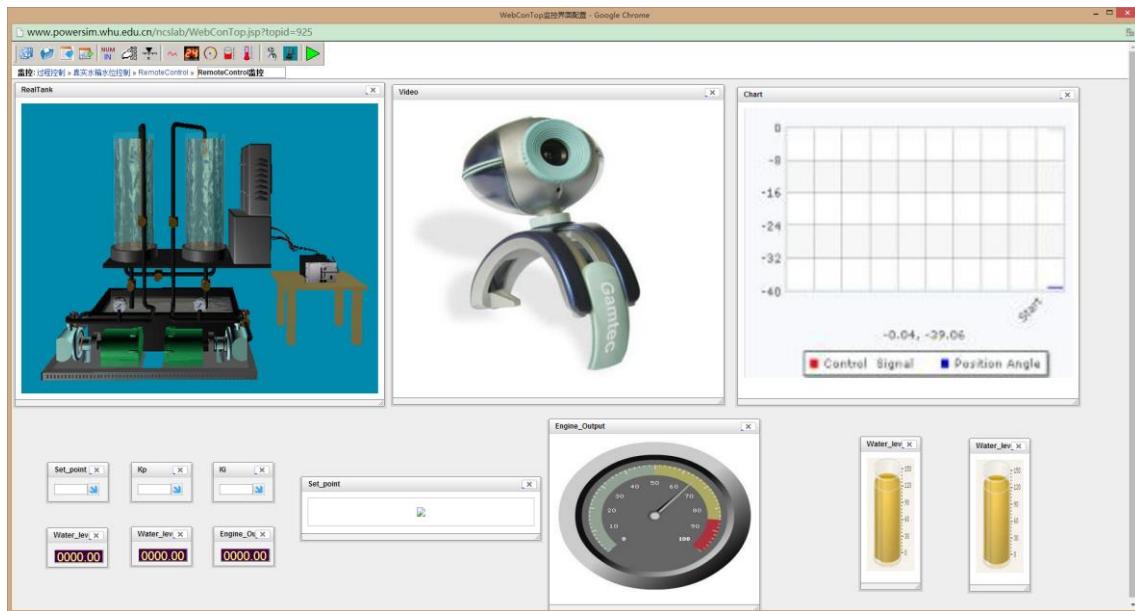


图 10 参数配置完成界面

8. 点击 ，就进入了监控界面，如图 11 所示，界面中实时显示了水位三维图，包含了水位以及 PI 输出的变化值，两个水箱水位变化实时曲线以及给定值和比例系数以及积分系数的设定框，这三个参数可以改变来调节水位变化情况。

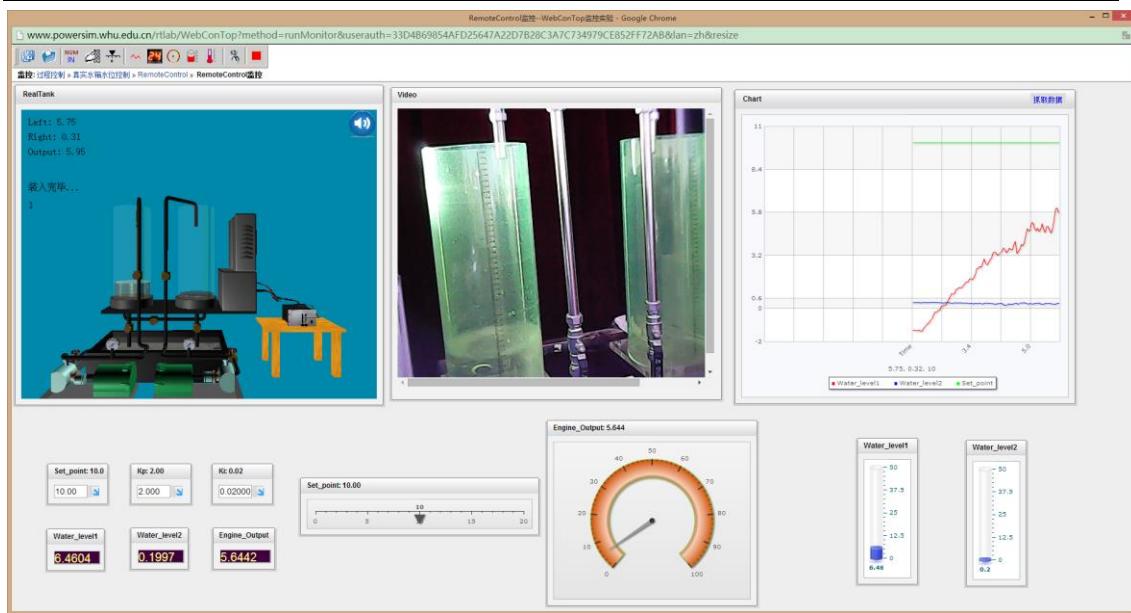


图 11 水箱水位监控界面

9. 通过修改 PI 参数，使得水位从 5 阶跃到 10 的调节时间最短

10. 将最佳的实验结果抓图，并量出调节时间。

## 五 思考题

(1) 比较双容水箱和风扇这两个被控对象，在特性上有什么共同点，有什么区别。

(2) 双容水箱水位控制的 PI 参数应该怎么选择，有什么特定的方法。

## 六 实验报告

(1) 需要将实验过程中的每一个关键步骤抓图粘贴到实验报告上。

(2) 实验报告中要回答思考题提出的问题。



## 实验八 虚拟球杆系统的 LQR 控制

### 一 实验目的

- 1、了解球杆系统的结构、原理和数学模型
- 2、了解球杆系统的 LQR 控制算法

### 二 实验内容

- 1、了解球杆系统的原理、结构
- 2、了解球杆系统数学模型的建立方法
- 3、使用远程虚拟实验的方式，操作虚拟的球杆控制平台，实现球杆系统的 LQR 控制。

### 三 实验原理

#### 1、球杆系统的原理、结构

球杆系统主要由以下几部分组成，如图 1 和图 2 所示。包括底座、小球、横杆、减速皮带轮、支撑部分、马达等。小球可以在横杆上自由的滚动，横杆的一端通过转轴固定，另一端可以上下转动，通过控制直流伺服电机的位置，带动皮带轮转动，通过传动机构就可以控制横杆的倾斜角。直流伺服电机带有增量式编码器，可以检测电机的实际位置，即  $\theta$  角度已检测，在横杆上的凹槽内，有一线性的传感器用于检测小球的实际位置，两个实际位置的信号都被传送给控制系统，构成一个闭环反馈系统。当带轮转动角度  $\theta$ ，横杆的转动角度为  $\alpha$ ，当横杆偏离水平的平衡位置后，在重力作用下，小球开始沿横杆滚动。我们的目的是设计一个控制器，通过控制电机的转动，使小球稳定在横杆上的某一平衡位置。

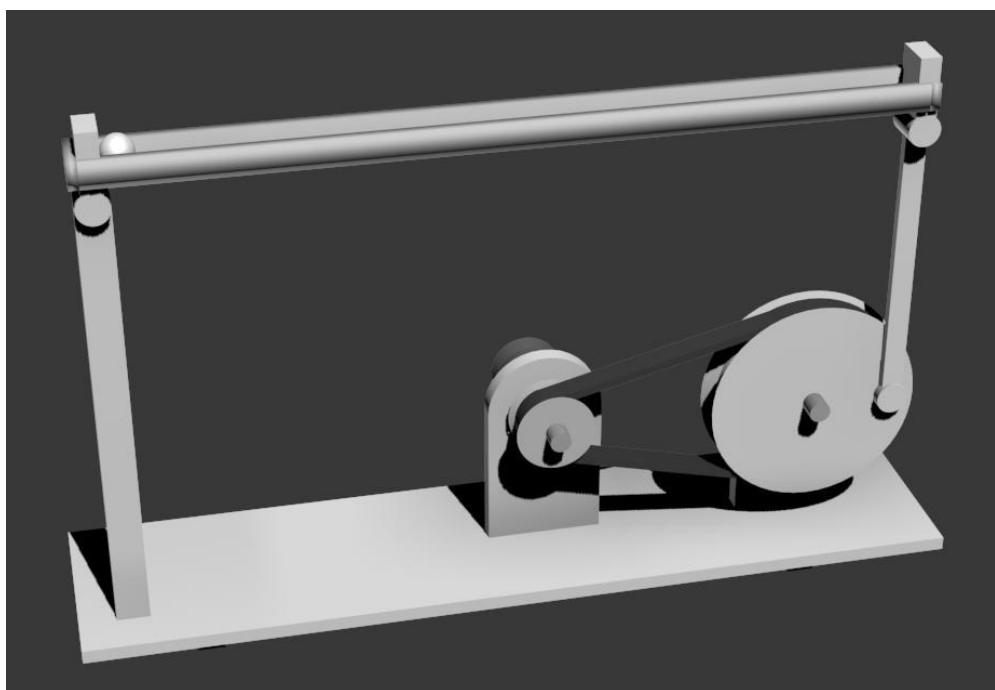


图 1 球杆系统示意图 a

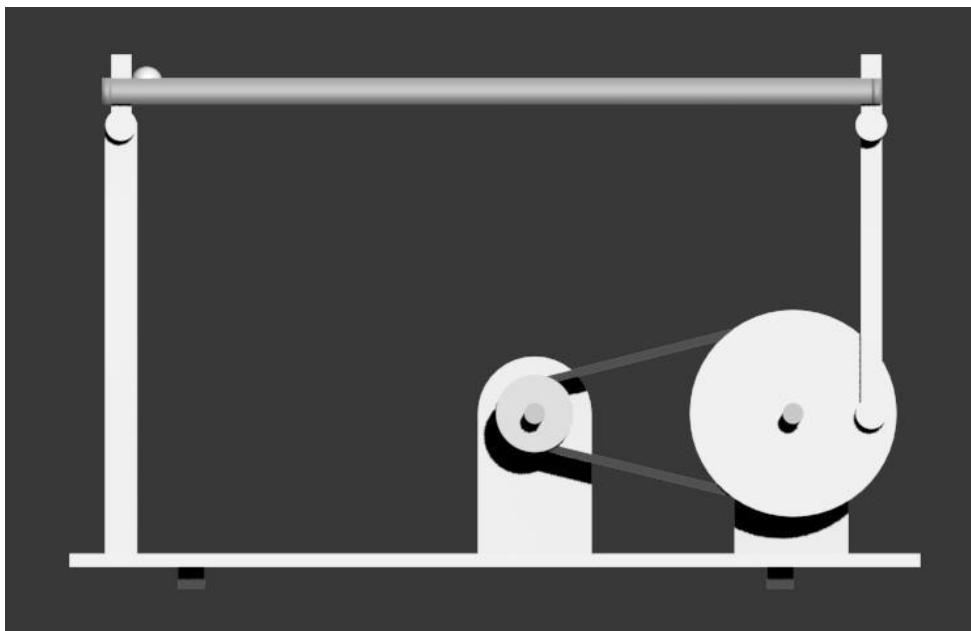


图 2 球杆系统示意图 b

## 2、球杆系统的数学模型

球杆系统的机械系统原理图如图 3 所示。

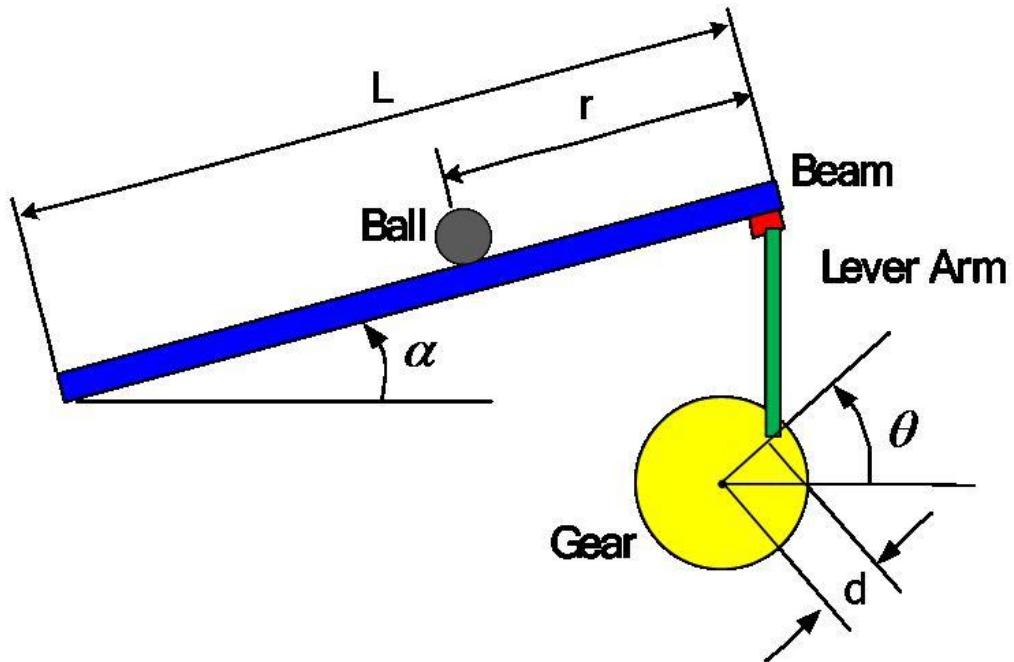


图 3 球杆系统机械结构



连线(连杆和同步带轮的连接点与齿轮中心的连线)和水平线的夹角为  $\theta$  ( $\theta$  的角度存在一定的限制, 在最小和最大的范围之间), 连杆和齿轮的连接点与齿轮中心的距离为  $d$ , 横杆的长度为  $L$ , 于是, 横杆的倾斜角  $\alpha$  和  $\theta$  之间的有如下的数学关系:

$$\alpha = \frac{d}{L} \theta$$

如前所述, 角度  $\theta$  和电机轴之间存在一个减速比  $n=4$  的同步带, 控制器设计的任务是通过调整齿轮的角度  $\theta$ , 使得小球在某一位置平衡。小球的重力、惯量以及离心力等动力学关系, 可以建立小球在横杆上滚动的动力学方程。若设定如下状态:

$$\dot{x}_1 = r, x_2 = \dot{r}$$

其中

$x_1$  —— 小球的位移

$x_2$  —— 小球的速度

输入  $u$  代表齿轮输入转角, 经过线性化动力学方程可建立如下的状态空间方程

$$\dot{X} = AX + Bu$$

$$Y = CX$$

其中

$$X = [x_1 \quad x_2]^T$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

带入参数得参数矩阵为

$$B = \begin{bmatrix} 0 \\ -0.21 \end{bmatrix}$$

### 3、演示算法: LQR 控制方法



在 Command Window 中输入

```
A=[0 1;0 0];  
B=[0;-0.21];  
Q=[10 0;0 1];  
R=1;  
K=lqr(A,B,Q,R)
```

得反馈矩阵  $K = -10.0000 \quad -9.8101$

建立如图 4 的 Simulink 控制框图，其中球杆系统的模型部分为根据非线性状态空间方程用 S 函数编写的模块，控制模块为上述 LQR 方法建立的控制器。分别设定小球期望的位置为 50cm，得到如图 5 的输出曲线。

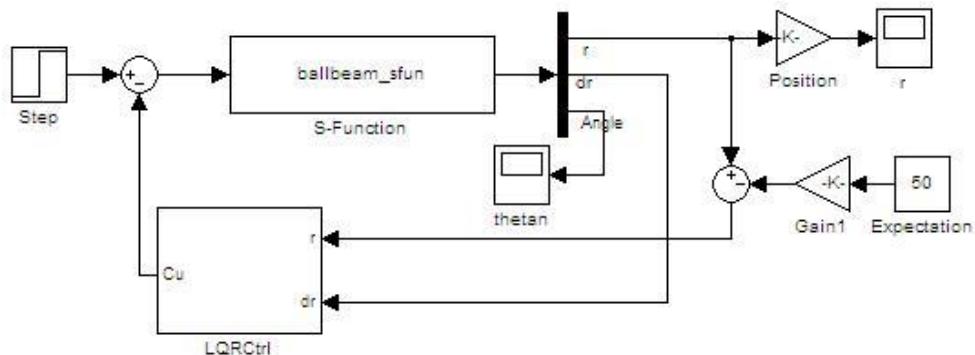


图 4 球杆系统 Simulink 框图

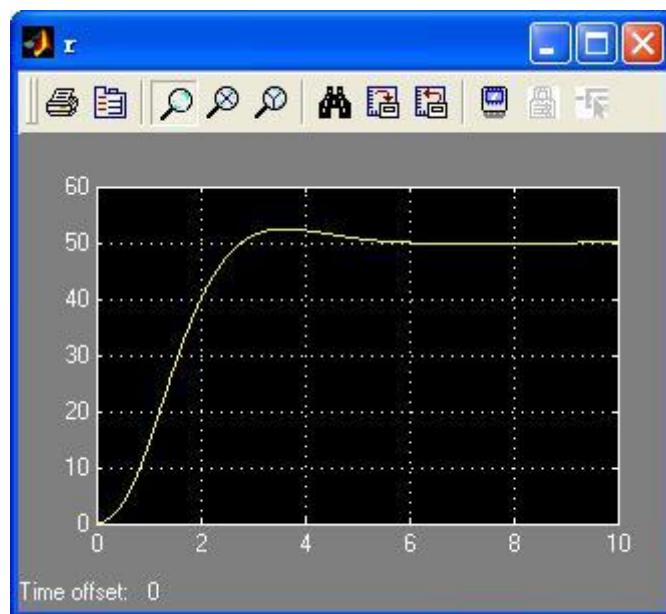


图 4 球杆系统相应曲线



### 三 实验步骤

1、登陆 NCSLab，在复杂系统实验室中找到球杆系统虚拟实验台（球杆系统 mini），如图 6 所示。

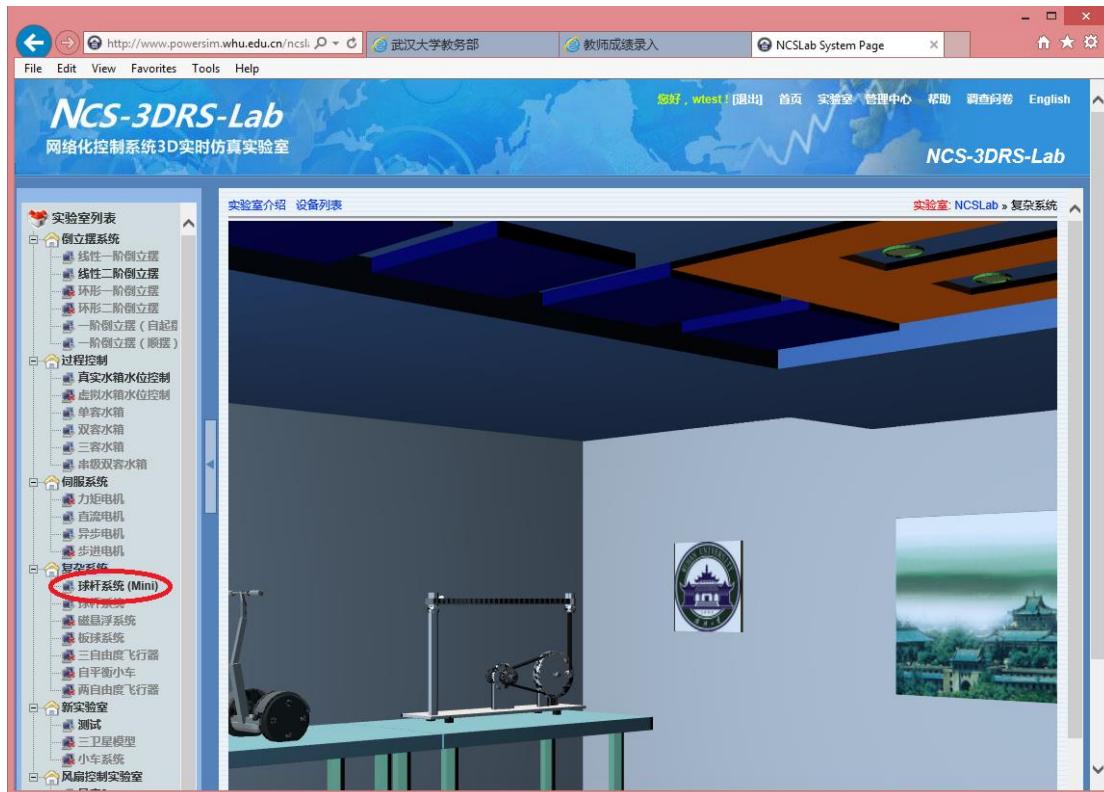


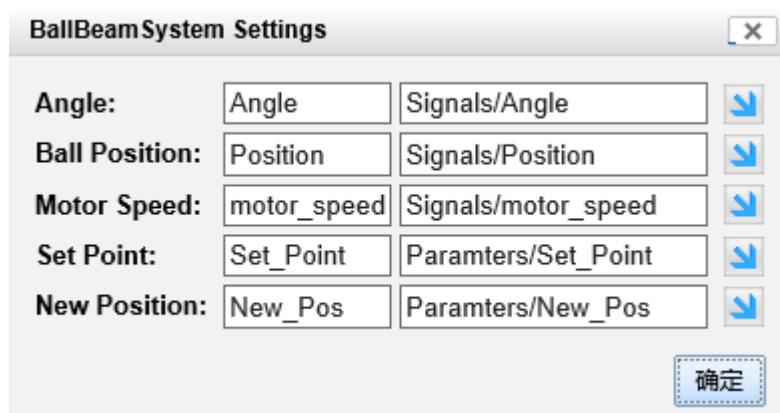
图 6. 找到球杆系统

2、进入球杆系统的主界面，仔细阅读设备介绍和设备文档

3、在实验算法中选择 LQ Control，下载到远程控制器中。

4、新建组态，如图 7 所示。建立如下的关联关系：

三维组件：





输入组件:

Numeric Input Settings X

名称:	Set_Point
下限值:	<input type="text"/>
上限值:	<input type="text"/>
数据精度:	3 个有效数字
参数:	Paramters/Set_Point

选择参数 确定

趋势图组件:

Chart Monitor Settings X

名称:	Chart
时间范围:	10 seconds
刷新时间间隔:	<input type="text"/> seconds
下限值:	<input type="text"/>
上限值:	<input type="text"/>

**关联信号**

Name:	Path	X
Position	Signals/Position	X
Angle	Signals/Angle	X

remove all

添加信号 确定

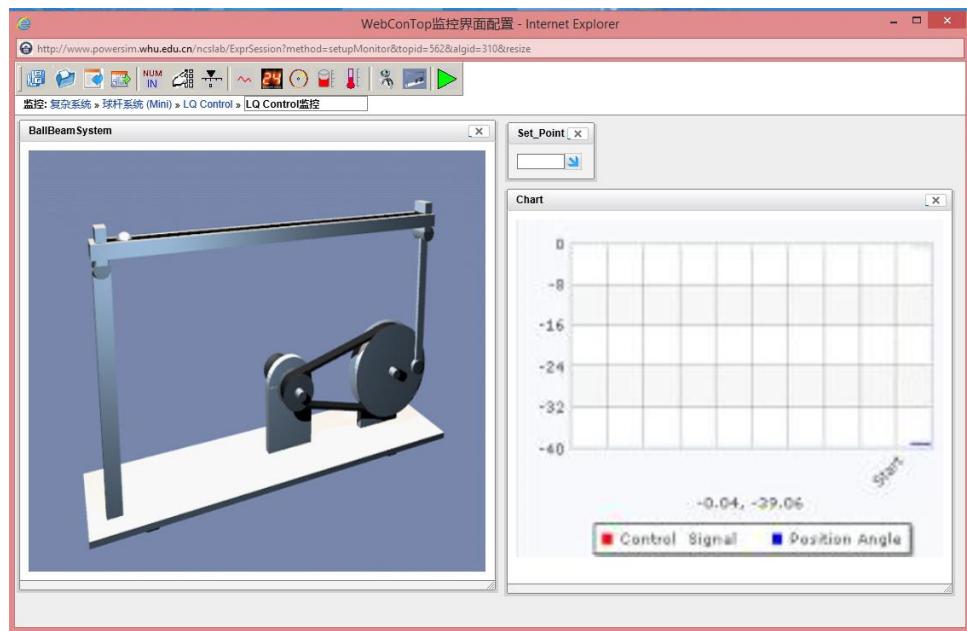


图 7. 组态图

- 5、点击运行按钮，开始进行虚拟实验。
- 6、改变小球位置的设定值，观察虚拟的球杆系统的控制过程是怎样的。如图 8 所示。

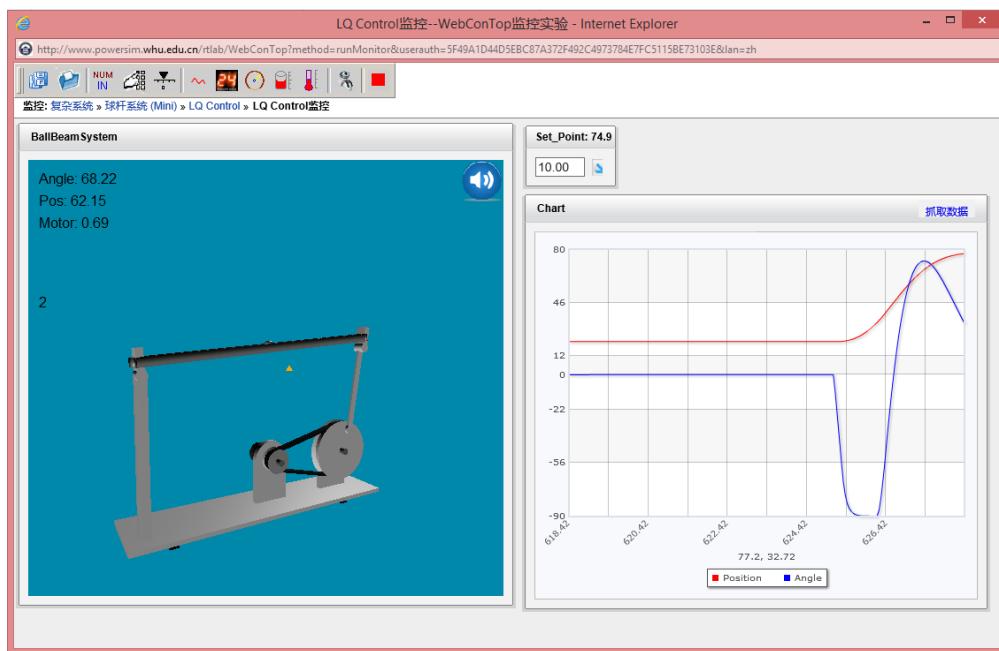


图 8. 虚拟球杆系统的控制过程

## 五 思考题

- (1) 解释球杆系统的原理。



---

(2) 分析 Simulink 框图，解释球杆系统 LQR 算法的细节。

## 六 实验报告

- (1) 需要将实验过程中的每一个关键步骤抓图粘贴到实验报告上。
- (2) 实验报告中要回答思考题提出的问题。



## 实验九 虚拟球杆系统的 PD 控制

### 一 实验目的

- 1、了解球杆系统的结构、原理和数学模型
- 2、了解球杆系统的 PD 控制算法

### 二 实验内容

- 1、了解球杆系统的原理、结构
- 2、了解球杆系统数学模型的建立方法
- 3、使用远程虚拟实验的方式，操作虚拟的球杆控制平台，实现球杆系统的 PD 控制。

### 三 实验原理

#### 1、球杆系统的原理、结构

球杆系统主要由以下几部分组成，如图 1 和图 2 所示。包括底座、小球、横杆、减速皮带轮、支撑部分、马达等。小球可以在横杆上自由的滚动，横杆的一端通过转轴固定，另一端可以上下转动，通过控制直流伺服电机的位置，带动皮带轮转动，通过传动机构就可以控制横杆的倾斜角。直流伺服电机带有增量式编码器，可以检测电机的实际位置，即  $\theta$  角度已检测，在横杆上的凹槽内，有一线性的传感器用于检测小球的实际位置，两个实际位置的信号都被传送给控制系统，构成一个闭环反馈系统。当带轮转动角度  $\theta$ ，横杆的转动角度为  $\alpha$ ，当横杆偏离水平的平衡位置后，在重力作用下，小球开始沿横杆滚动。我们的目的是设计一个控制器，通过控制电机的转动，使小球稳定在横杆上的某一平衡位置。

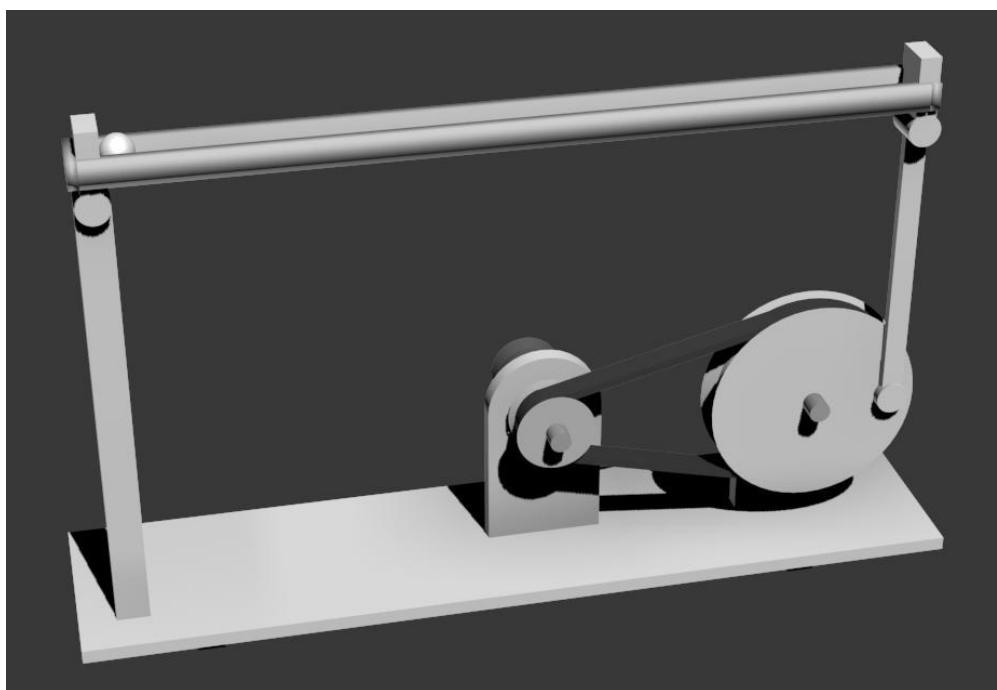


图 1 球杆系统示意图 a

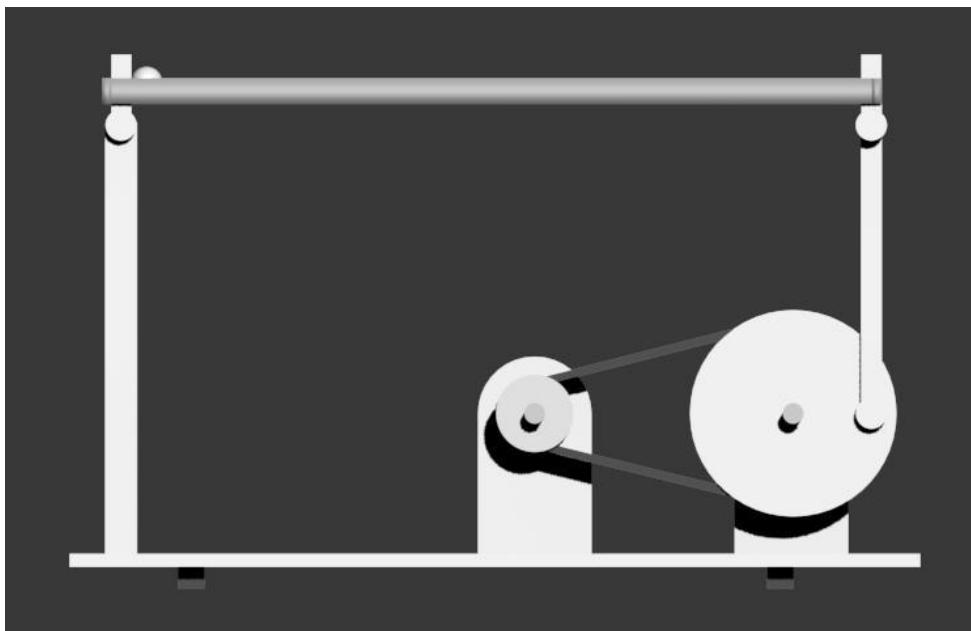


图 2 球杆系统示意图 b

## 2、球杆系统的数学模型

球杆系统的机械系统原理图如图 3 所示。

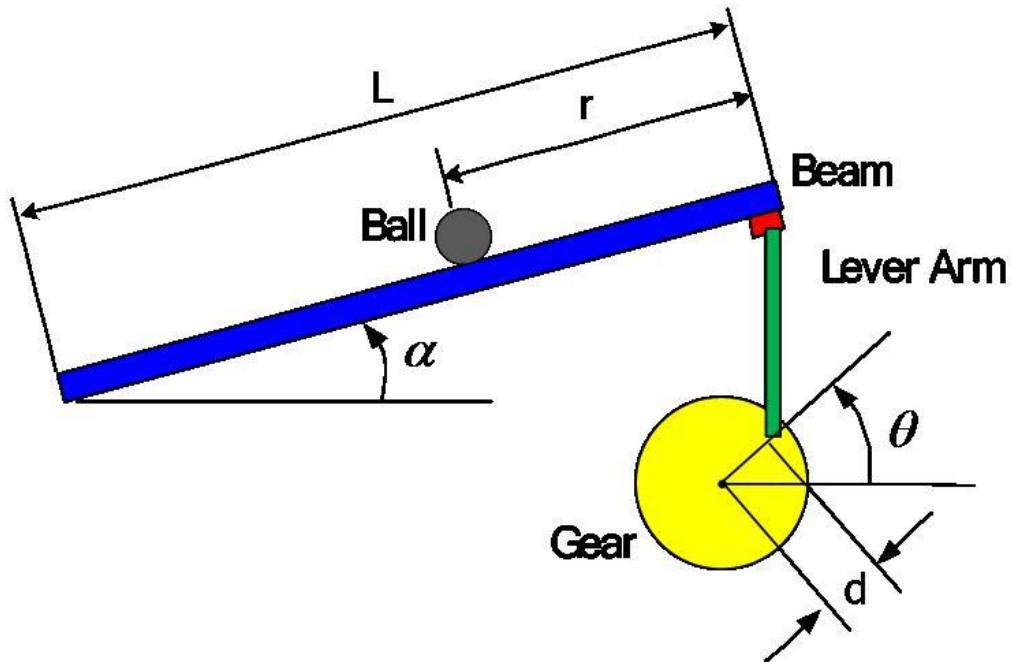


图 3 球杆系统机械结构



连线(连杆和同步带轮的连接点与齿轮中心的连线)和水平线的夹角为  $\theta$  ( $\theta$  的角度存在一定的限制, 在最小和最大的范围之间), 连杆和齿轮的连接点与齿轮中心的距离为  $d$ , 横杆的长度为  $L$ , 于是, 横杆的倾斜角  $\alpha$  和  $\theta$  之间的有如下的数学关系:

$$\alpha = \frac{d}{L} \theta$$

如前所述, 角度  $\theta$  和电机轴之间存在一个减速比  $n=4$  的同步带, 控制器设计的任务是通过调整齿轮的角度  $\theta$ , 使得小球在某一位置平衡。小球的重力、惯量以及离心力等动力学关系, 可以建立小球在横杆上滚动的动力学方程。若设定如下状态:

$$\dot{x}_1 = r, x_2 = \dot{r}$$

其中

$x_1$  —— 小球的位移

$x_2$  —— 小球的速度

输入  $u$  代表齿轮输入转角, 经过线性化动力学方程可建立如下的状态空间方程

$$\dot{X} = AX + Bu$$

$$Y = CX$$

其中

$$X = [x_1 \quad x_2]^T$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

带入参数得参数矩阵为

$$B = \begin{bmatrix} 0 \\ -0.21 \end{bmatrix}$$

### 3、演示算法: LQR 控制方法



建立如图 4 的 Simulink 控制框图, 其中 PD 控制器中 P 参数和 D 参数均为 0.1。分别设定小球期望的位置为 10cm, 得到如图 5 的输出曲线。

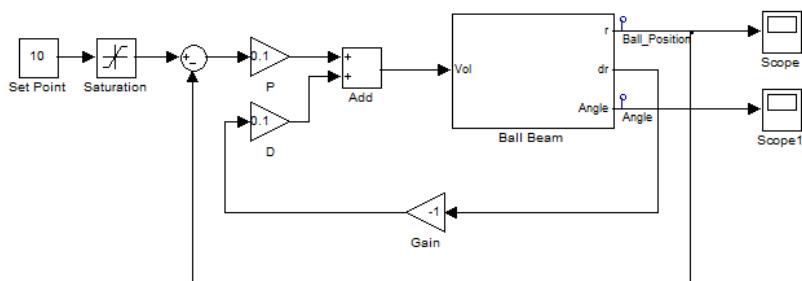


图 4 球杆系统 Simulink 框图

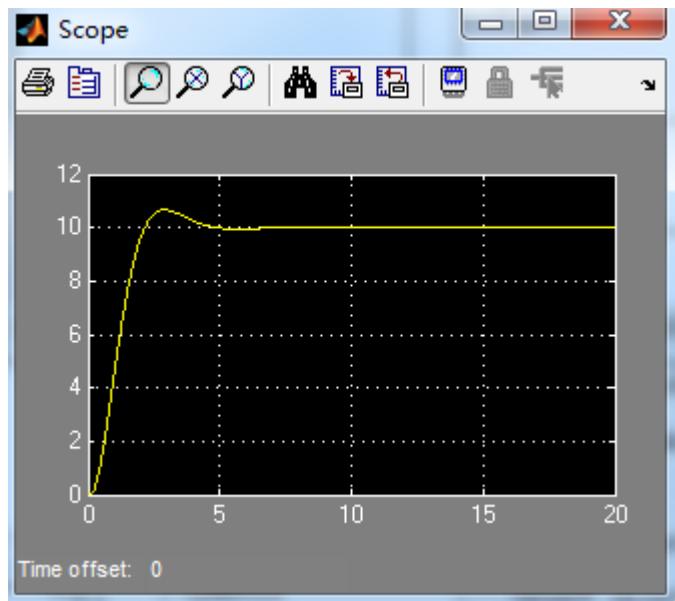


图 4 球杆系统相应曲线

### 三 实验步骤

1、登陆 NCSLab, 在复杂系统实验室中找到球杆系统虚拟实验台 (球杆系统 mini), 如图 6 所示。

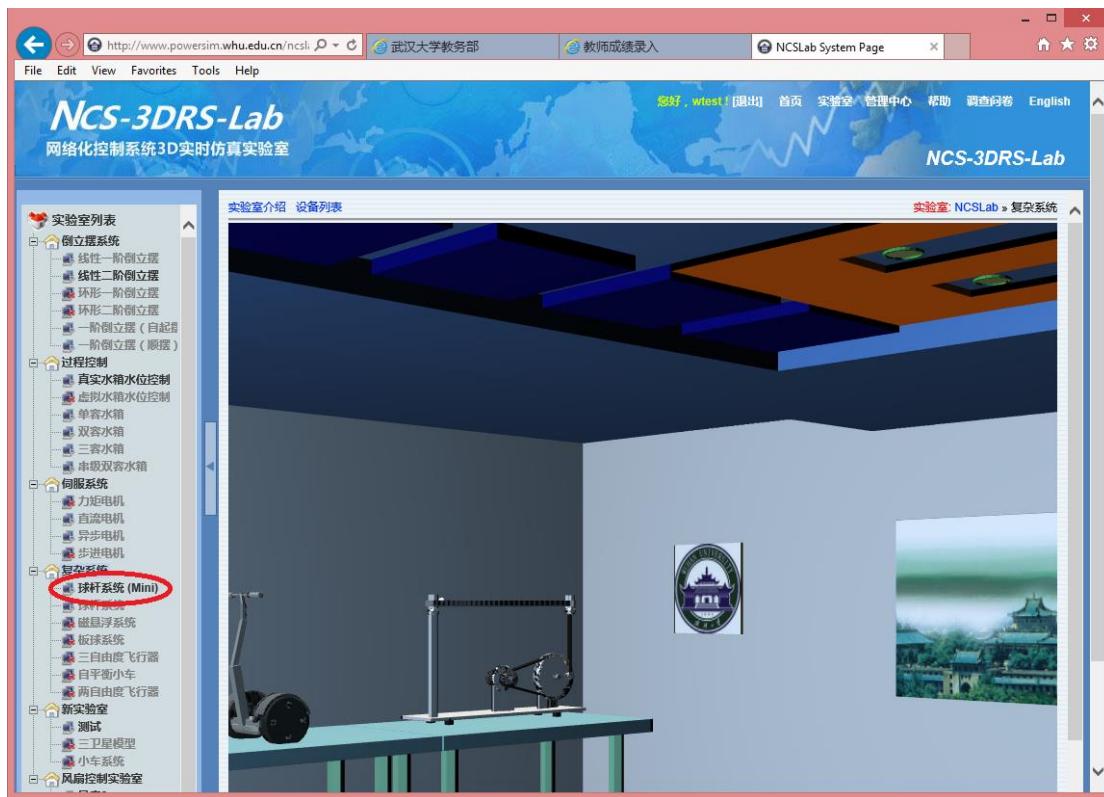
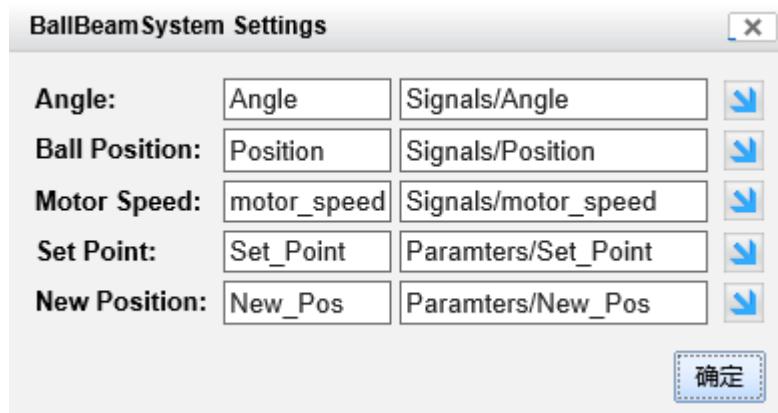


图 6. 找到球杆系统

- 2、进入球杆系统的主界面，仔细阅读设备介绍和设备文档
- 3、在实验算法中选择 PD Control，下载到远程控制器中。
- 4、新建组态，如图 7 所示。建立如下的关联关系：

三维组件：



输入组件 1：



Numeric Input Settings

名称:	Set_Point
下限值:	
上限值:	
数据精度:	3 个有效数字
参数:	Paramters/Set_Point

输入组件 2:

Numeric Input Settings

名称:	P
下限值:	
上限值:	
数据精度:	3 个有效数字
参数:	Paramters/P

输入组件 3:

Numeric Input Settings

名称:	D
下限值:	
上限值:	
数据精度:	3 个有效数字
参数:	Paramters/D

趋势图组件:

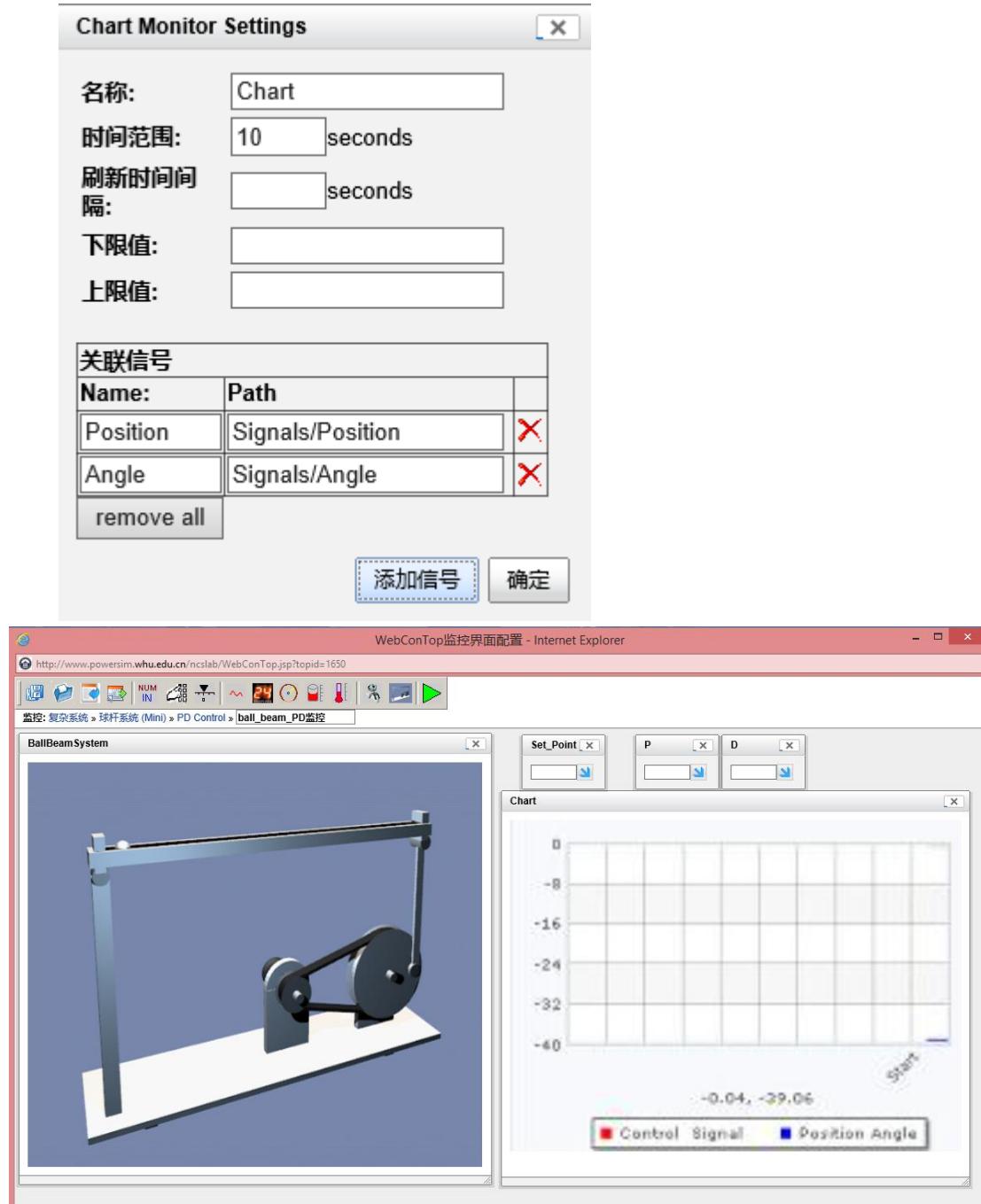


图 7. 组态图

- 5、点击运行按钮，开始进行虚拟实验。
- 6、改变小球位置的设定值，观察虚拟的球杆系统的控制过程是怎样的。如图 8 所示。
- 7、增大或者减小微分项的参数，改变小球位置的设定值，观察控制效果是怎么变化的。

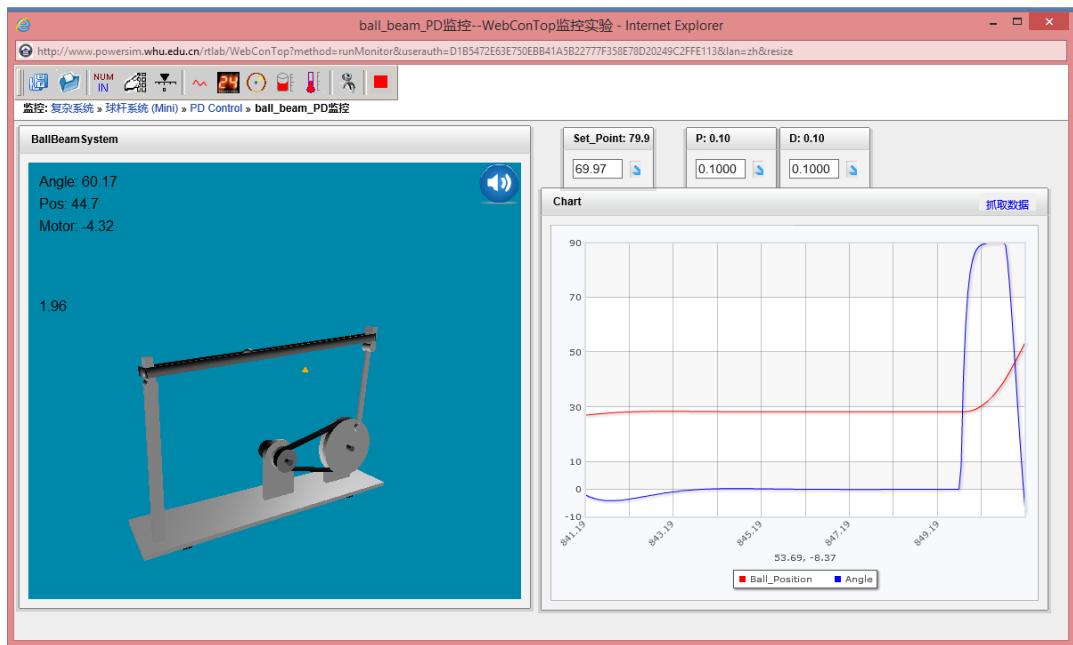


图 8. 虚拟球杆系统的控制过程

8、在不同的微分项取值情况下，将小球的设定值从 10 调整到 50，观察超调量和调节时间，完成下表，分析微分对控制效果的影响。

微分参数	超调量	调节时间
0.3		
0.2		
0.1		
0		

## 五 思考题

- (1) 解释球杆系统的原理。
- (2) 分析 Simulink 框图，解释球杆系统 PD 算法的细节。
- (3) 结合本实验的结果，阐明微分作用的超前调节在控制系统中有什么作用。

## 六 实验报告

- (1) 需要将实验过程中的每一个关键步骤抓图粘贴到实验报告上。
- (2) 实验报告中要回答思考题提出的问题。



## 实验十 虚拟线性一阶倒立摆的 LQR 控制

### 一 实验目的

- 1、了解线性一阶倒立摆的结构、原理和数学模型
- 2、了解线性一阶倒立摆的 LQR 控制算法

### 二 实验内容

- 1、了解线性一阶倒立摆的原理、结构
- 2、了解线性一阶倒立摆数学模型的建立方法
- 3、使用远程虚拟实验的方式，操作虚拟的线性一阶倒立摆控制平台，实现 LQR 控制。

### 三 实验原理

#### 1、线性一阶倒立摆的原理

线性一阶倒立摆系统主要由以下几部分组成，如图 1 所示。其机械本体主要包括底座（导轨）、小车、驱动小车的交流伺服电机、同步皮带、摆杆、限位开关及光电码盘等。通过控制交流伺服电机，带动皮带转动，在皮带的带动下小车可以在导轨上运动从而控制摆杆的运动状态。交流伺服电机带有光电式脉冲编码盘，根据脉冲数目可得出工作轴的回转角度，由传动比换算出小车线性位移。在小车的运动导轨上有用于检测小车位置的传感器，小车位置的信号被传送给控制系统，通过控制算法计算出控制量控制电机，从而控制小车的位置，使摆杆垂直于水平面。我们的目的是设计一个控制器，通过控制电机的转动，使摆杆稳定在垂直于水平面的位置。

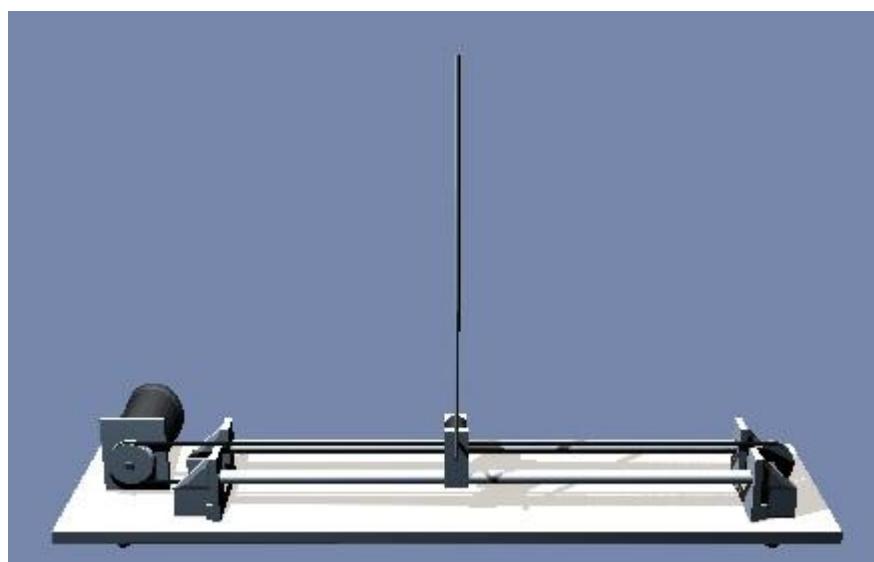


图 1 线性一阶倒立摆示意图



## 2、线性一阶倒立摆的数学模型

若忽略空气阻力和各种摩擦力之后,可将线性一阶倒立摆系统抽象成小车和质量均匀的摆杆组成的系统,如图 2 所示其中符号意为:

F — 小车所受力

l — 摆杆转动轴心到摆杆质心的长度

$\phi$  — 摆杆与垂直向上方向的夹角

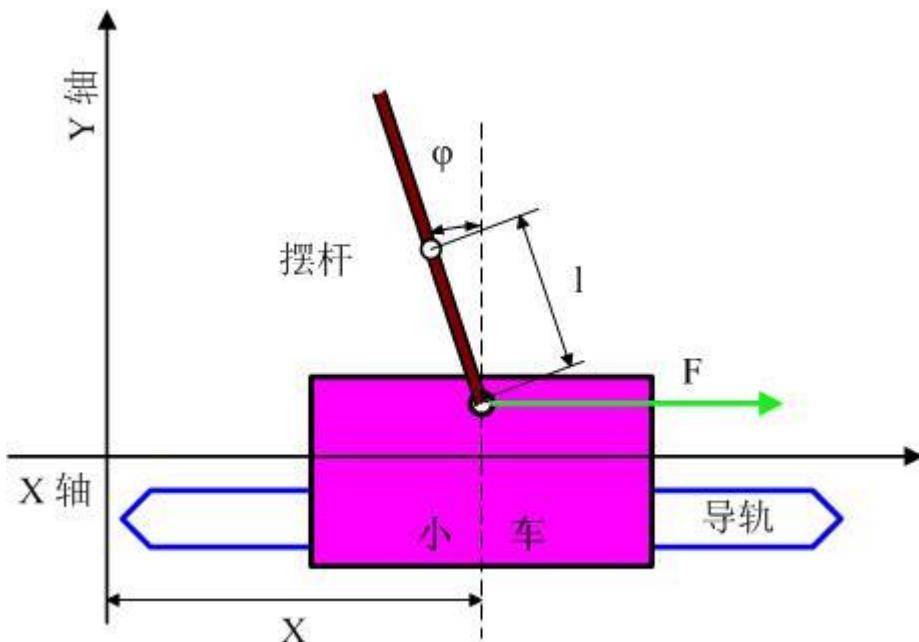


图 2 线性一阶倒立摆受力示意图

通过动力学分析建立拉格朗日方程可得到系统非线性动力学方程。若设定如下状态:

$$x_1 = x, x_2 = \dot{x}, x_3 = \varphi, x_4 = \dot{\varphi}$$

其中

x1 —— 小车相对于初始位置的位移

x2 —— 摆杆的转角

x3 —— 小车的速度

x4 —— 摆杆的角速度

线性化后的系统状态空间方程为

$$\dot{X} = AX + Bu$$

$$Y = CX$$

其中



$$X = [x_1 \quad x_2 \quad x_3 \quad x_4]^T$$

$$C = I_{4 \times 4}$$

带入参数得参数矩阵为

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0883167 & 0.629317 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.235655 & 27.8285 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0.883167 \\ 0 \\ 2.35655 \end{bmatrix}$$

## 2、LQR 控制算法

在 Command Window 中输入

```
A=[0 1 0 0;0 -0.0883167 0.629317 0;0 0 0 1;0 -0.235655  
27.8285 0];  
B=[0;0.883167;0;2.35655];  
Q=[10 0 0 0;0 1 0 0;0 0 10 0;0 0 0 1];  
R=1;  
K=lqr(A,B,Q,R)
```

得反馈矩阵  $K = [-3.1623 \quad -4.2854 \quad 37.3882 \quad 7.2137]$

设定摆杆初始位置为导轨的中点，导轨长度为 1.4m，设定小车的期望平衡位置为 50cm 处，建立如图 3 的 Simulink 控制框图，其中模型部分根据线性一阶倒立摆非线性模型由 S 函数编写，控制模块根据 LQR 控制方法建立。得到如图 4 的输出曲线（小车位置）和如图 5 的输出曲线（摆杆角度）。

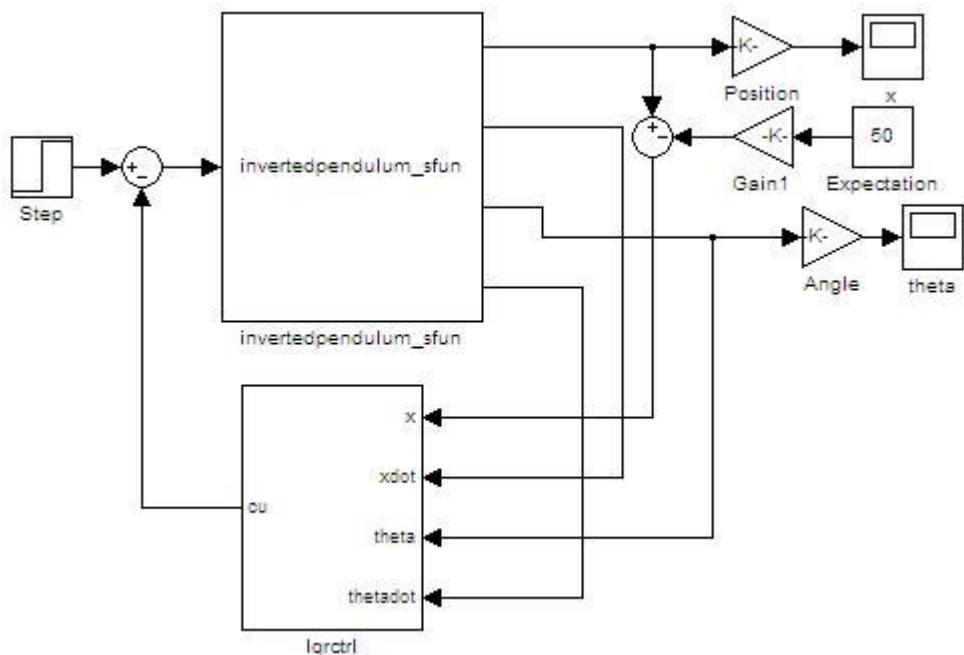


图 3 线性一阶倒立摆 Simulink 框图

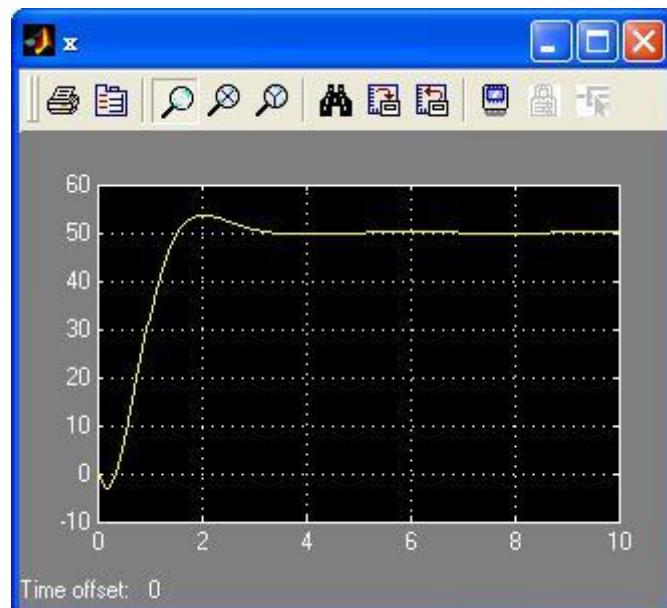


图 4 小车位置图

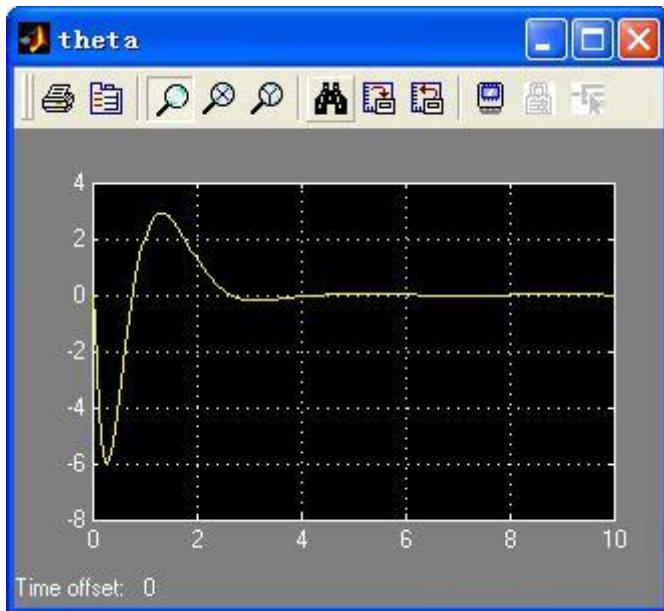


图 5 摆杆与垂直方向角度图

### 三 实验步骤

1、登陆 NCSLab，在倒立摆实验室中找到一阶线性倒立摆虚拟实验台，如图 6 所示。

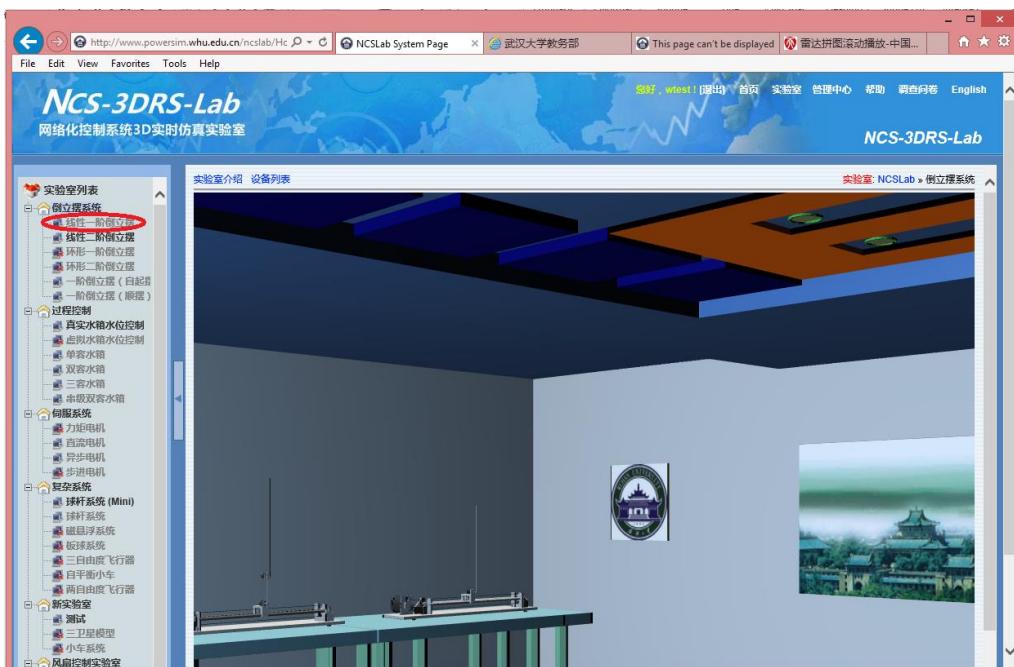


图 6. 找到线性一阶倒立摆系统

- 2、进入线性一阶倒立摆的主界面，仔细阅读设备介绍和设备文档
- 3、在实验算法中选择 LQR Control，下载到远程控制器中。
- 4、新建组态，如图 7 所示。建立如下的关联关系：



三维组件：

L1IP Settings

Angle:	Angle	Signals/Angle	
Position:	Position	Signals/Position	
Expectation:	Expectation	Paramters/Expectation	

输入组件：

Numeric Input Settings

名称:	Expectation
下限值:	
上限值:	
数据精度:	3 个有效数字
参数:	Paramters/Expectation

趋势图组件：

Chart Monitor Settings

名称:	Chart
时间范围:	10 seconds
刷新时间间隔:	
下限值:	
上限值:	

**关联信号**

Name:	Path	
Angle	Signals/Angle	
Position	Signals/Position	

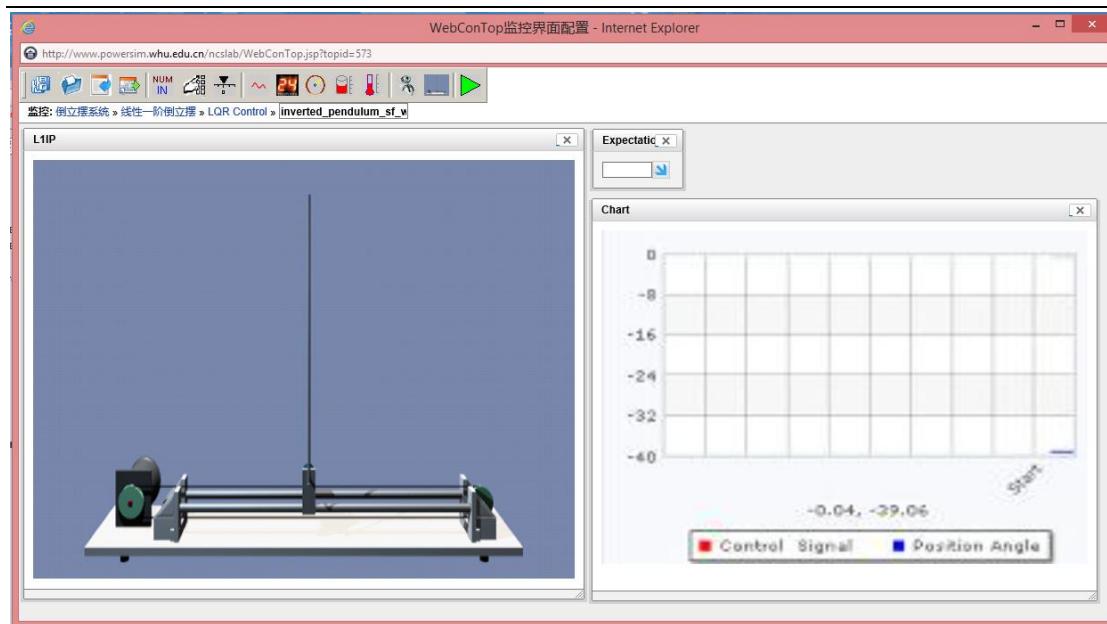


图 7. 组态图

- 5、点击运行按钮，开始进行虚拟实验。
- 6、改变小车位置的设定值，观察虚拟的倒立摆系统的控制过程是怎样的。如图 8 所示。

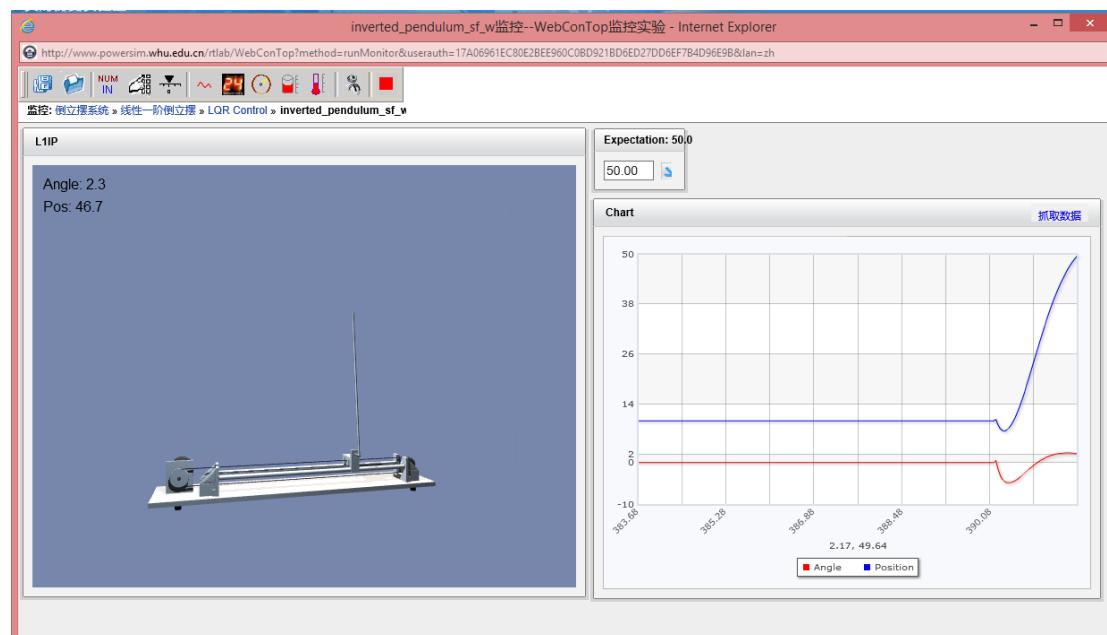


图 8. 虚拟线性一阶倒立摆的控制过程

## 五 思考题

- (1) 解释一阶倒立摆系统的原理。
- (2) 分析 Simulink 框图，解释一阶倒立摆系统 LQR 算法的细节。



## 六 实验报告

- (1) 需要将实验过程中的每一个关键步骤抓图粘贴到实验报告上。
- (2) 实验报告中要回答思考题提出的问题。



## 实验十一 虚拟线性二阶倒立摆的 LQR 控制

### 一 实验目的

- 1、了解线性二阶倒立摆的结构、原理和数学模型
- 2、了解线性二阶倒立摆的 LQR 控制算法

### 二 实验内容

- 1、了解线性二阶倒立摆的原理、结构
- 2、了解线性二阶倒立摆数学模型的建立方法
- 3、使用远程虚拟实验的方式，操作虚拟的线性二阶倒立摆控制平台，实现 LQR 控制。

### 三 实验原理

- 1、线性二阶倒立摆的结构和工作原理

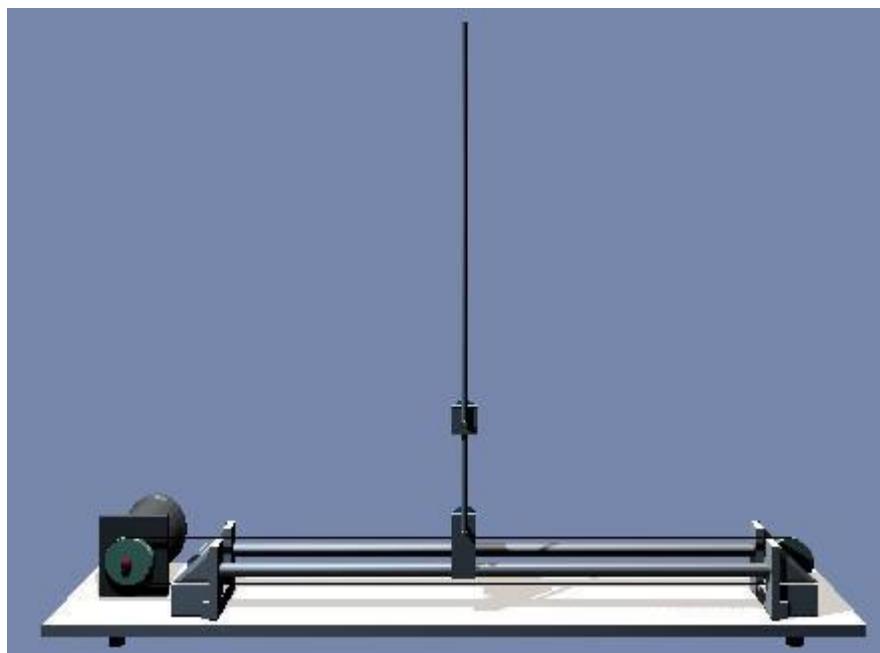


图 1 线性二阶倒立摆示意图

线性二阶倒立摆系统主要由以下几部分组成，如图 1 所示。其机械本体主要包括底座（导轨）、小车、驱动小车的交流伺服电机、同步皮带、一阶摆杆、二阶摆杆、限位开关及光电码盘等。通过控制交流伺服电机，带动皮带转动，在皮带的带动下小车可以在导轨上运动从而控制两阶摆杆的运动状态。交流伺服电机带有光电式脉冲编码盘，根据脉冲数目可得出工作轴的回转角度，由传动比换算出小车线性位移。在小车的运动导轨上有用于检测小车位置的传感器，小车位置



的信号被传送给控制系统，通过控制算法计算出控制量控制电机，从而控制小车的位置，使两阶摆杆垂直于水平面。我们的目的是设计一个控制器，通过控制电机的转动，使两阶摆杆稳定在垂直于水平面的位置。

## 2、线性二阶倒立摆的数学模型

若忽略空气阻力和各种摩擦力之后，可将线性二阶倒立摆系统抽象成小车和质量均匀的摆杆组成的系统，如图 2 所示。其中符号意为：

$F$  —— 小车所受力

$\theta_1$  —— 一阶摆杆与垂直向上方向的夹角（摆杆初始位置为垂直向下）

$\theta_2$  —— 二阶摆杆与垂直向上方向的夹角（摆杆初始位置为垂直向下）

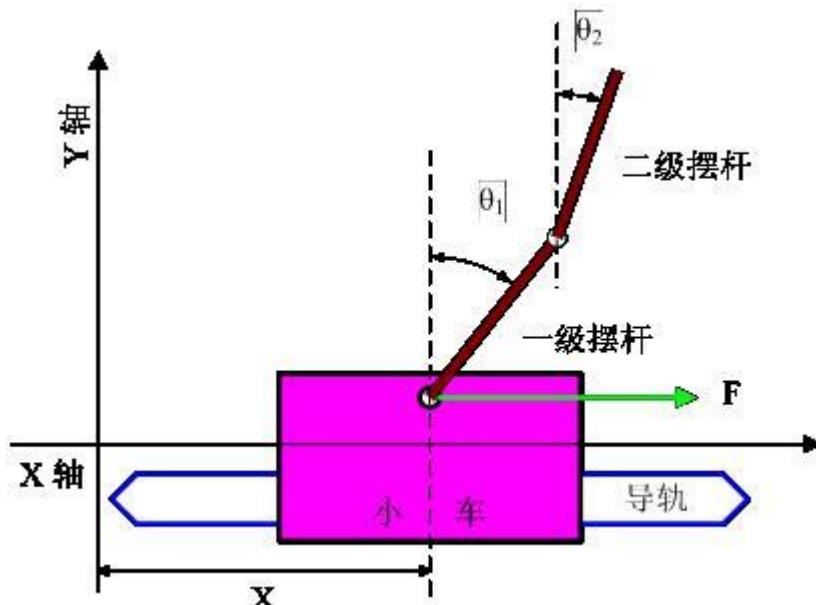


图 2 线性二阶倒立摆受力示意图

通过动力学分析建立拉格朗日方程可得到系统分析非线性动力学方程。若设定如下的状态：

$$x_1 = x, \quad x_2 = \theta_1, \quad x_3 = \theta_2, \quad x_4 = \dot{x}, \quad x_5 = \dot{\theta}_1, \quad x_6 = \dot{\theta}_2$$

其中

$x_1$  —— 小车相对于初始位置的位移

$x_2$  —— 一阶摆杆（下摆杆）的转角

$x_3$  —— 二阶摆杆（上摆杆）的转角

$x_4$  —— 小车的速度

$x_5$  —— 一阶摆杆（下摆杆）的角速度



x6——二阶摆杆（上摆杆）的角速度

带入参数可得系统在竖直向上位置处的状态空间方程如下式所示：

$$\dot{X} = AX + Bu$$

$$Y = CX$$

其中

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T$$

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -11.689 & 1.2821 & -20.2945 & 0.0624 & -0.02 \\ 0 & 114.8452 & -44.3562 & 70.7948 & -0.7292 & 0.3119 \\ 0 & -123.5638 & 102.0255 & -21.6317 & 0.9825 & -0.5335 \end{bmatrix}$$

$$B = [0 \ 0 \ 0 \ 1.6236 \ -5.6636 \ 1.7305]^T$$

$$C = I_{6 \times 6}$$

### 3、演示算法：LQR 控制方法

在 Command Window 中输入

```
A=[0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1;0 -11.689 1.2821 -20.2945  
0.0624 -0.02;0 114.8452 -44.3562 70.7948 -0.7292 0.3119;0 -123.5638 102.0255  
-21.6317 0.9825 -0.5335];
```

```
B=[0;0;1.6236;-5.6636;1.7305];
```

```
Q=[10 0 0 0 0 0;0 10 0 0 0 0;0 0 10 0 0 0;0 0 0 1 0 0;0 0 0 0 1 0;0 0  
0 0 0 1];
```

```
R=0.2;
```

```
K=lqr(A,B,Q,R)
```

得反馈矩阵

```
K = [7.0711 -185.9545 276.7819 5.1163 4.0862 30.1920]
```

设定摆杆初始位置为导轨的中点，导轨长度为 1.4m，设定小车的期望平衡



位置为 50cm 处，建立如图 3 的 Simulink 控制框图，其中模型部分根据线性一阶倒立摆非线性模型由 S 函数编写，控制模块根据 LQR 控制方法建立。得到如图 4 的输出曲线（小车位置）、如图 5 的输出曲线（一阶摆杆角度）和如图 6 的输出曲线（二阶摆杆角度）。

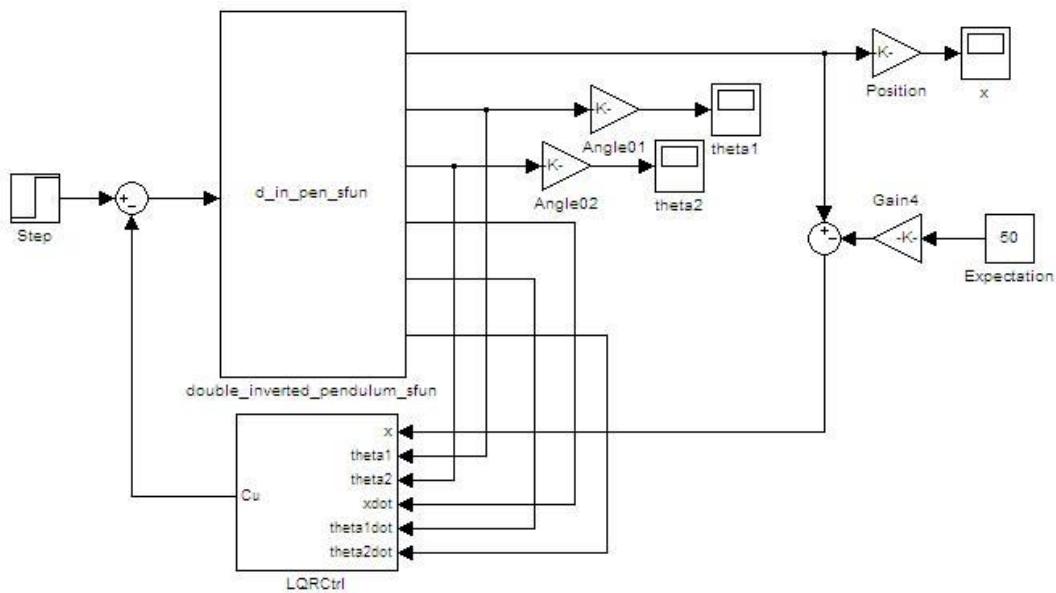


图 3 线性二阶倒立摆的 Simulink 框图

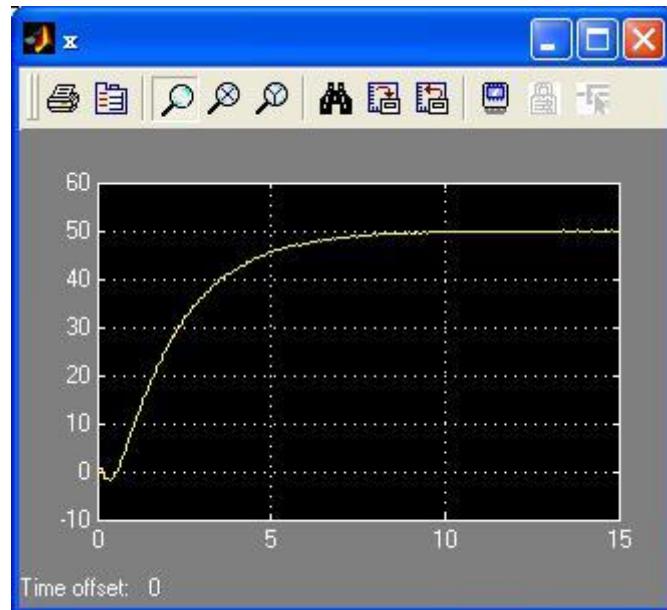


图 4 小车位置图

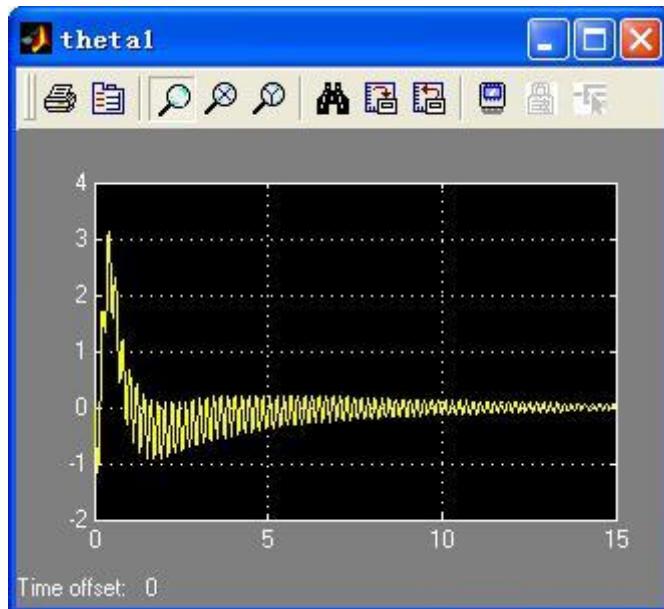


图 5 一阶摆杆与垂直方向角度图

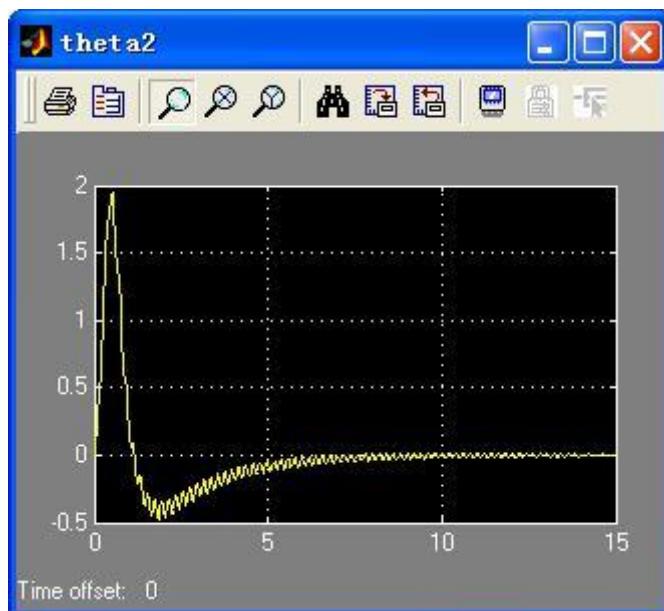


图 6 二阶摆杆与垂直方向角度图

### 三 实验步骤

1、登陆 NCSLab，在倒立摆实验室中找到二阶线性倒立摆虚拟实验台，如图 6 所示。



编写带有积分抗饱和的 PI 控制 S 函数

国家级电力生产过程  
虚拟仿真实验教学中心

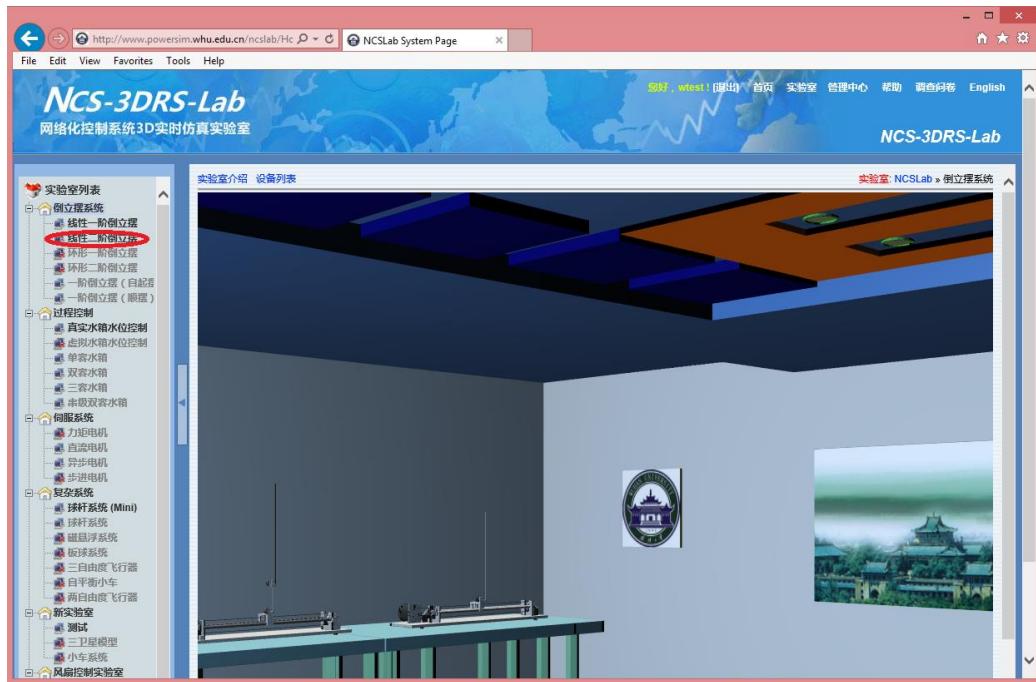
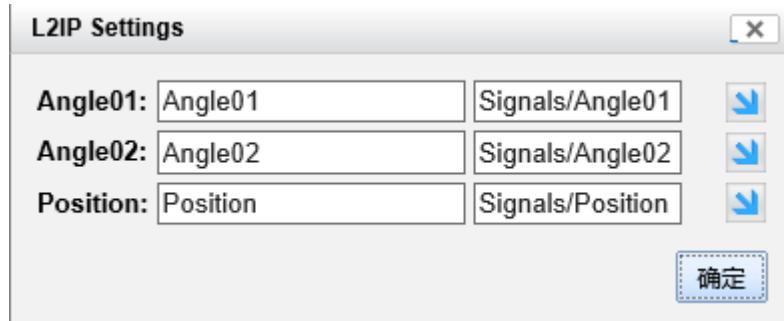


图 6. 找到线性二阶倒立摆系统

- 2、进入线性二阶倒立摆的主界面，仔细阅读设备介绍和设备文档
- 3、在实验算法中选择 LQR Control，下载到远程控制器中。
- 4、新建组态，如图 7 所示。建立如下的关联关系：

三维组件：



输入组件：



**Numeric Input Settings**

名称:	Expectation
下限值:	
上限值:	
数据精度:	3 个有效数字
参数:	Paramters/Expectation

趋势图组件:

**Chart Monitor Settings**

名称:	Chart
时间范围:	10 seconds
刷新时间间隔:	
下限值:	
上限值:	

**关联信号**

Name:	Path	
Angle01	Signals/Angle01	X
Angle02	Signals/Angle02	X
Position	Signals/Position	X



编写带有积分抗饱和的 PI 控制 S 函数

国家级电力生产过程  
虚拟仿真实验教学中心

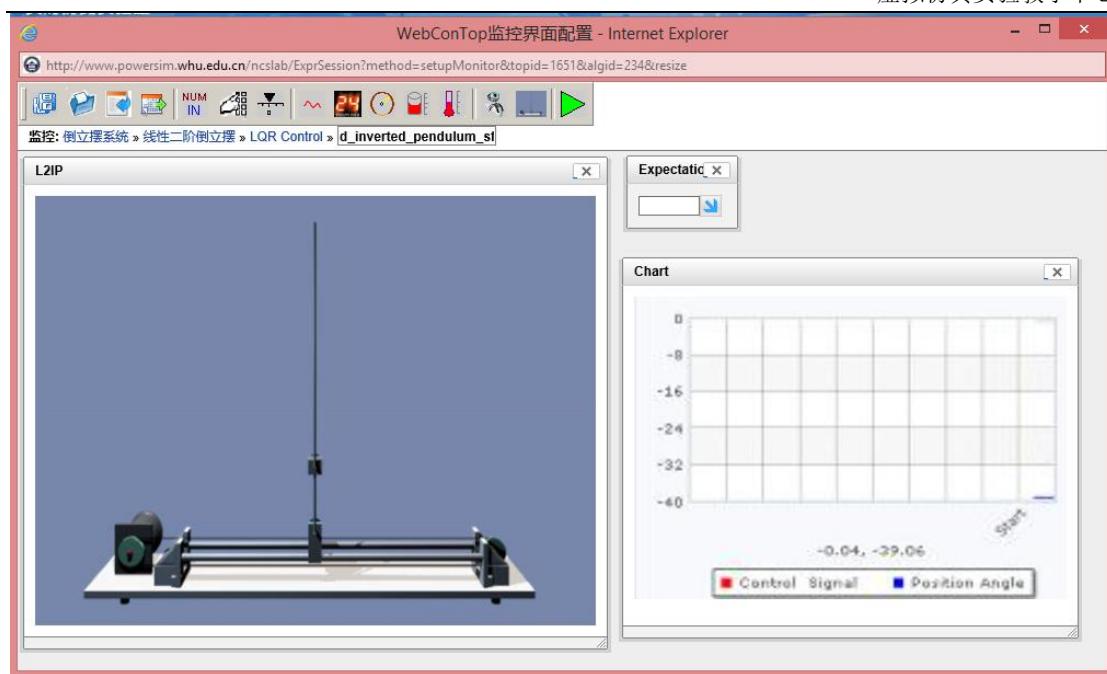


图 7. 组态图

5、点击运行按钮，开始进行虚拟实验。

6、改变小车位置的设定值，观察虚拟的倒立摆系统的控制过程是怎样的。如图 8 所示。

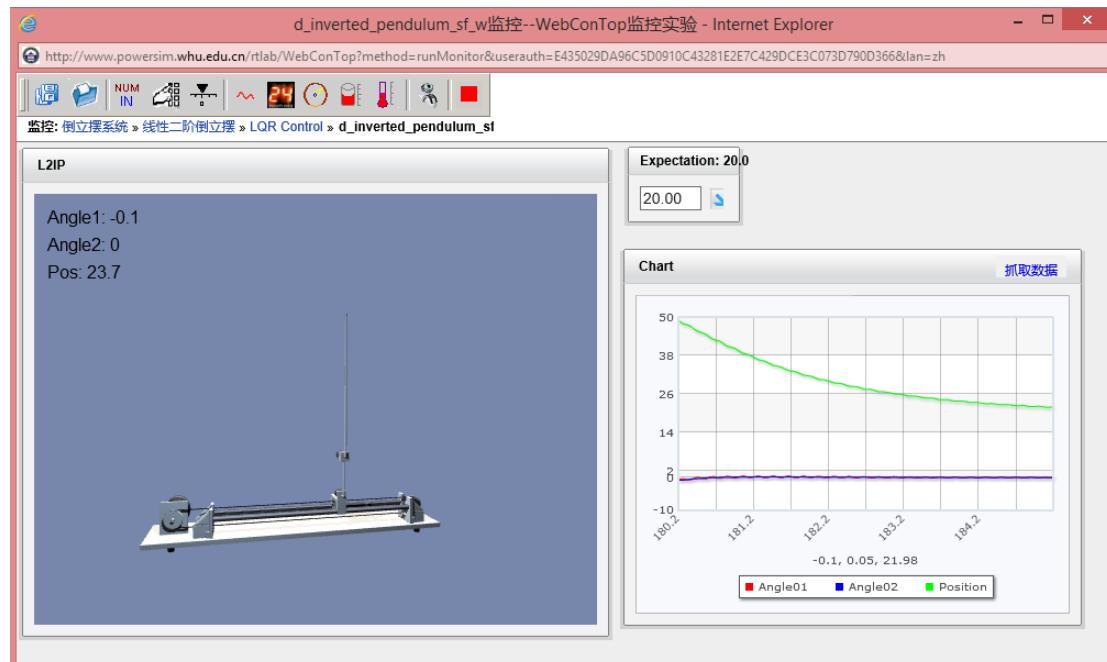


图 8. 虚拟线性一阶倒立摆的控制过程

## 五 思考题

(1) 解释二阶倒立摆系统的原理。



编写带有积分抗饱和的 PI 控制 S 函数

国家级电力生产过程  
虚拟仿真实验教学中心

---

(2) 分析 Simulink 框图，解释二阶倒立摆系统 LQR 算法的细节。

## 六 实验报告

- (1) 需要将实验过程中的每一个关键步骤抓图粘贴到实验报告上。
- (2) 实验报告中要回答思考题提出的问题。



## 实验十二 风扇系统数学模型的辨识（综合性实验）

### 一 实验目的

- 1、了解系统辨识过程中的实验设置，数据采集的方法
- 2、了解使用 Matlab 工具，进行数据分析，辨识被控对象数学模型的方法

### 二 实验内容

- 1、了解风扇控制系统的原理、结构
- 2、了解使用 NCSLab 平台进行远程数据采集实验的方法
- 3、了解使用 NCSLab 平台进行远程数据采集的方法
- 4、了解使用 Matlab 的系统辨识工具箱，使用采集的数据，进行风扇控制系统数学模型辨识的方法。

### 三 实验原理

#### 1、风扇控制系统原理

风扇控制系统放置在主教 917 电力生产过程虚拟仿真教学中心的实验室中，如图 1 所示。它已经连入 NCSLab 网络控制平台，允许学生使用 Internet 进行远程访问。

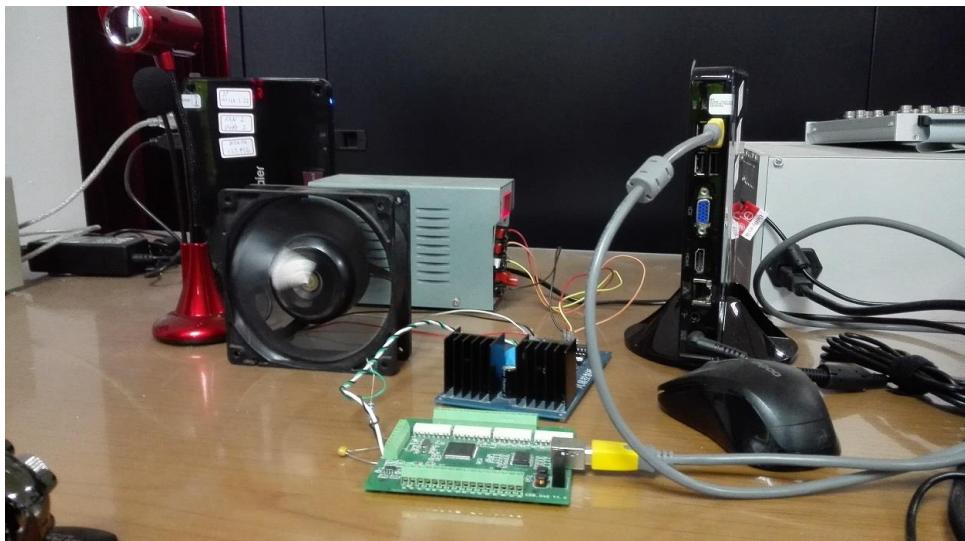


图 1. 实体的风扇控制系统

它的原理框图如图 2 所示。使用控制器在风扇上施加电压，就能驱动风扇转动。通过测量风扇叶片转动的频率，就可以得到风扇的转速。在控制器上使用反馈控制算法，就可以将风扇调节在我们需要的转速上。



图 2. 风扇系统的原理

## 2、风扇系统数学模型辨识的 Simulink 框图

选择合适输入信号，施加到风扇控制系统上，建立系统辨识的框图，如图 3 所示。

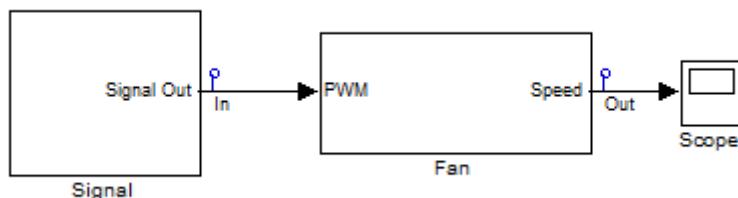


图 4. 风扇系统的原理

其中信号机理模块 Signal 的结构如图所示，它可以生成频率均匀变化的方波，幅值在 0.1 到 0.9 之间，如图 5 所示。

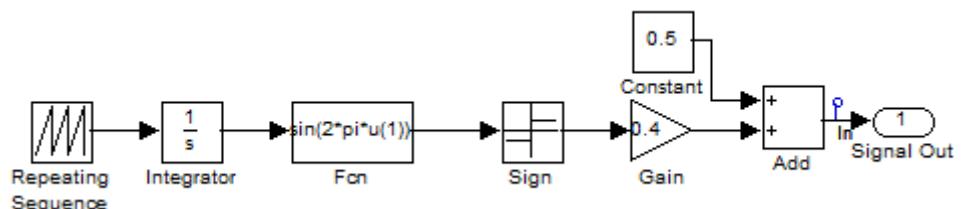


图 5. 信号发生模块的机构

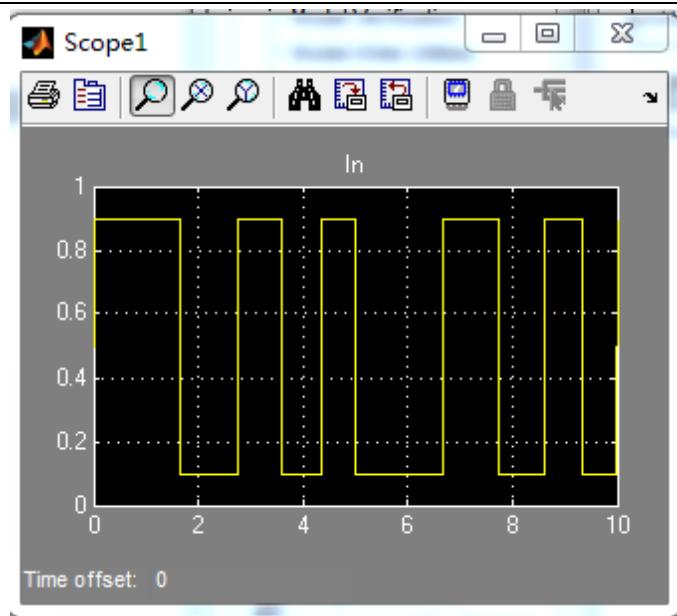


图 6. 激励信号的波形

### 3、风扇系统数学模型辨识的实验和数据采集

使用 Matlab 的 RTW 工具箱，生成可执行的程序，通过 NCSLab 网站下载到远程的实验平台上进行实验。通过网站远程采集风扇系统的输入量和输出量，将实验数据保存在文件中。

### 4、风扇系统数学模型辨识的辨识

使用 Matlab 的系统辨识工具箱，见实验数据导入到工作区中，一部分数据作为辨识量，另一部分作为验证量。辨识出系统的数学模型。

## 四 实验步骤

- 1、安装 WMWare 虚拟机软件
- 2、将带有 Matlab 和 Visual Studio 的 Win7 虚拟机导入到虚拟机软件中
- 3、在虚拟机软件中运行 Win7 系统
- 4、打开虚拟机中的 Matlab 软件，打开 Simulink
- 5、在 Simulink 中打开 FanIdent.mdl 文件，使用 RTW 工具，生成可执行程序 FanIdent.exe
- 6、使用 Simulink 生成可执行算法，通过复制粘贴把可执行程序从虚拟机中拷贝出来
- 7、去掉可执行文件的.exe 后缀，使用自己的账号登陆 NCSLab，将算法文件上传到服务器上。
- 8、获得系统控制权，将上传算法下载到远程的风扇控制系统中去



9、建立组态，如图 7 所示。

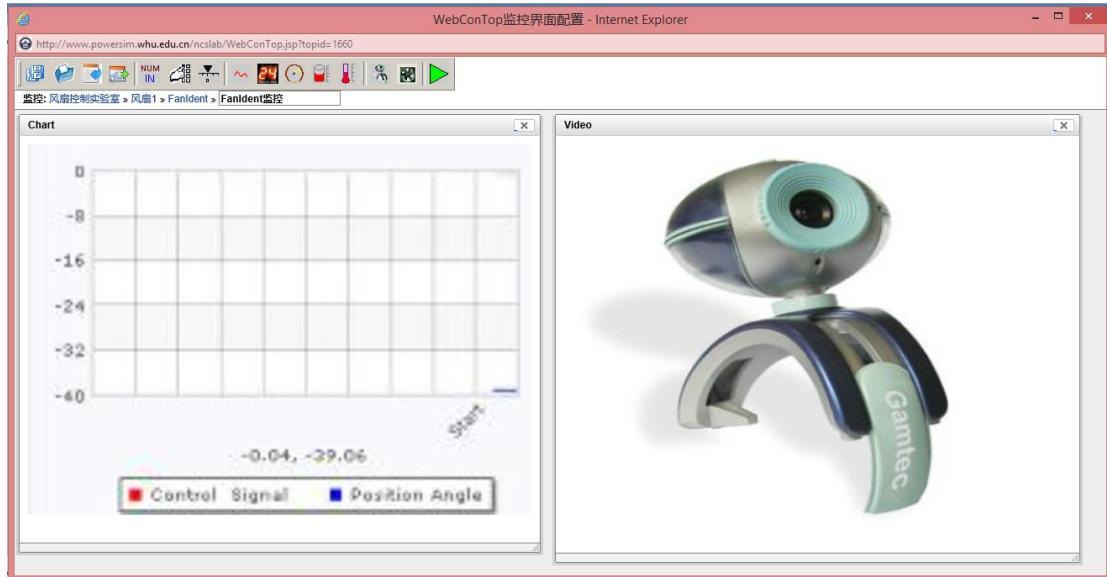


图 7. 系统辨识的组态

其中，趋势图的关联关系为：

**Chart Monitor Settings**

名称:	Chart
时间范围:	10 seconds
刷新时间间隔:	seconds
下限值:	
上限值:	

**关联信号**

Name:	Path	X
In	Signals/In	X
Speed	Signals/Speed	X

**操作按钮**

添加信号 确定

10、点击运行按钮，开始远程实验，如图 8 所示。



编写带有积分抗饱和的 PI 控制 S 函数

国家级电力生产过程  
虚拟仿真实验教学中心

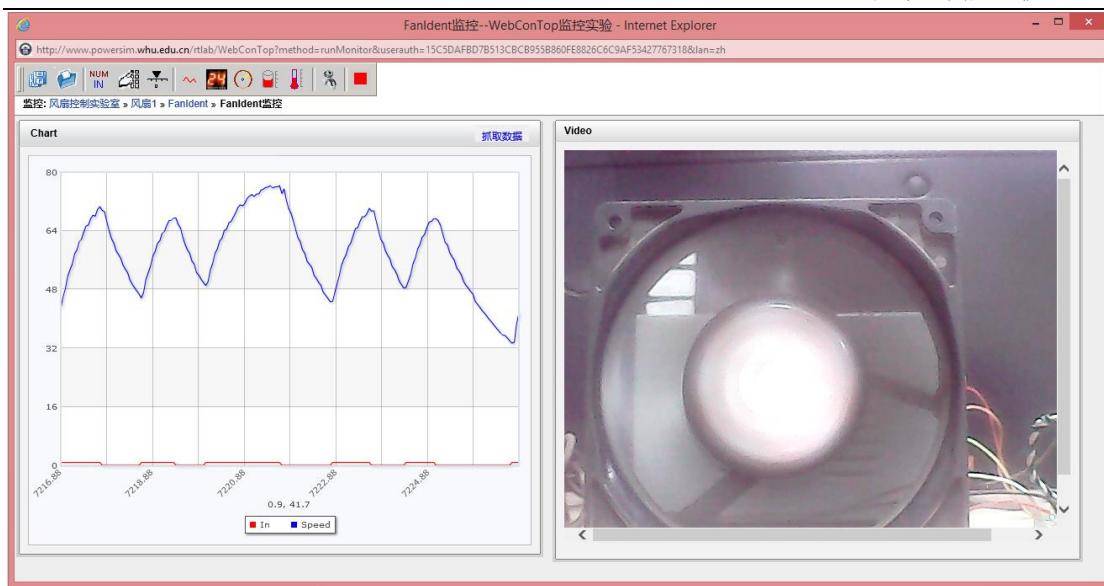


图 8. 数据采集实验界面

11、点击抓取数据按钮，将实时数据保存，并用记事本打开，如图 9 所示。

data_20150708_172626.txt - Notepad			
File	Edit	Format	View
Time	data0	data1	
7326.84	0.9	40.20000076293945	
7326.85	0.9	43.099998474121094	
7326.8600000000001	0.9	43.099998474121094	
7326.87	0.9	44.099998474121094	
7326.88	0.9	44.099998474121094	
7326.89	0.9	44.099998474121094	
7326.9000000000001	0.9	45.79999923706055	
7326.91	0.9	45.79999923706055	
7326.92	0.9	46.599998474121094	
7326.93	0.9	46.599998474121094	
7326.9400000000005	0.9	48.099998474121094	
7326.95	0.9	48.099998474121094	
7326.96	0.9	48.70000076293945	
7326.97	0.9	48.70000076293945	
7326.9800000000005	0.9	50.20000076293945	
7326.99	0.9	50.20000076293945	
7327.0	0.9	50.70000076293945	
7327.01	0.9	50.70000076293945	
7327.02	0.9	52.20000076293945	
7327.03	0.9	52.20000076293945	
7327.04	0.9	52.400001525878906	
7327.05	0.9	52.400001525878906	
7327.06	0.9	53.70000076293945	
7327.07	0.9	54.099998474121094	
7327.08	0.9	54.099998474121094	
7327.09	0.9	55.29999923706055	
7327.1	0.9	55.29999923706055	
7327.1100000000001	0.9	55.5	
7327.12	0.9	55.5	
7327.13	0.9	56.79999923706055	
7327.14	0.9	56.79999923706055	
7327.1500000000001	0.9	56.900001525878906	
7327.16	0.9	58.099998474121094	
7327.17	0.9	58.099998474121094	

图 9. 抓取的实时数据

12、抓取两组数据，导入到 Matlab 中，这两组数据的输入序列分别命名 u1 和 u2，输出序列分别命名 y1 和 y2

13、打开系统辨识工具箱，将两组数据导入，其中一组取名 estimate，用于辨识，另一组取名 validate，用于验证。分别导入系统辨识工具箱的 GUI 界面中，



如图 10 所示。

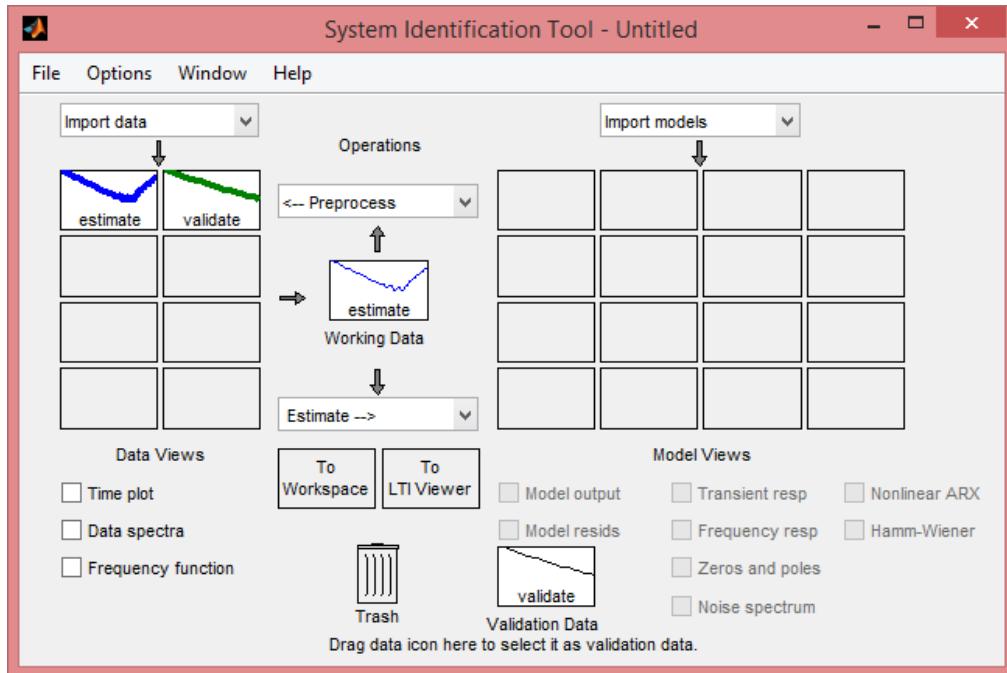


图 10. 将两组数据导入系统辨识工具箱

14、进行系统辨识，获得几组候选模型，如 arx212、arx222、arx772 等，如图 11 所示。

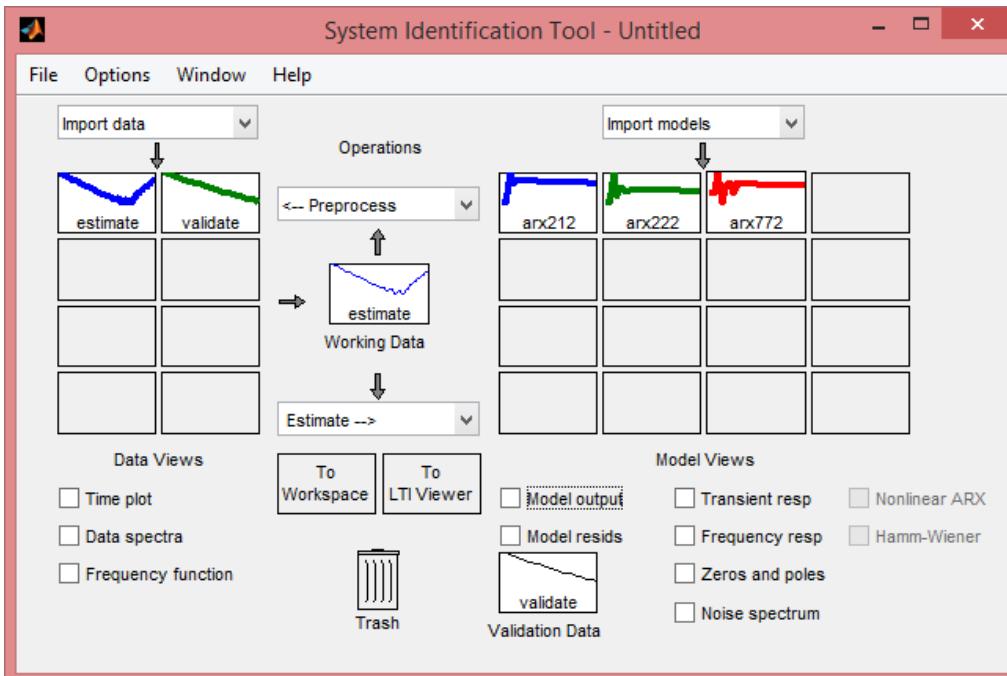


图 11. 获得几组候选数学模型

15、根据 Model Output 选择一个最佳的数学模型，如图 12 所示。

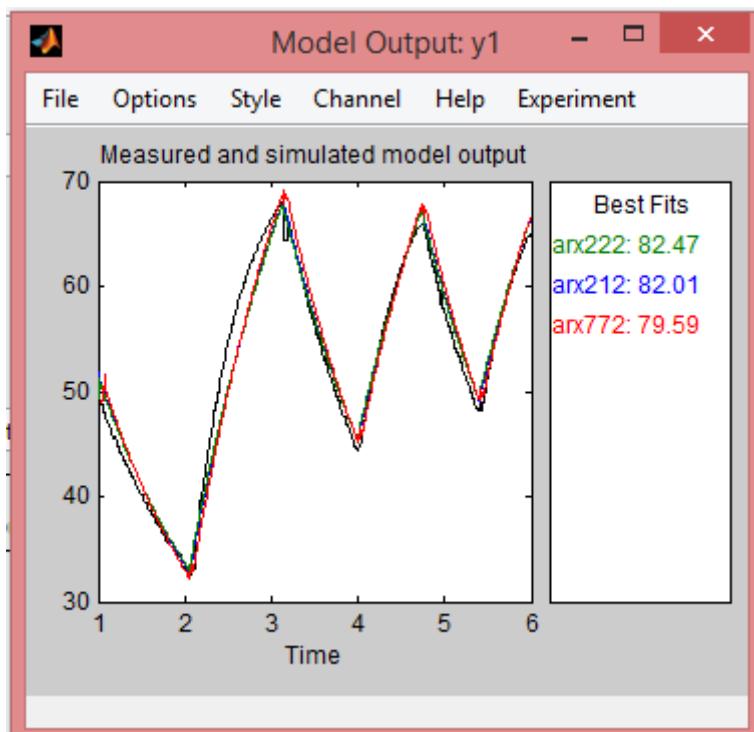


图 12. 选择最佳的数学模型

16、将最佳的数学模型导出到 Matlab 的工作区中，得到表达式，附在实验报告中。

## 五 思考题

(1) 什么是振幅因数？为什么在进行系统辨识实验中，我们要选取振幅因数较大的输入信号序列？

(2) 是否能够优化系统辨识的实验设置，如采用不同的输入数据序列，获得更加准确的数学模型？（有附加分数 10 分）

## 六 实验报告

(1) 需要将实验过程中的每一个关键步骤抓图粘贴到实验报告上。

(2) 实验报告中要回答思考题提出的问题。