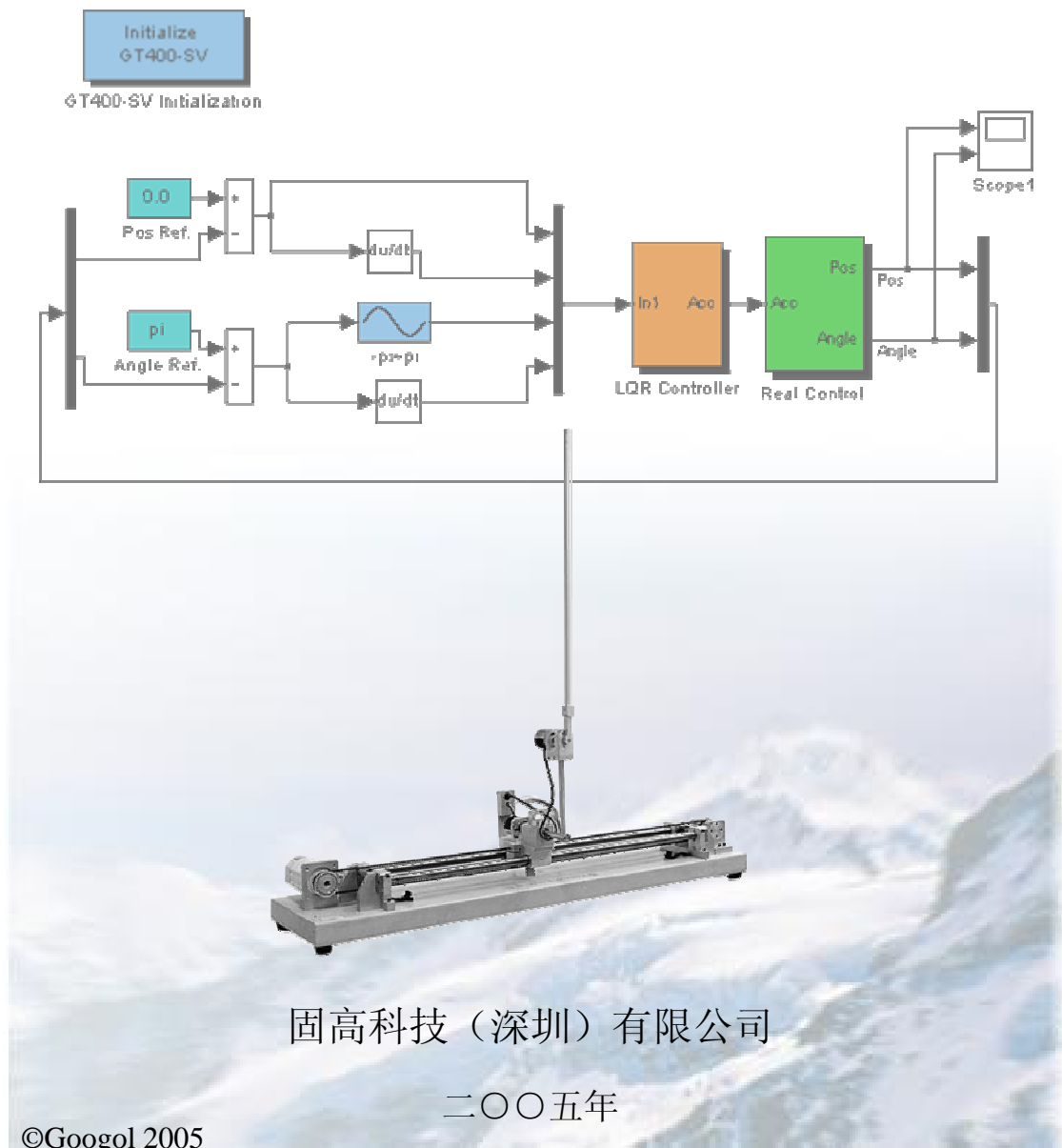


倒立摆与 自动控制原理实验 V2.0



版权声明

固高科技（深圳）有限公司

保留所有版权

固高科技有限公司（以下简称固高科技）具有本产品及其软件的专利权、版权和其它知识产权。未经授权，不得直接或间接的复制、制造、加工、使用本产品及相关部分。

直线倒立摆系统GLIP 系列包含《直线倒立摆系统GLIP 系列安装与使用手册》和《倒立摆与自动控制原理实验》。

声明

固高科技保留在不预先通知的情况下修改设备和文档的权力。

固高科技不承担由于使用本说明书或本产品不当，所造成的直接的、间接的、特殊的、附带的、或相应的损失和赔偿。

商标声明

Windows 和 Microsoft 为 Microsoft 公司注册商标。

Mathematica 为沃尔夫勒姆研究公司（Wolfram Research Inc.）公司注册商标。

Matlab 为 Mathworks 公司注册商标





安全注意事项

直线倒立摆系统主要用于教学和科研。在安装，使用和维护之前，请仔细阅读本安装手册。请将本手册妥善保存，以备需要时随时查阅。


使用注意事项

使用（安装、运转、保养、检修）前，请务必熟悉并全部掌握本手册和其它相关资料，在熟知全部机器知识、安全知识、以及注意事项后再使用设备。
本手册将安全注意事项分为“危险”“注意”“强制”“禁止”分别记载。

表 1-1 警告标志

 危 险	不正确的操作将会导致重大人身事故。
 注 意	不正确的操作会导致设备损坏。
 强 制	必须要做的操作。
 禁 止	被禁止的操作。

另外，即使“注意”所记载的内容，也可能因为不同的情况产生严重后果，因此任何一条注意事项都很重要，在设备使用过程中请严格遵守。

	虽然不符合“危险”“注意”的内容，但是用户在使用过程中必须严格遵守的事项，在相关地方以记载。
---	--



强 制

- 本手册记述了安全上一般应该注意的事项，在实际实验环境下用于实验研究人员的安全措施不可能完全记载，敬请原谅。
- 为了安全的使用直线倒立摆控制系统，用户必须按照本说明书的要求对设备操作及维护人员进行安全教育，直接操作人员必须认真阅读直线倒立摆系统所有说明书。



注 意

- 本手册中的图以及相片，为代表性实例，和实际产品可能有所不同；
- 由于破损或是丢失说明书需要定购说明书，请与我公司销售部门联系；
- 客户自行进行产品改造，不在本公司保修范围之内，本公司概不负责。



危 险

- 在操作运行的机器中，存在一定的危险，请实验人员注意可能会出现的情况。

内容简介

这是一本为工科自动化和机电一体化的专科、本科以及研究生编写的实验教科书，可以作为控制领域各门控制课程的配套实验教材，本书的主要内容包括经典控制理论的部分实验内容和现代控制理论的部分实验内容，经典控制理论内容包括实际系统模型的建立、根轨迹分析和控制器设计、频率响应分析、PID 控制分析等内容，现代控制理论内容主要包括状态空间极点配置和线性最优控制 LQR 方法，本书实验内容主要基于固高科技倒立摆完成。

前言

倒立摆是进行控制理论研究的典型实验平台。由于倒立摆系统的控制策略和杂技运动员顶杆平衡表演的技巧有异曲同工之处，极富趣味性，而且许多抽象的控制理论概念如系统稳定性、可控性和系统抗干扰能力等等，都可以通过倒立摆系统实验直观的表现出来，因此在欧美发达国家的高等院校，它已成为必备的控制理论教学实验设备。学习自动控制理论的学生通过倒立摆系统实验来验证所学的控制理论和算法，非常的直观、简便，在轻松的实验中对所学课程加深了理解。

倒立摆不仅仅是一种优秀的教学实验仪器，同时也是进行控制理论研究的理想实验平台。由于倒立摆系统本身所具有的高阶次、不稳定、多变量、非线性和强耦合特性，许多现代控制理论的研究人员一直将它视为典型的研究对象，不断从中发掘出新的控制策略和控制方法，相关的科研成果在航天科技和机器人学方面获得了广阔的应用。二十世纪九十年代以来，更加复杂多种形式的倒立摆系统成为控制理论研究领域的热点，每年在专业杂志上都会有大量的优秀论文出现。

固高科技有限公司（以下简称固高科技）为高等院校的自动控制教学提供了整套基于倒立摆系统的实验解决方案。包括各种摆的开发生产、实验内容的安排和配置，以及对应的自动控制理论教学内容和相关经典教材的推荐。固高科技开发生产的倒立摆系列包括直线运动型、圆周运动型和复合倒立摆三个大系列，主要特点包括：

开放性：采用四轴运动控制板卡，机械部分和电气部分非常容易扩展，可以根据用户需要进行配置。系统软件接口充分开放，用户不仅可以使⽤配套的实验软件，而且可以根据⾃⼰的实际需要扩展软件的功能。

模块化：系统的机械部分可以选用直线或者旋转平台，根据实际需要配置成一级、二级或者三级倒立摆。而三级摆可以方便地改装成两级摆，两级摆可以改装成一级摆。系统实验软件同样是基于模块化的思想设计，用户可以根据需要增加或者修改相应的功能模块。

简易安全：摆实验系统包括运动控制板卡、电控箱（旋转平台系统中和机械本体联在一起）、机械本体和微型计算机几个部分组成，安装升级方便。同时在机械、运动控制板卡和实验软件上都采取了积极措施，保证实验时人员的安全可靠和仪器安全。

方便性：倒立摆系统易于安装、升级，同时软件界面操作简单。

先进性：采用工业级四轴运动控制板卡作为核心控制系统，先进的交流伺服电机作为驱动，检测元件使用高精度高性能光电码盘。系统设计符合当今先进的运动控制发展方向。

实验软件多样化：用于实验的软件包括经典的BorlandC++，VC++，以及控制领域使用最多的仿真工具Matlab，提供完备的设备接口和程序接口，方便用户进行实验和开发。

配套实验教材：配备完备的《安装手册》、《使用手册》和《实验指导书》，用户可以根据学生实际情况开设相应的实验课程，也可以自行设计实验。

固高倒立摆系统适应如下课程的实验：自动控制原理，现代控制理论，现代控制工程，最优控制，非线性系统控制，智能控制，模糊控制和神经网络控制等

等。本书针对固高直线一级倒立摆组织了一系列实验，教师可以根据学生实际情况开设相应的实验课程。在开设本书实验课程时，用户必须同时具备有《GT 系列运动控制器用户手册》、《GT 系列运动控制器编程手册》、《Panasonic 交流伺服电机驱动器使用说明书》和其它相关元器件使用说明书。

目录

版权声明	I
声明 I	
商标声明	I
安全注意事项	II
使用注意事项	II
内容简介	IV
前言 V	
目录 VII	
第 1 章 倒立摆系统介绍.....	1
1.1 倒立摆系统简介	1
1.2 倒立摆分类.....	1
1.3 倒立摆的特性.....	3
1.4 控制器设计方法.....	4
第 2 章 运动控制基础实验.....	5
2.1 编码器原理及使用实验.....	5
2.1.1 编码器原理.....	5
2.1.2 角度换算	6
2.1.3 编码器使用实验.....	6
2.2 MATLAB SIMULINK 环境下电机控制实现.....	15
第 3 章 直线倒立摆建模、仿真及实验.....	24
3.1 直线一级倒立摆.....	24
3.1.1 直线一级倒立摆的物理模型.....	24
3.1.1.1 微分方程的推导	24
3.1.1.2 系统物理参数.....	32
3.1.1.3 实际系统模型.....	32
3.1.1.4 系统可控性分析	33
3.1.2 系统阶跃响应分析.....	35
3.1.3 直线一级倒立摆根轨迹控制实验.....	37
3.1.3.1 根轨迹分析.....	37
3.1.3.2 根轨迹校正及仿真.....	39
3.1.3.3 根轨迹校正实时控制实验	53
3.1.3.4 实验结果及实验报告	56
3.1.4 直线一级倒立摆频率响应控制实验.....	58
3.1.4.1 频率响应分析.....	58
3.1.4.2 频率响应设计及仿真	60
3.1.4.3 直线一级倒立摆频率响应校正法实验	68
3.1.4.4 实验结果及实验报告	71
3.1.5 直线一级倒立摆 PID 控制实验.....	72

3.1.5.1	PID 控制分析	72
3.1.5.2	PID 控制参数设定及仿真.....	73
3.1.5.3	PID 控制实验	77
3.1.5.4	实验结果与实验报告	81
3.1.6	状态空间极点配置控制实验.....	82
3.1.6.1	状态空间分析.....	82
3.1.6.2	极点配置及仿真.....	83
3.1.6.3	极点配置控制实验.....	94
3.1.6.4	实验结果及实验报告	97
3.1.7	线性二次最优控制 LQR 控制实验.....	98
3.1.7.1	线性二次最优控制 LQR 基本原理及分析	98
3.1.7.2	LQR 控制参数调节及仿真.....	98
3.1.7.3	直线一级倒立摆 LQR 控制实验	103
3.1.7.4	实验结果及实验报告	106
第 4 章	直线一级顺摆建模和实验.....	107
4.1	直线一级顺摆的建模与分析.....	107
4.1.1	直线一级顺摆的建模.....	107
4.1.2	实际系统模型.....	111
4.1.3	系统可控性分析.....	112
4.2	直线一级顺摆根轨迹分析.....	113
4.3	直线一级顺摆频率响应分析.....	114
4.4	直线一级顺摆阶跃响应分析.....	115
4.5	直线一级顺摆的 PID 控制仿真与实验.....	116
4.5.1	直线一级顺摆的 PID 控制分析与仿真.....	116
4.5.2	直线一级顺摆的 PID 实时控制实验.....	119
4.6	直线一级顺摆的 LQR 控制仿真与实验.....	122
4.6.1	直线一级顺摆的 LQR 控制分析与仿真.....	122
4.6.2	直线一级顺摆的 LQR 控制控制实验.....	126
第 5 章	直线一级倒立摆自动摆起控制实验.....	130
5.1	摆起的能量控制策略.....	130
5.2	直线一级倒立摆摆起控制实验	130
5.3	其它算法实验实现方法.....	133
第 6 章	直线两级倒立摆.....	135
6.1	系统物理模型.....	135
6.2	系统可控性分析.....	141
6.3	直线两级倒立摆 MATLAB 仿真.....	143
6.4	LQR 控制器设计及仿真.....	145
6.5	直线两级倒立摆 LQR 控制实验.....	149
6.6	实验结果及实验报告	153
6.7	其它算法实验实现方法.....	153
第 7 章	直线三级倒立摆.....	154

7.1 系统物理模型.....	154
第 8 章 附录.....	164
8.1 图表目录.....	164
8.2 程序目录.....	164

第1章 倒立摆系统介绍

1.1 倒立摆系统简介

倒立摆是机器人技术、控制理论、计算机控制等多个领域、多种技术的有机结合，其被控系统本身又是一个绝对不稳定、高阶次、多变量、强耦合的非线性系统，可以作为一个典型的控制对象对其进行研究。最初研究开始于二十世纪50年代，麻省理工学院（MIT）的控制论专家根据火箭发射助推器原理设计出一级倒立摆实验设备。近年来，新的控制方法不断出现，人们试图通过倒立摆这样一个典型的控制对象，检验新的控制方法是否有较强的处理多变量、非线性和绝对不稳定系统的能力，从而从中找出最优秀的控制方法。倒立摆系统作为控制理论研究中的一种比较理想的实验手段，为自动控制理论的教学、实验和科研构建一个良好的实验平台，以用来检验某种控制理论或方法的典型方案，促进了控制系统新理论、新思想的发展。由于控制理论的广泛应用，由此系统研究产生的方法和技术将在半导体及精密仪器加工、机器人控制技术、人工智能、导弹拦截控制系统、航空对接控制技术、火箭发射中的垂直度控制、卫星飞行中的姿态控制和一般工业应用等方面具有广阔的利用开发前景。平面倒立摆可以比较真实的模拟火箭的飞行控制和步行机器人的稳定控制等方面的研究。

1.2 倒立摆分类

倒立摆已经由原来的直线一级倒立摆扩展出很多种类，典型的有直线倒立摆，环形倒立摆，平面倒立摆和复合倒立摆等，倒立摆系统是在运动模块上装有倒立摆装置，由于在相同的运动模块上可以装载不同的倒立摆装置，倒立摆的种类由此而丰富很多，按倒立摆的结构来分，有以下类型的倒立摆：

1) 直线倒立摆系列

直线倒立摆是在直线运动模块上装有摆体组件，直线运动模块有一个自由度，小车可以沿导轨水平运动，在小车上装载不同的摆体组件，可以组成很多类别的倒立摆，直线柔性倒立摆和一般直线倒立摆的不同之处在于，柔性倒立摆有两个可以沿导轨滑动的小车，并且在主动小车和从动小车之间增加了一个弹簧，作为柔性关节。直线倒立摆系列产品如图 1-1 所示。

2) 环形倒立摆系列

环形倒立摆是在圆周运动模块上装有摆体组件，圆周运动模块有一个自由度，可以围绕齿轮中心做圆周运动，在运动手臂末端装有摆体组件，根据摆体组件的级数和串连或并联的方式，可以组成很多形式的倒立摆。如图 1-2 所示。

3) 平面倒立摆系列

平面倒立摆是在可以做平面运动的运动模块上装有摆杆组件，平面运动模块主要有两类：一类是 XY 运动平台，另一类是两自由度 SCARA 机械臂；摆体组件也有一级、二级、三级和四级很多种。如图 1-3 所示

4) 复合倒立摆系列

复合倒立摆为一类新型倒立摆，由运动本体和摆杆组件组成，其运动本体可以很方便的调整成三种模式，一是 2)中所述的环形倒立摆，还可以把本体翻转 90 度，连杆竖直向下和竖直向上组成托摆和顶摆两种形式的倒立摆。

按倒立摆的级数来分：有一级倒立摆、两级倒立摆、三级倒立摆和四级倒立摆，一级倒立摆常用于控制理论的基础实验，多级倒立摆常用于控制算法的研究，倒立摆的级数越高，其控制难度更大，目前，可以实现的倒立摆控制最高为四级倒立摆。

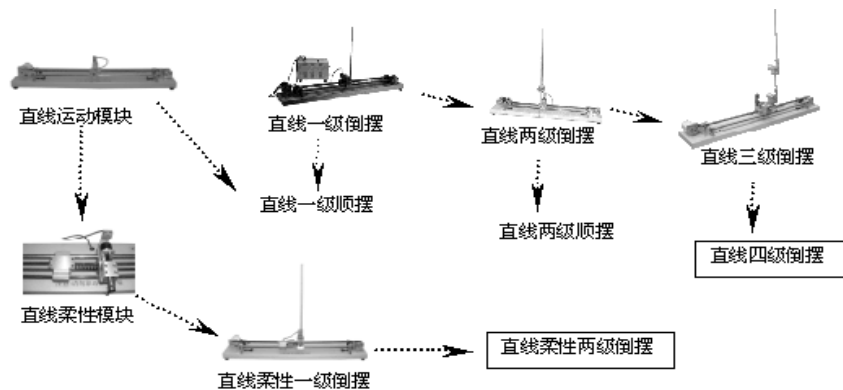


图 1-1 直线倒立摆系列

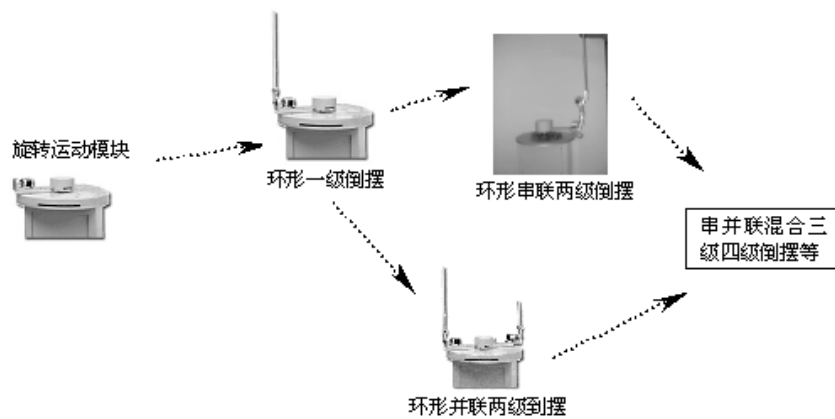


图 1-2 环形倒立摆系列

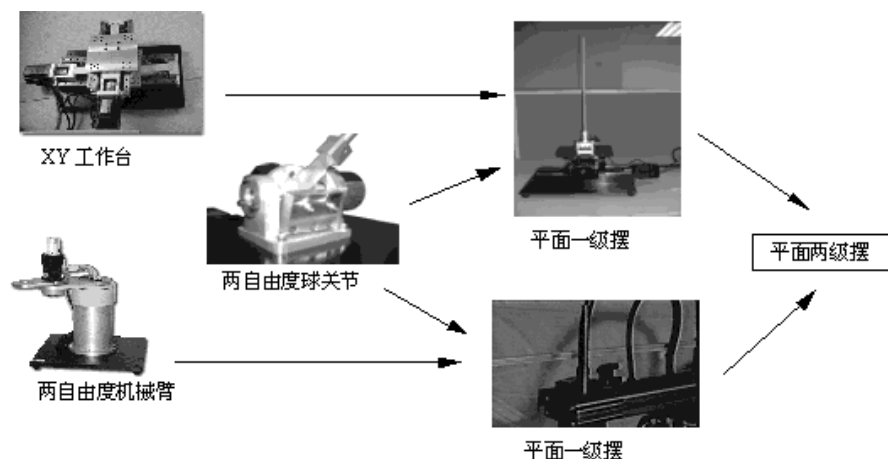


图 1-3 平面倒立摆系列



图 1-4 复合倒立摆

1.3 倒立摆的特性

虽然倒立摆的形式和结构各异，但所有的倒立摆都具有以下的特性：

1) 非线性

倒立摆是一个典型的非线性复杂系统，实际中可以通过线性化得到系统的近似模型，线性化处理后再进行控制。也可以利用非线性控制理论对其进行控制。倒立摆的非线性控制正成为一个研究的热点。

2) 不确定性

主要是模型误差以及机械传动间隙，各种阻力等，实际控制中一般通过减少各种误差来降低不确定性，如通过施加预紧力减少皮带或齿轮的传动误差，利用滚珠轴承减少摩擦阻力等不确定因素。

3) 耦合性

倒立摆的各级摆杆之间，以及和运动模块之间都有很强的耦合关系，在倒立摆的控制中一般都在平衡点附近进行解耦计算，忽略一些次要的耦合量。

4) 开环不稳定性

倒立摆的平衡状态只有两个，即在垂直向上的状态和垂直向下的状态，其中垂直向上为绝对不稳定的平衡点，垂直向下为稳定的平衡点。

5) 约束限制

由于机构的限制，如运动模块行程限制，电机力矩限制等。为了制造方便和降低成本，倒立摆的结构尺寸和电机功率都尽量要求最小，行程限制对倒立摆的摆起影响尤为突出，容易出现小车的撞边现象。

1.4 控制器设计方法

控制器的设计是倒立摆系统的核心内容，因为倒立摆是一个绝对不稳定的系统，为使其保持稳定并且可以承受一定的干扰，需要给系统设计控制器，目前典型的控制器设计理论有：PID 控制、根轨迹以及频率响应法、状态空间法、最优控制理论、模糊控制理论、神经网络控制、拟人智能控制、鲁棒控制方法、自适应控制，以及这些控制理论的相互结合组成更加强大的控制算法。

1.5 倒立摆实验

本书中可以完成的倒立摆实验如下：

表 1-1 倒立摆实验

实验类型	序号	实验名称	实验设备
基础实验	1.	编码器原理及使用实验	倒立摆各系列
	2.	Matlab Simulink 环境下电机控制实现	倒立摆各系列
建模实验	3.	直线倒立摆建模、仿真及实验	一级倒立摆
	4.	直线一级顺摆的建模与分析	一级倒立摆
	5.	直线两级倒立摆建模与分析	两级倒立摆
	6.	直线三级倒立摆建模与分析	三级倒立摆
控制实验	7.	直线一级倒立摆根轨迹控制实验	一级倒立摆
	8.	直线一级倒立摆频率响应控制实验	一级倒立摆
	9.	直线一级倒立摆 PID 控制实验	一级倒立摆
	10.	直线一级顺摆的 PID 控制仿真与实验	一级倒立摆
	11.	直线一级顺摆的 LQR 控制仿真与实验	一级倒立摆
	12.	状态空间极点配置控制实验	一级倒立摆
	13.	线性二次最优控制 LQR 控制实验	一级倒立摆
	14.	直线两级倒立摆 LQR 控制实验	两级倒立摆

注：直线三级倒立摆可以改装成两级倒立摆，直线两级倒立摆可以改装成一级倒立摆。

第2章 运动控制基础实验

本实验的目的是让实验者熟悉角度编码器的基本原理，掌握利用计算机和运动控制卡对编码器进行数据采集的方法，以及利用计算机对电机进行控制的基本原理和方法，了解机电一体化的两个重要内容：传感技术和运动控制技术。以便顺利的完成倒立摆的各项实验。

2.1 编码器原理及使用实验

2.1.1 编码器原理

旋转编码器是一种角位移传感器，它分为光电式、接触式和电磁感应式三种，其中光电式脉冲编码器是闭环控制系统中最常用的位置传感器。

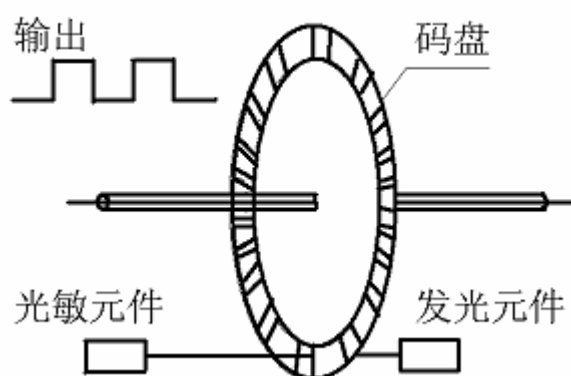


图 2-1 光电编码器原理示意图

旋转编码器有增量编码器和绝对编码器两种，图 2-1 为光电式增量编码器示意图，它由发光元件、光电码盘、光敏元件和信号处理电路组成。当码盘随工作轴一起转动时，光源透过光电码盘上的光栏板形成忽明忽暗的光信号，光敏元件把光信号转换成电信号，然后通过信号处理电路的整形、放大、分频、记数、译码后输出。为了测量出转向，使光栏板的两个狭缝比码盘两个狭缝距离小 $1/4$ 节距，这样两个光敏元件的输出信号就相差 $\pi/2$ 相位，将输出信号送入鉴向电路，即可判断码盘的旋转方向。

光电式增量编码器的测量精度取决于它所能分辨的最小角度 α (分辨角、分辨率)，而这与码盘圆周内所分狭缝的线数有关。

$$\alpha = \frac{360^\circ}{n}$$

其中 n ——编码器线数。

由于光电式脉冲编码盘每转过一个分辨角就发出一个脉冲信号，因此，根据脉冲数目可得出工作轴的回转角度，由传动比换算出直线位移距离；根据脉冲频率可得工作轴的转速；根据光栏板上两条狭缝中信号的相位先后，可判断光电码盘的正、反转。

绝对编码器通过与位数相对应的发光二极管和光敏二极管对输出的二进制码来检测旋转角度。

与增量编码器原理相同，用于测量直线位移的传感器是光栅尺。

由于光电编码器输出的检测信号是数字信号，因此可以直接进入计算机进行处理，不需放大和转换等过程，使用非常方便，因此应用越来越广泛。

2.1.2 角度换算

对于线数为 n 的编码器，设信号采集卡倍频数为 m ，则有角度换算关系为：

$$\phi = \frac{2 \times 3.14}{nm} N \text{ (弧度)} \text{ 或 } \phi = \frac{360^\circ}{nm} N \text{ (度)}$$

式中 ϕ ——为编码器轴转角；

N ——编码器读数

对于电机编码器，在倒立摆使用中需要把编码器读数转化为小车的水平位置，以下转换关系：

$$l = \frac{3.14 \times \Phi}{nm} N$$

式中 l ——小车位移；

Φ ——同步带轮直径

2.1.3 编码器使用实验

本实验对象为倒立摆系统上的光电式旋转编码器，在充分理解以上实验原理的基础上进行下列实验。

实验一 摆杆角度检测实验

按以下实验步骤完成在 MATLAB 下的摆杆角度检测实验，注意，在使用之前请仔细阅读倒立摆的相关使用手册和熟悉 MATLAB 部分知识，确定 MATLAB 已经安装好实时控制工具箱和 VC 编译环境（参见《固高 MATLAB 实时控制软件用户手册》）。

- 1) 打开 MATLAB 以及 Simulink 环境：

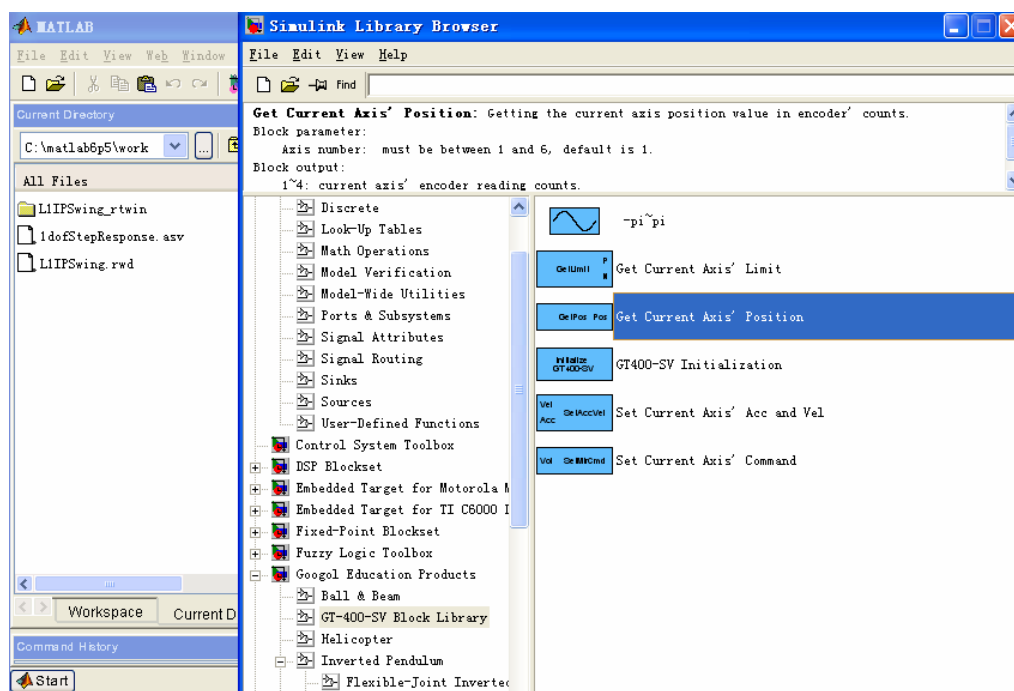

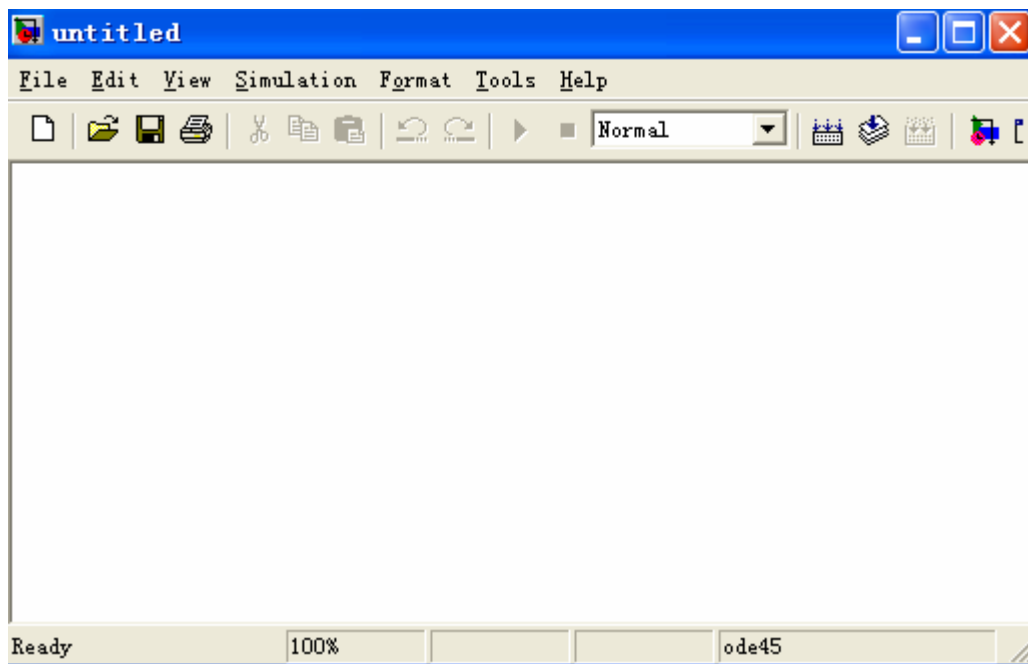
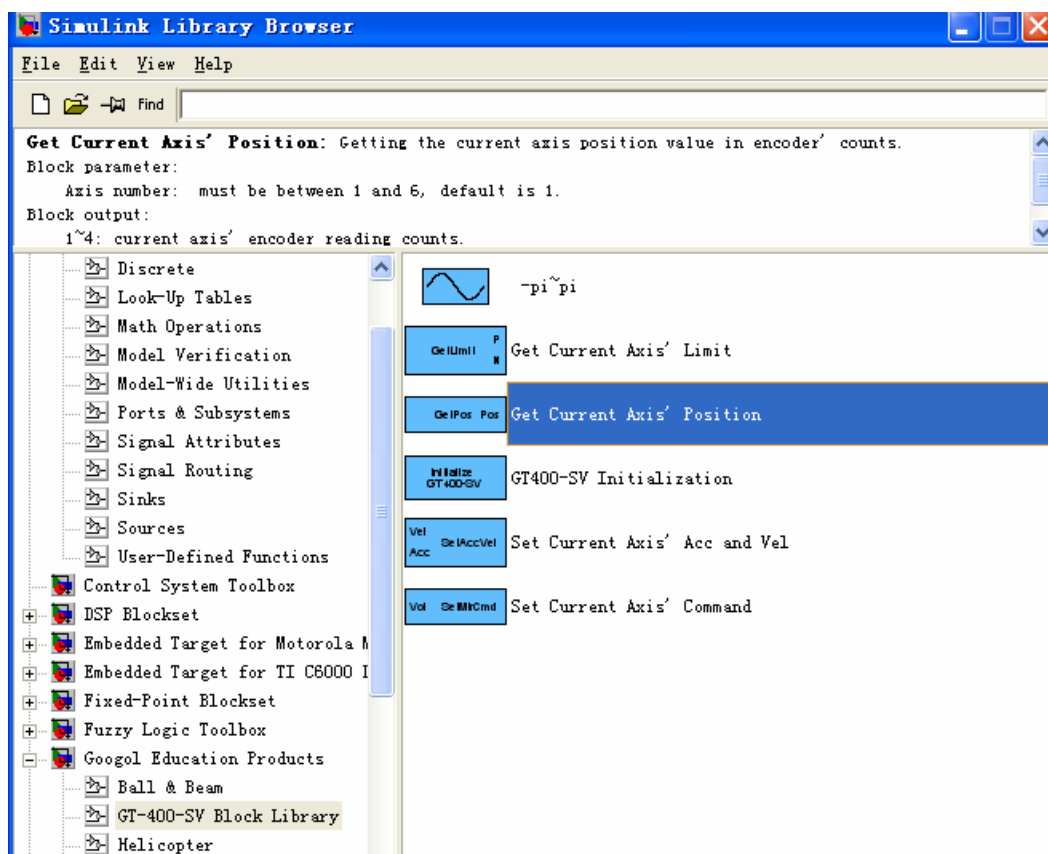


图 2-2 MATLAB Simulink 环境

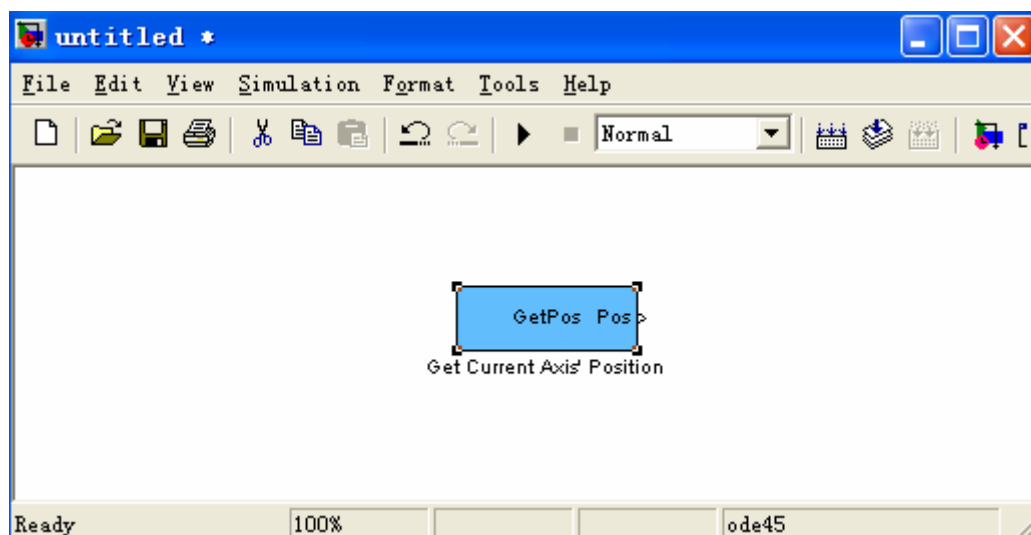
2) 在窗口的左上角点击 “” 建立一个新窗口:



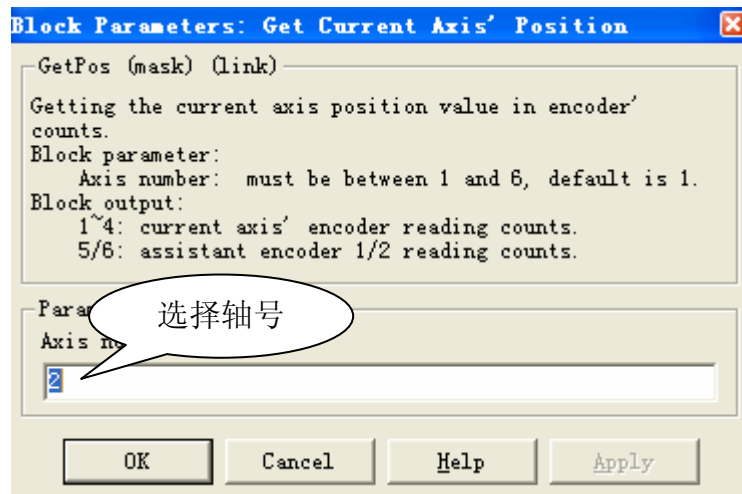
3) 在 Simulink 窗口中, 打开 “Googol Education Products\GT-400-SV Block Library” 如下图所示:



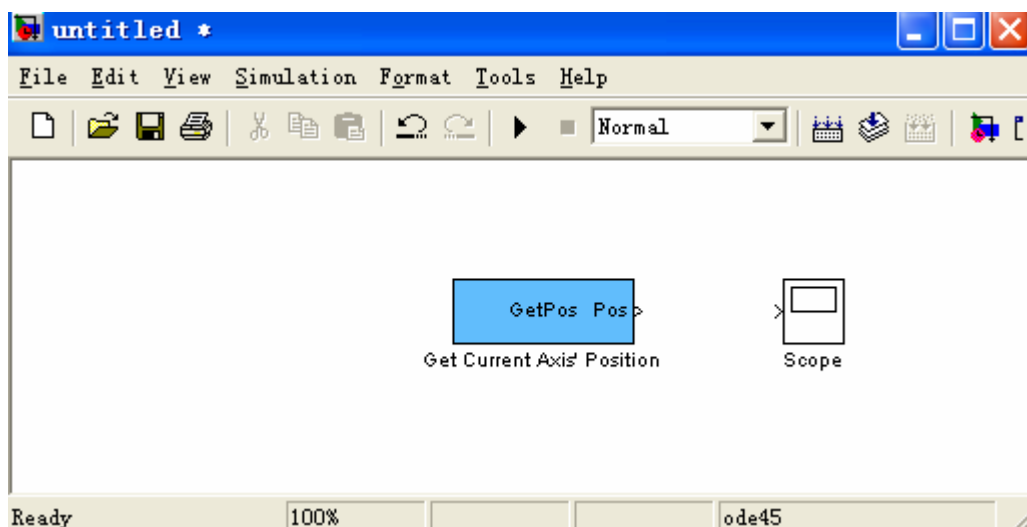
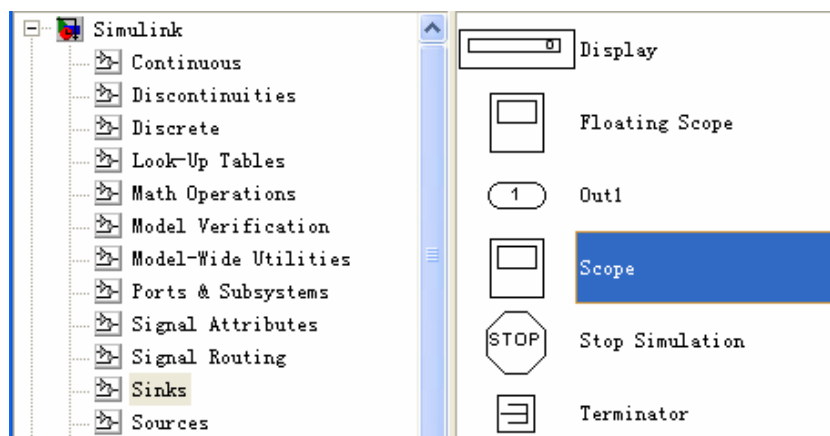
4) 在“Get Current Axis's Position”上点击鼠标左键并将模块拉到（以下简称为“拉”）刚才新建的窗口“untitled”中：



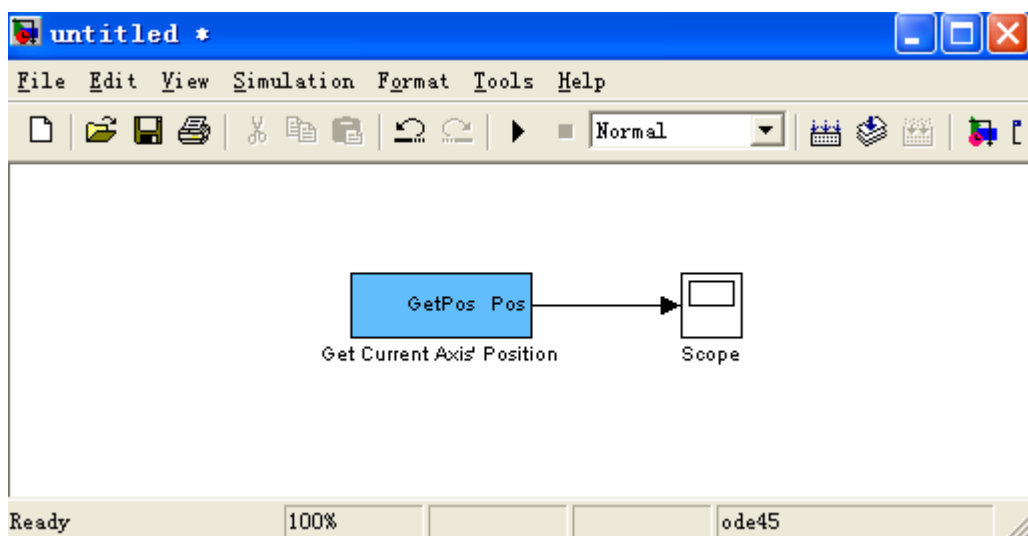
5) 双击“GetPos”模块，打开如下窗口，并选择轴号为“2”，即第一级摆杆连接的编码器，此编码器固定于小车上。



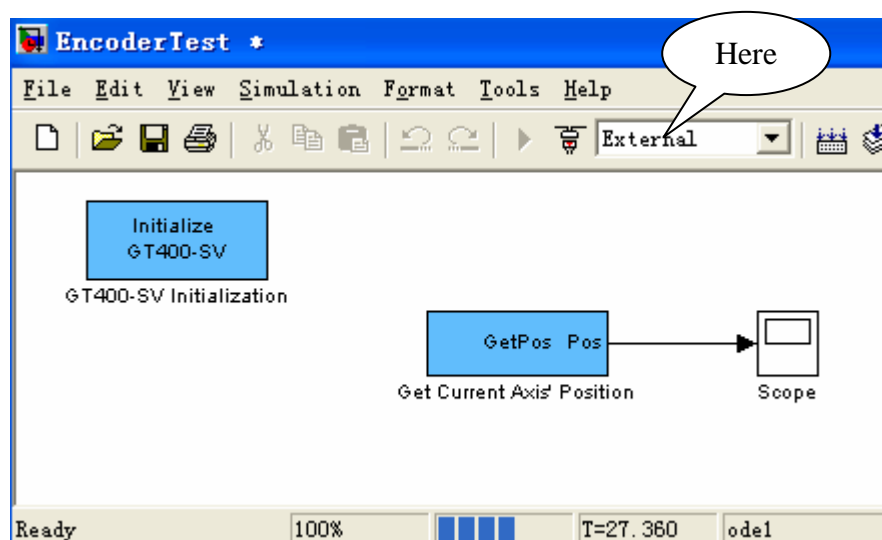
6) 从“Simulink\Sinks”中拉一个“Scope”到“untitled”窗口中:



7) 连接两个模块（移动鼠标到“<”上，鼠标箭头变成“+”，按下鼠标左键并移动到“>”上，松开鼠标）:

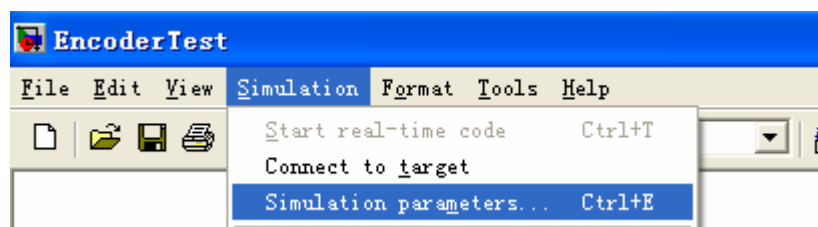


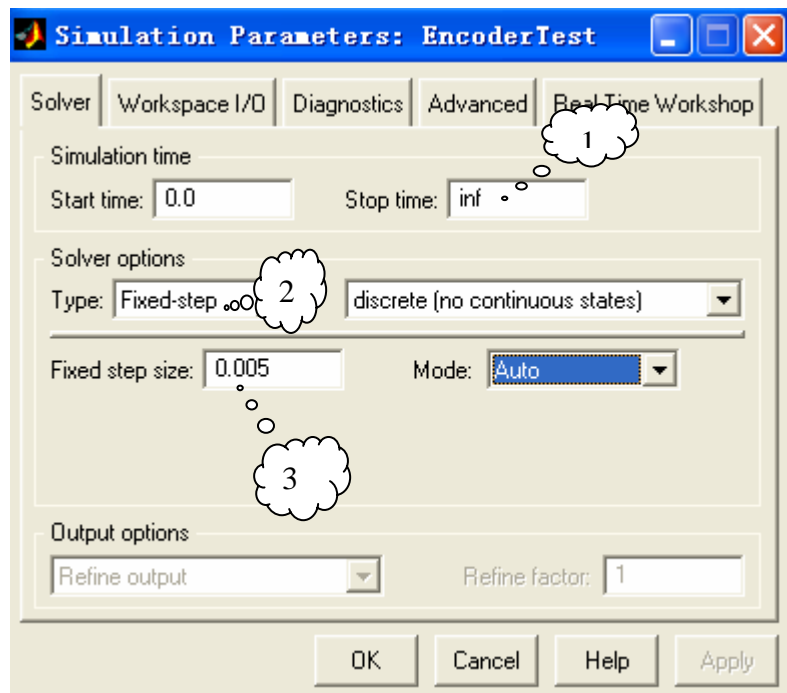
8) 在“Googol Education Products\GT-400-SV Block Library”中拉一个“GT400-SV Initialization”模块到窗口中：



9) 选择上图中上方的“Normal”为“External”：

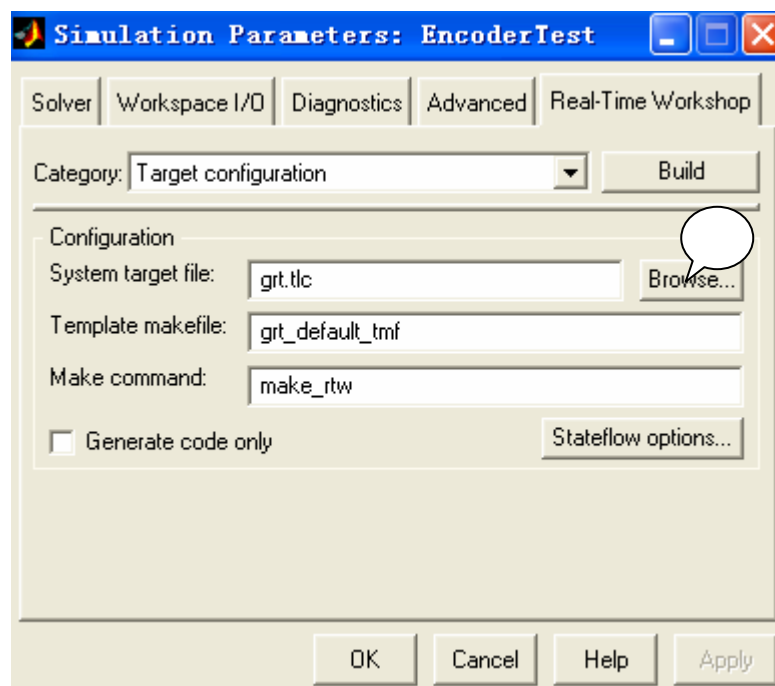
10) 将文件保存为“EncoderTest”，点击菜单“Simulation\Simulation Parameters”设置参数：



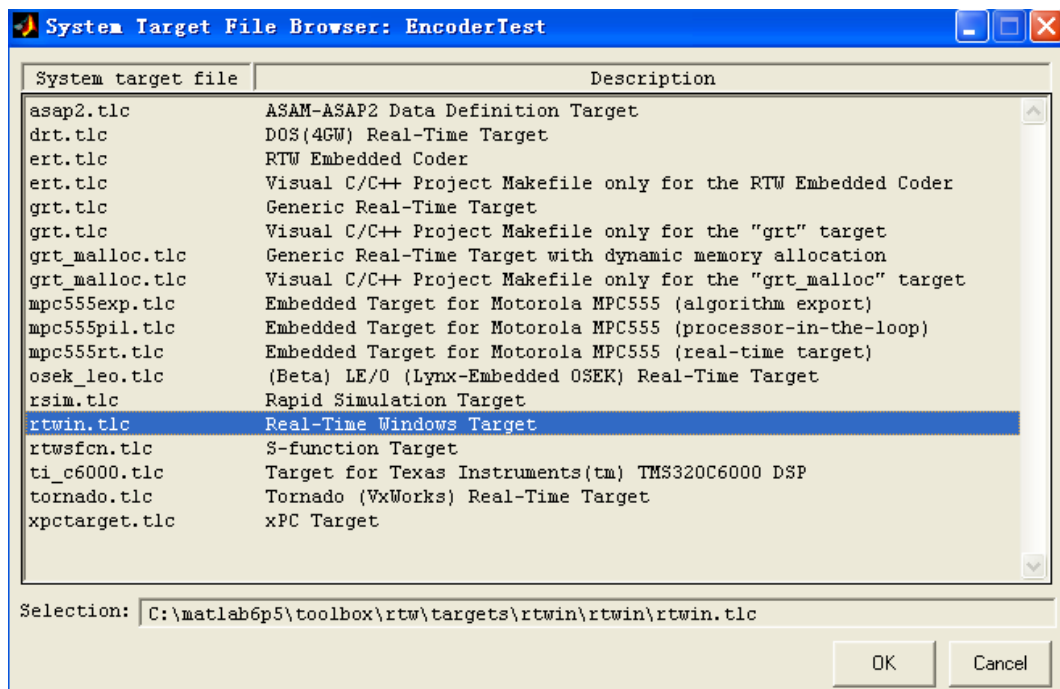


修改“Simulation time”和“Solver options”如上图所示，其中仿真时间“inf”表示无穷长，步长设置为0.005s。

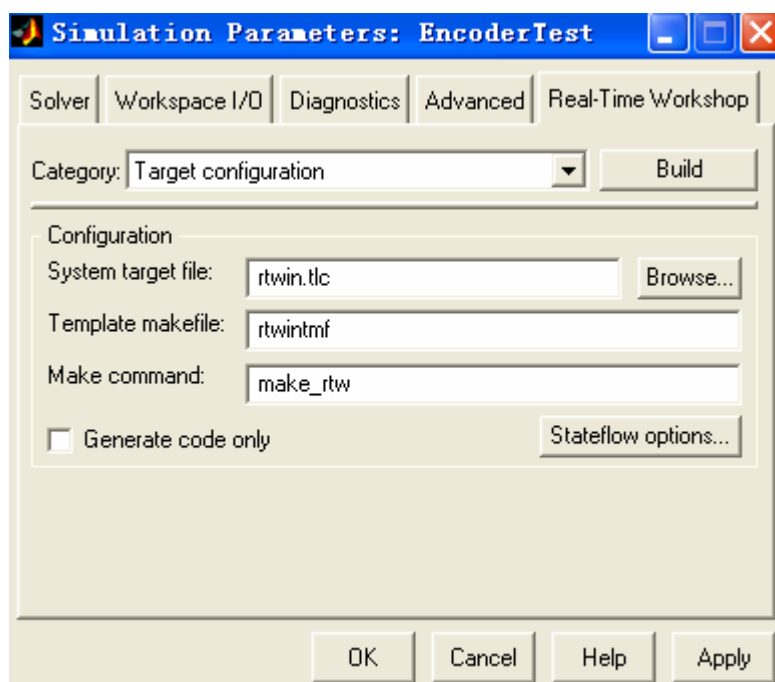
点击“Real-Time Workshop”打开如下所示窗口：



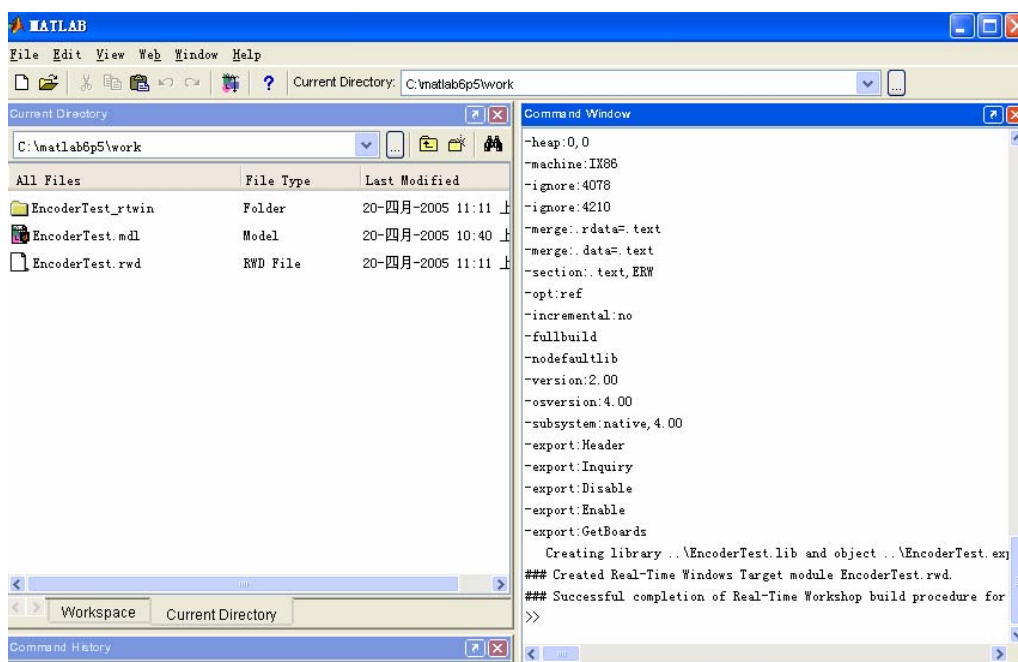
11) 点击“Browse”修改设置为“Real-Time Windows Target”



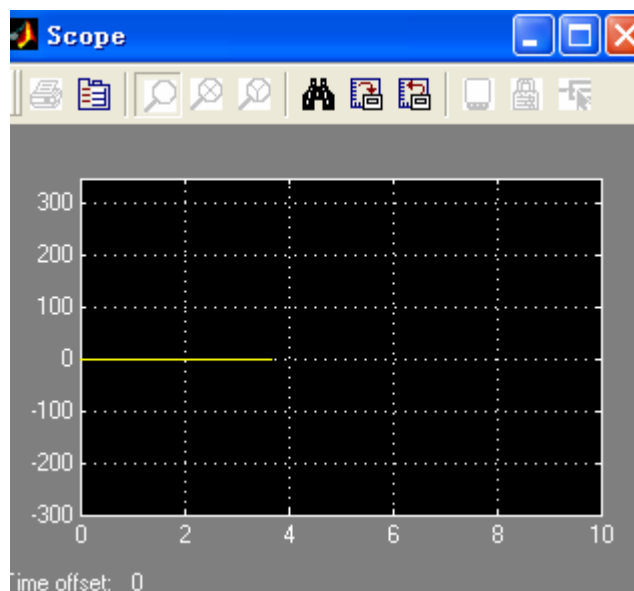
12) 点击“OK”如下图所示:




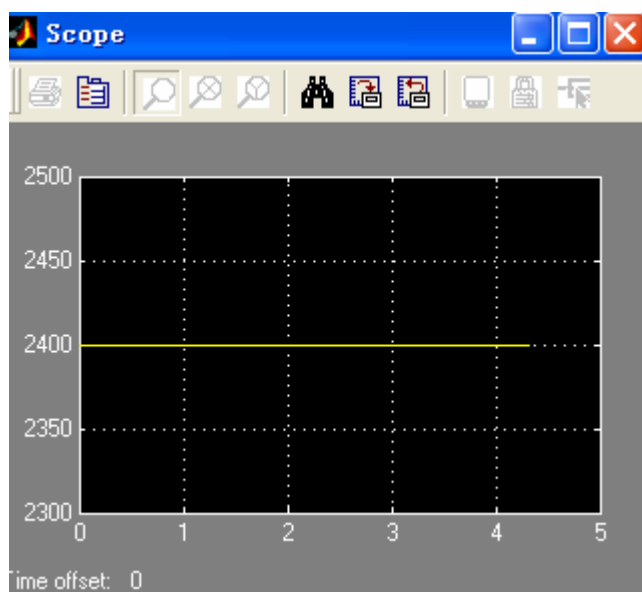
13) 点击“”编译程序，在 Command 窗口中会有编译信息显示:



- 14) 打开电控箱电源;
- 15) 点击 “” 连接程序;
- 16) 点击 “” 运行程序;
- 17) 双击 “Scope” 模块观察数据:



- 18) 手动逆时针转动摆杆一圈, 观察显示结果, 在数据超出显示范围时, 点击 “” 进行缩放。



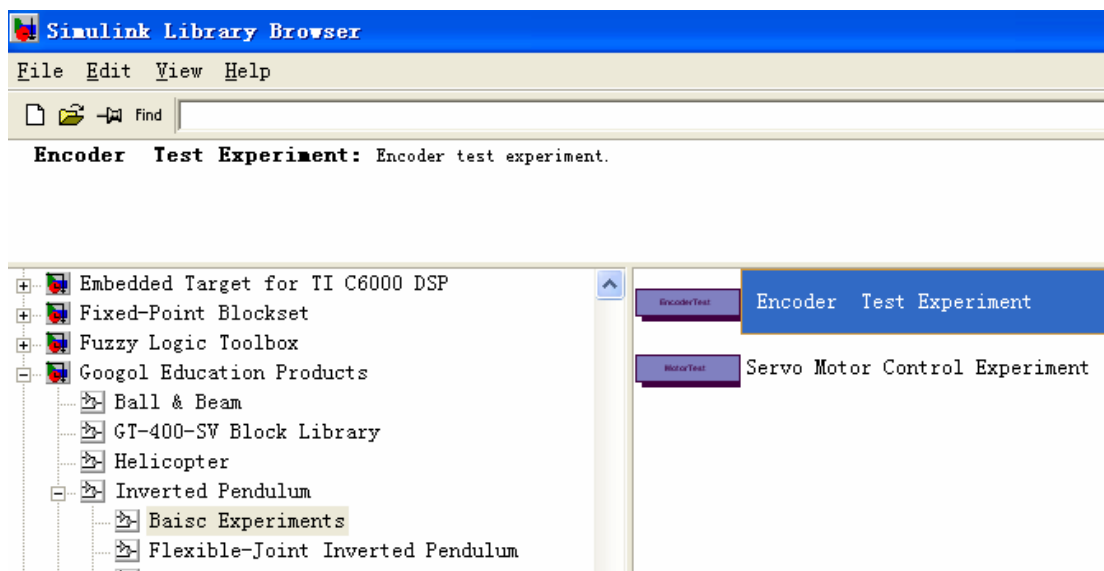
从图中可以看出, 编码器读数为 2400, 等于编码器的线数(600)的四倍 (板卡 4 倍频), 顺时针或逆时针转动摆杆, 观察读数和摆杆实际角度。

19) 记录实验结果, 分析实验数据并完成实验报告。

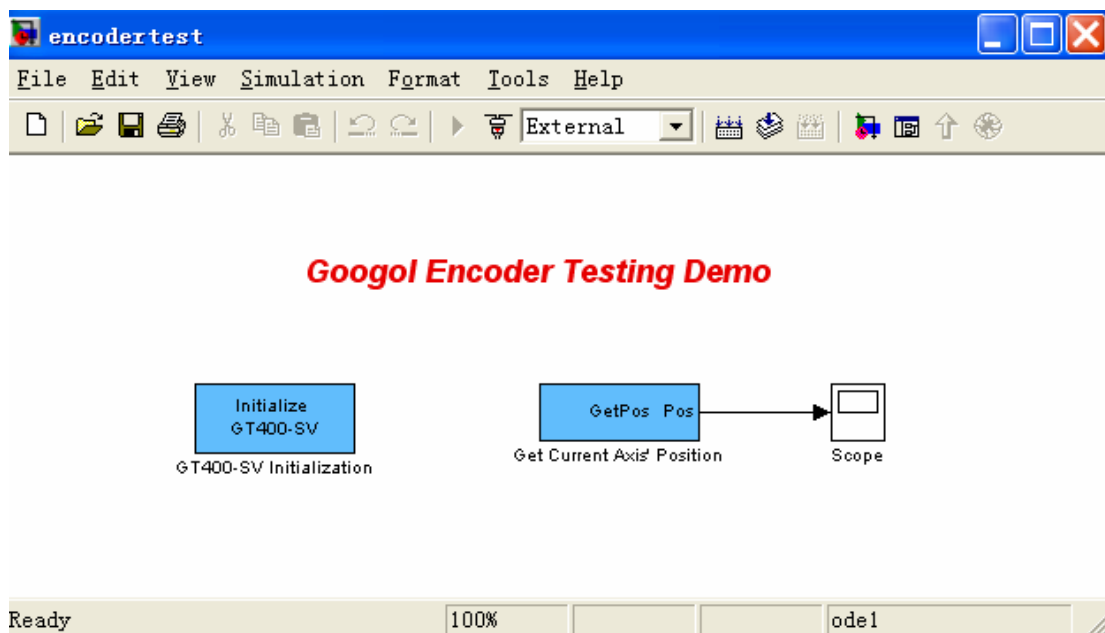
备注: 具体模型请参见 *EncoderTest.mdl*, 其路径如下:

“matlabroot\toolbox\GoogolTech\InvertedPendulum\”, 也可以在 Simulink 环境中打开模型:

进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Basic Experiments” 中的 “Encoder Test Experiment” 如下图所示:



系统模型如下:

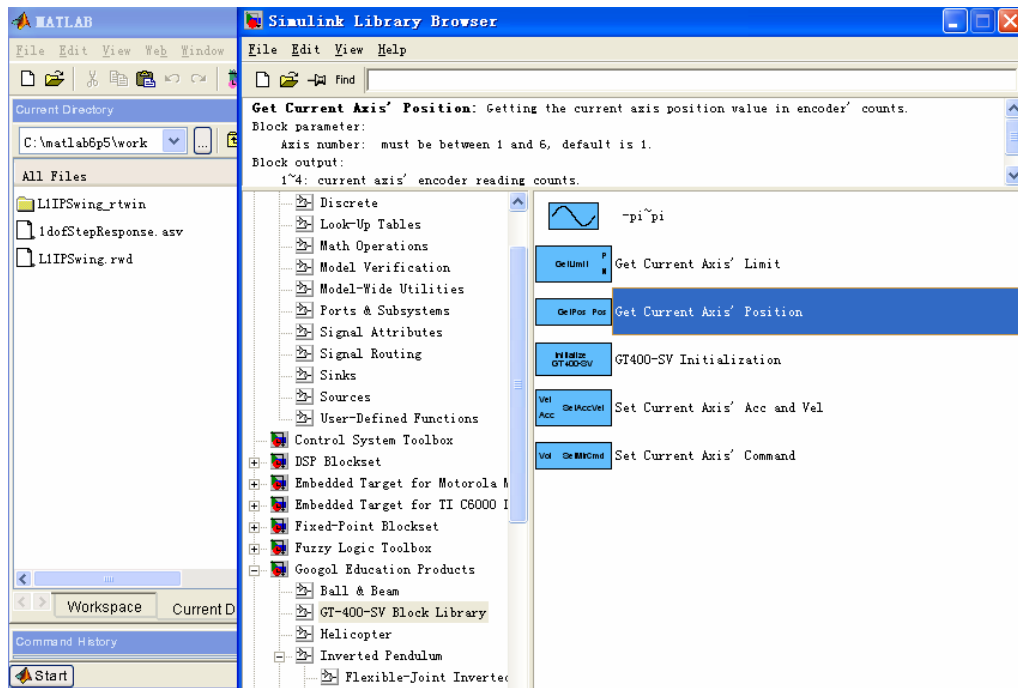



2.2 Matlab Simulink 环境下电机控制实现

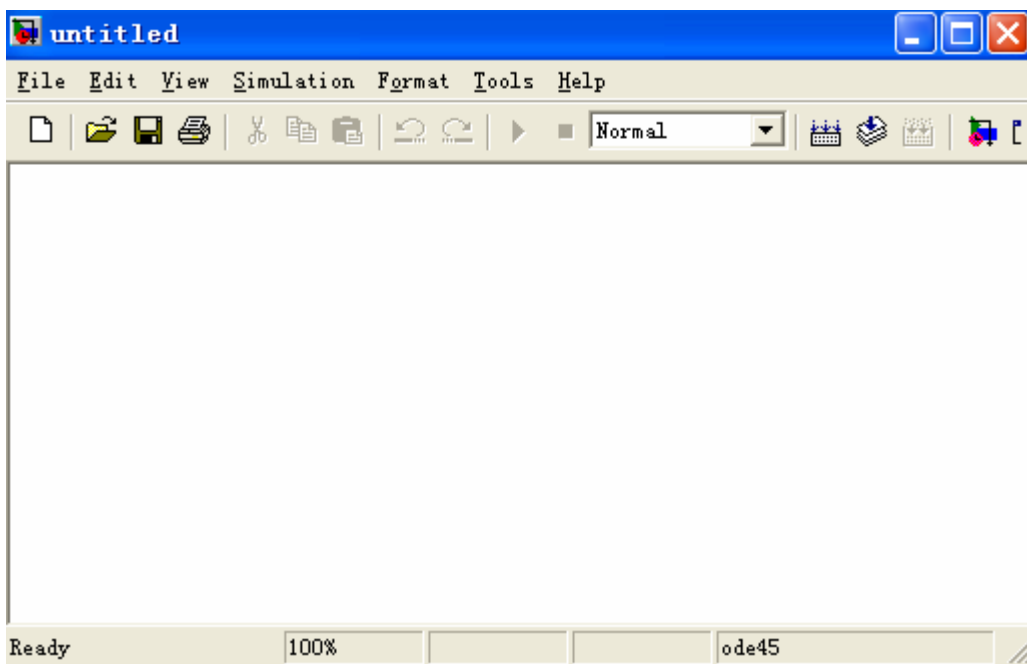
本实验对象为倒立摆系统上的交流伺服电机。将运动控制器当前轴设定成速度控制模式，用户需要设定最大速度和加速度两个参数。在该模式下，开始运动时将以设定的加速度连续加速到设定的最大速度，运动方向由速度的符号确定，即正速度产生正向运动，而负速度产生负向运动。在运动过程中，这两个运动参数可以随时修改。

实验步骤如下：

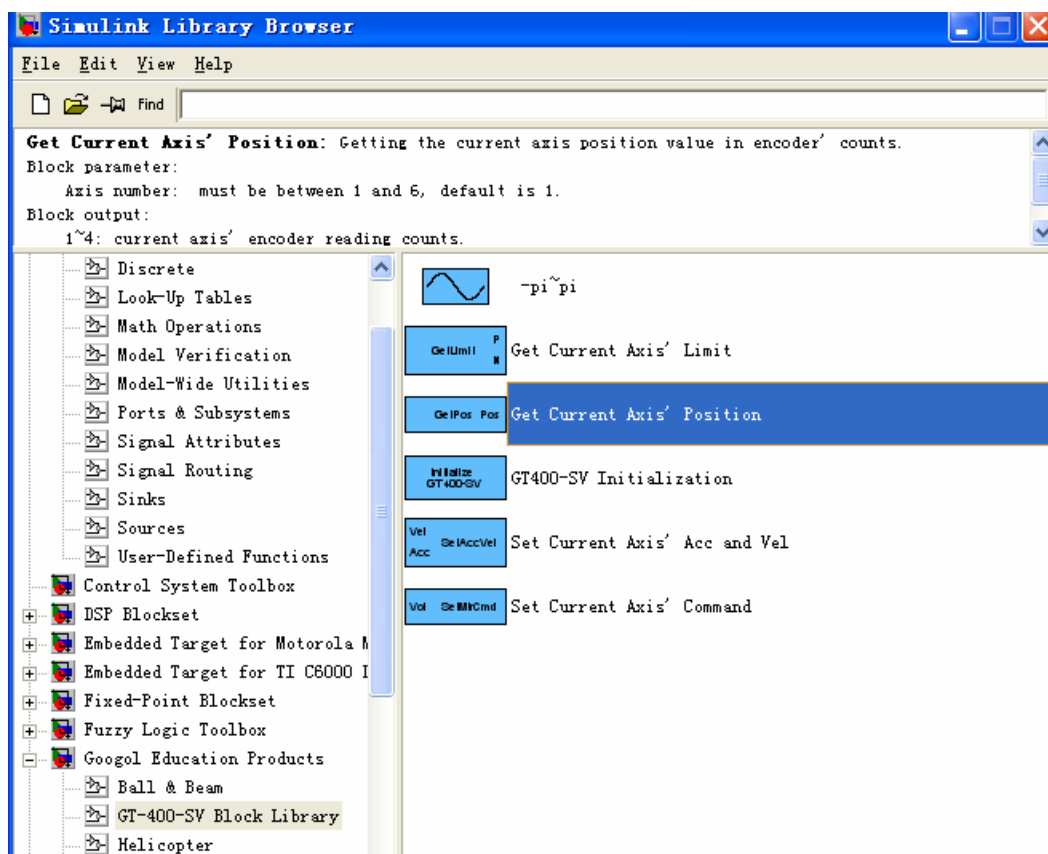
- 1) 打开 MATLAB/Simulink 仿真环境；



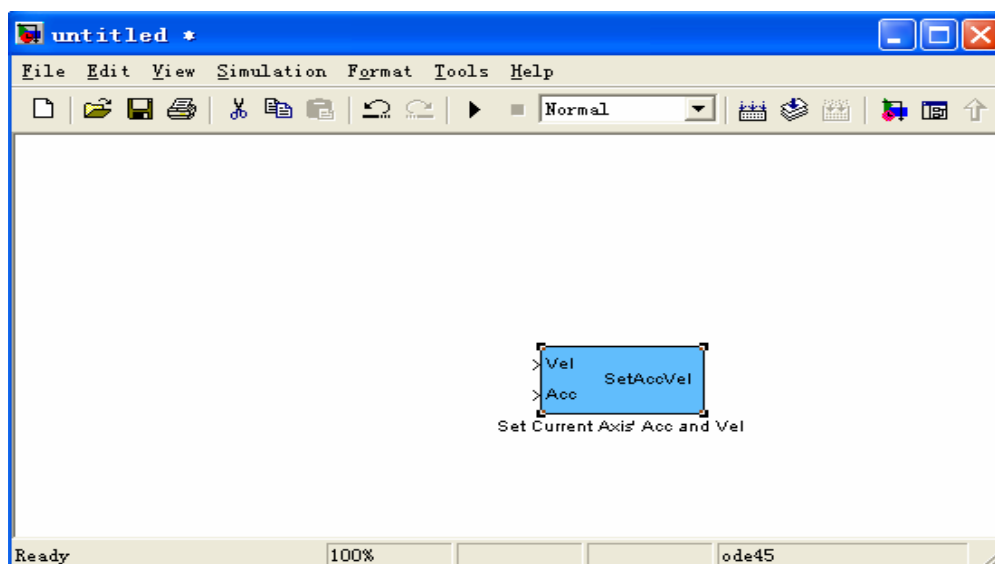
2) 在窗口的左上角点击 “” 新建一个 “Model” 窗口：



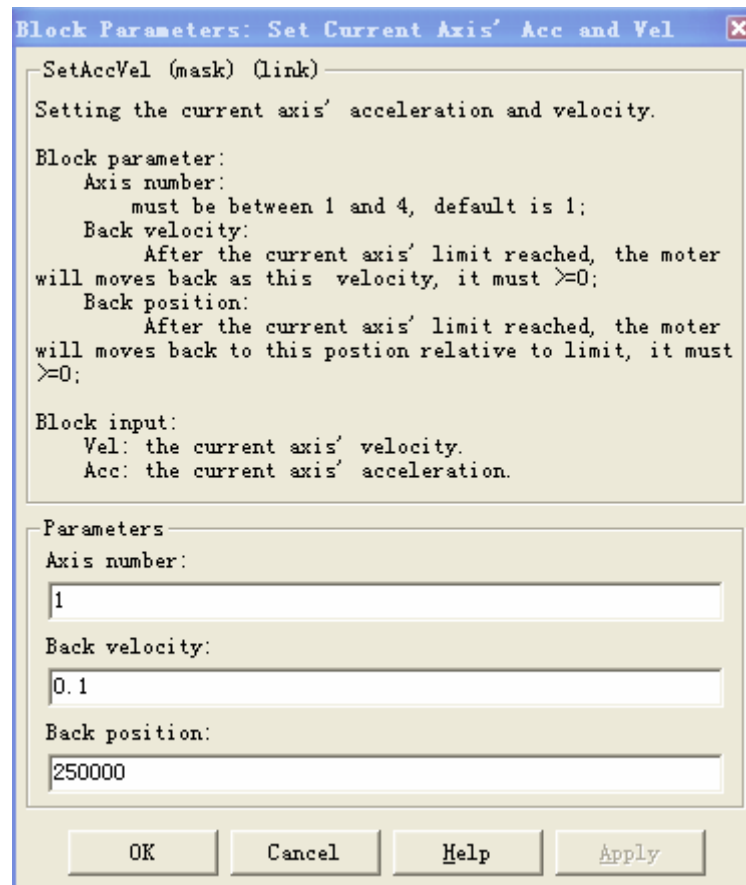
3) 在 “Simulink Library Browse” 窗口中，打开 “Googol Education Products\GT-400-SV Block Library”，如下图所示：



- 4) 拉“Set Current Axis's Acc and Vel”模块到刚才新建的“untitled”窗口中:



- 5) 双击“SetAccVel”模块，打开如下窗口，参数设置如下图:

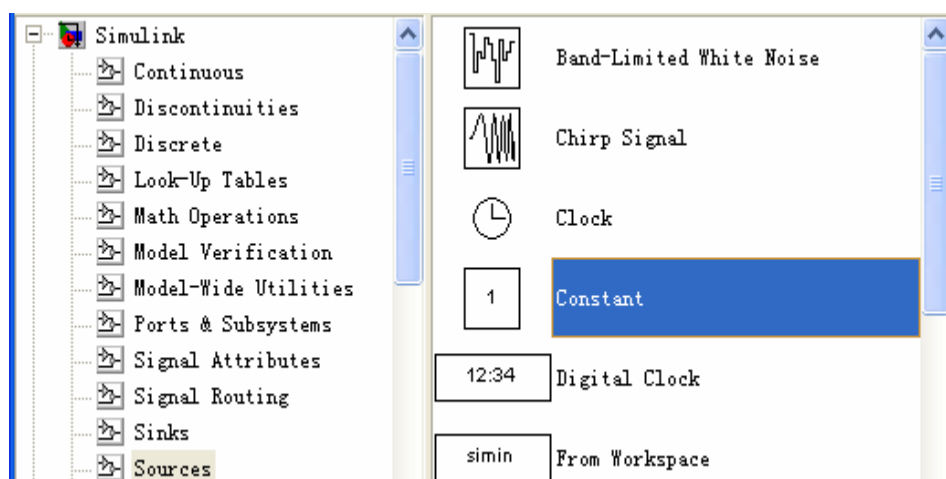


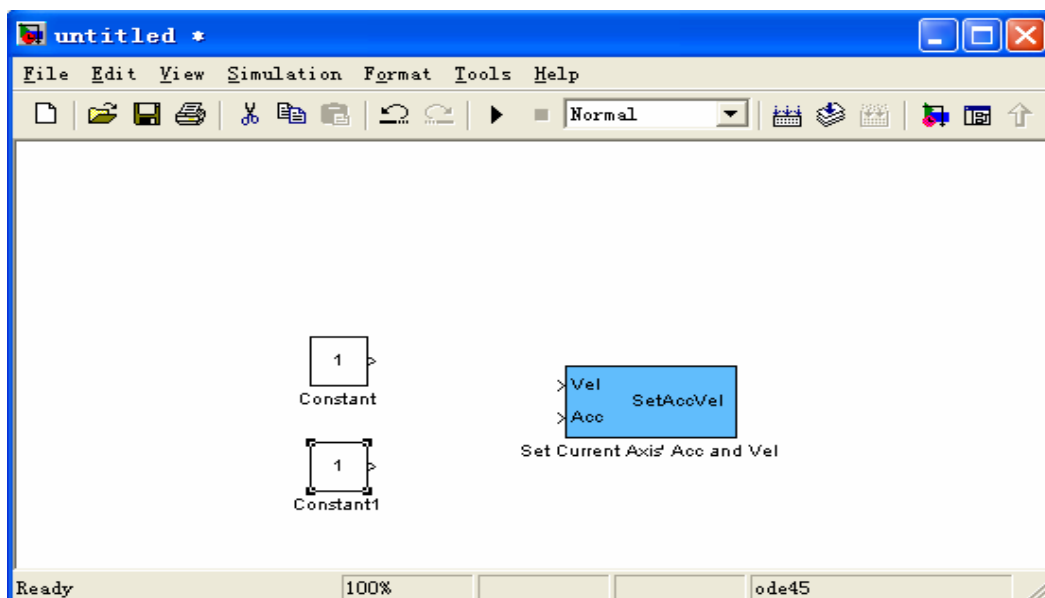
“Axis number”: 选择轴号为“1”，即小车电机所使用的轴。

“Back velocity”: 此参数设定小车限位后反方向运动的速度，单位为米/秒。

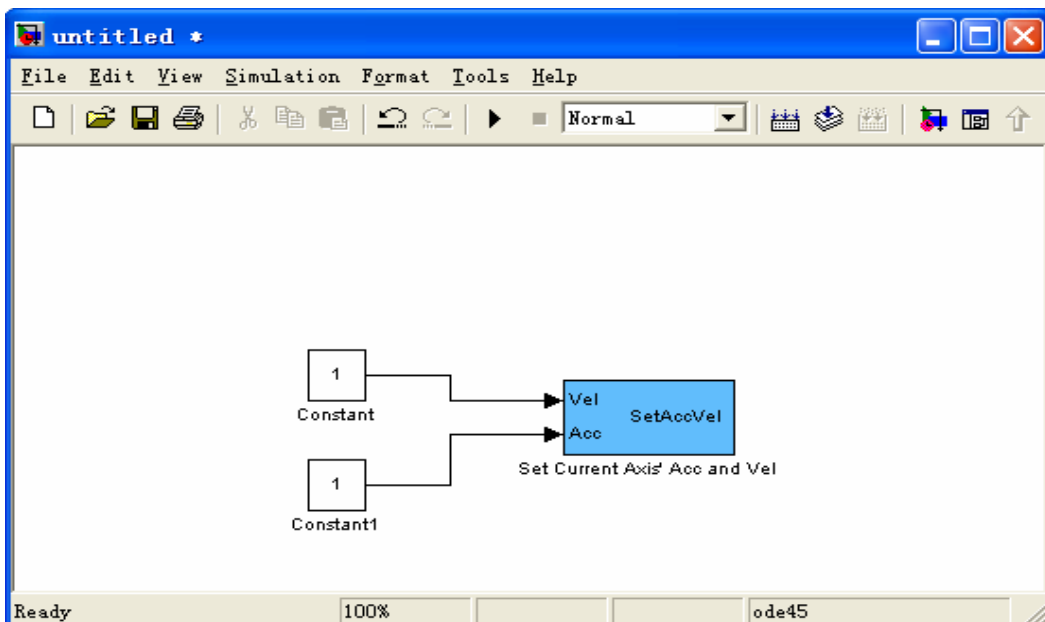
“Back position”: 此参数设定小车限位后反方向运动，单位为脉冲数。

6) 从“Simulink\Souces”中拉两个“Constant”到“untitled”窗口中:

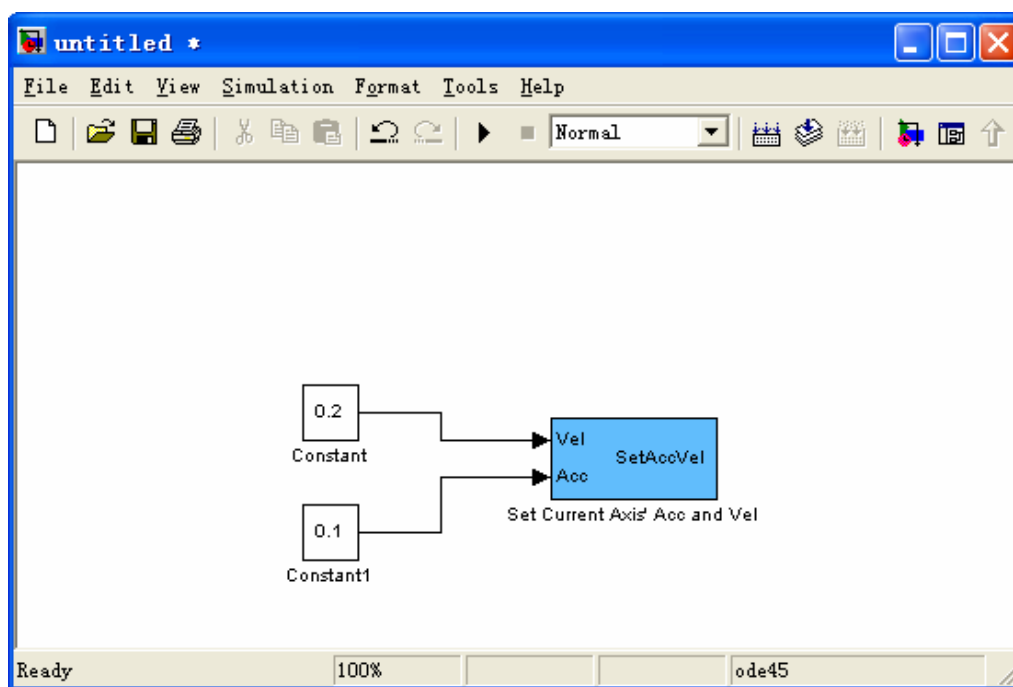




7) 按下图连接三个模块:

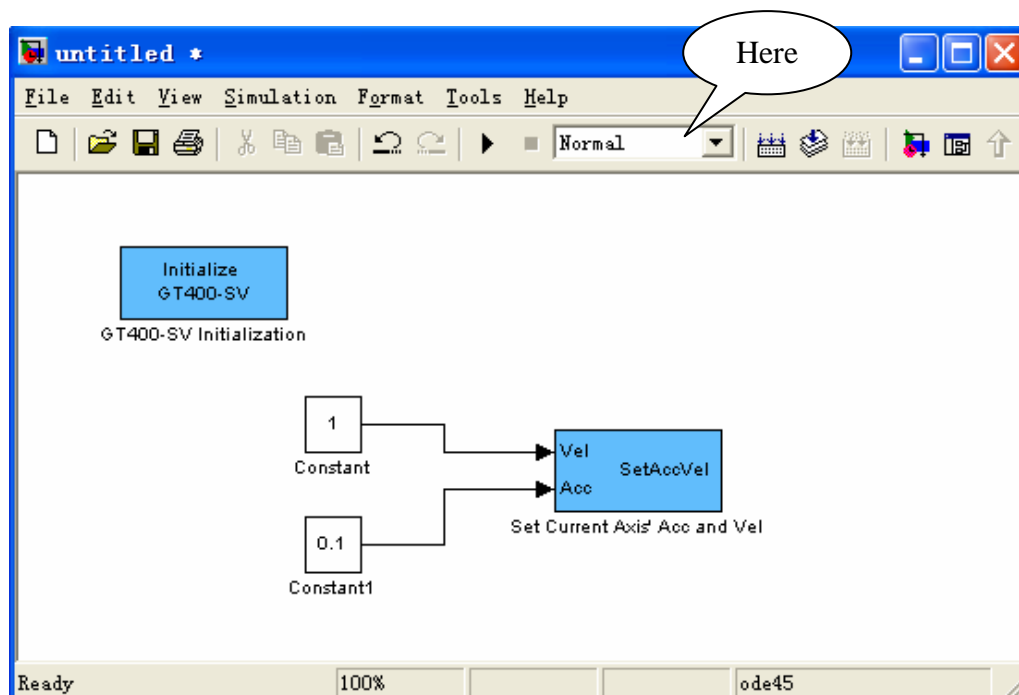


8) 设置两个“Constant”模块参数如下:



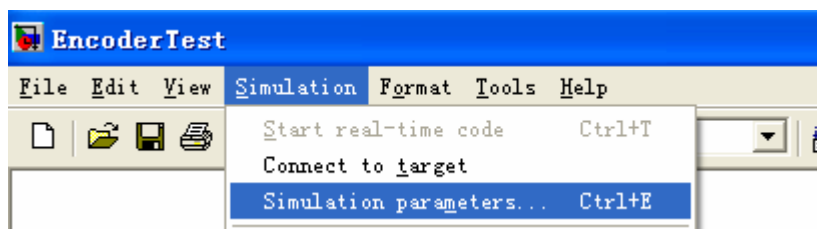
设置小车的运动速度参数为0.2米/秒，加速度为0.1米/秒²。

- 9) 在“Googol Education Products\GT-400-SV Block Library”中拉一个“GT400-SV Initialization”模块到窗口中：



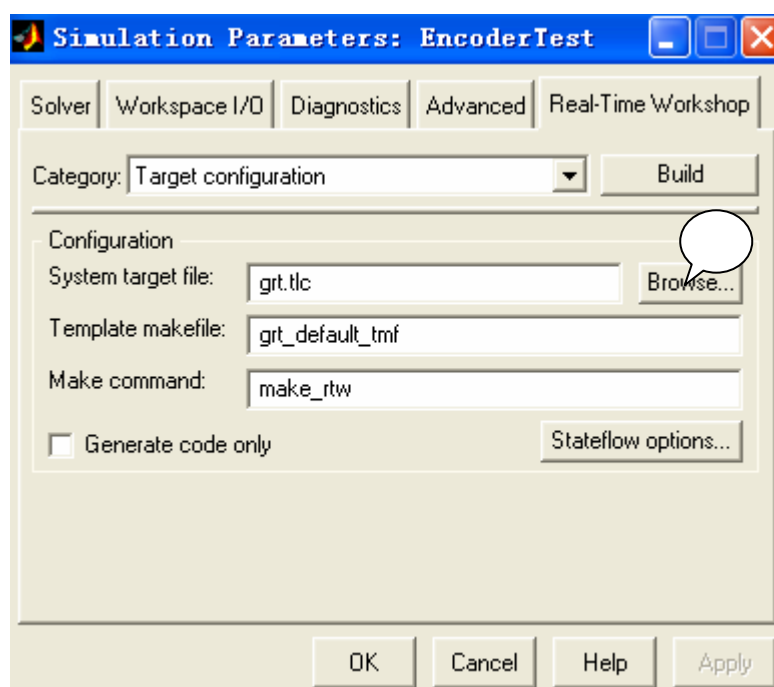
- 10) 按上图选择仿真模式为“External”。

- 11) 将文件保存为“MoterTest”，点击菜单“Simulation\Simulation Parameters”设置参数。

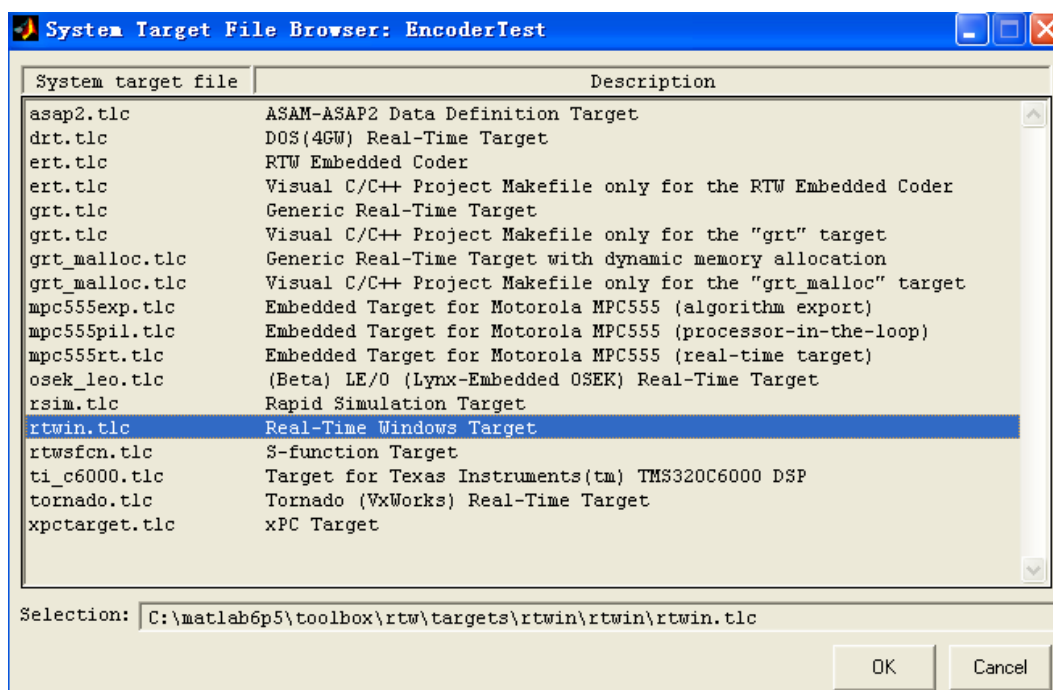


修改“Simulation time”和“Solver options”如上图所示，其中仿真时间“inf”表示无穷长，步长设置为0.005s。

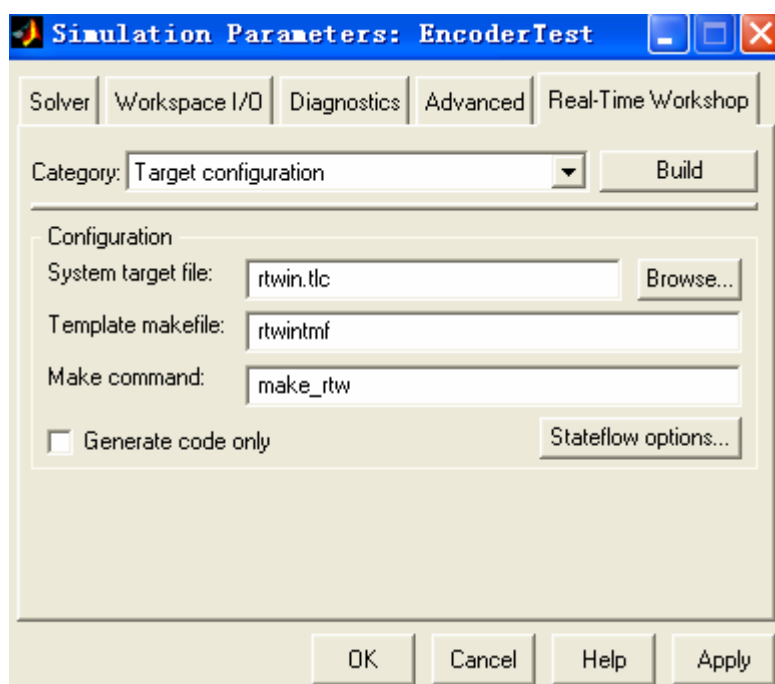
点击“Real-Time Workshop”打开如下所示窗口：



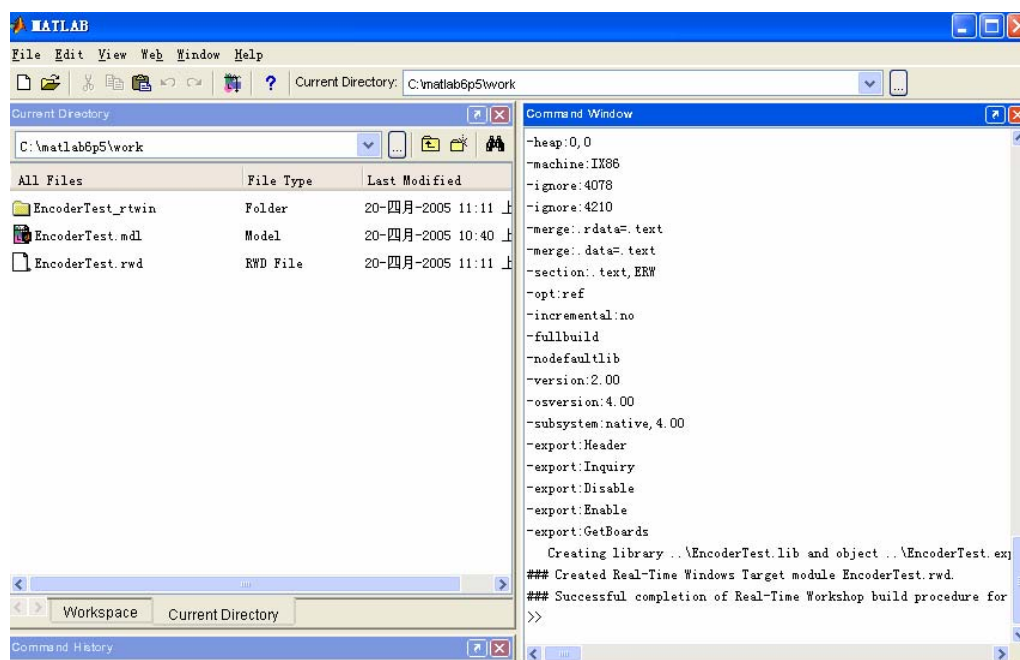
12) 点击“Browse”修改设置为“Real-Time Windows Target”





13) 点击“OK”如下图所示:



14) 点击“”编译程序，在 Command 窗口中会有编译信息显示:

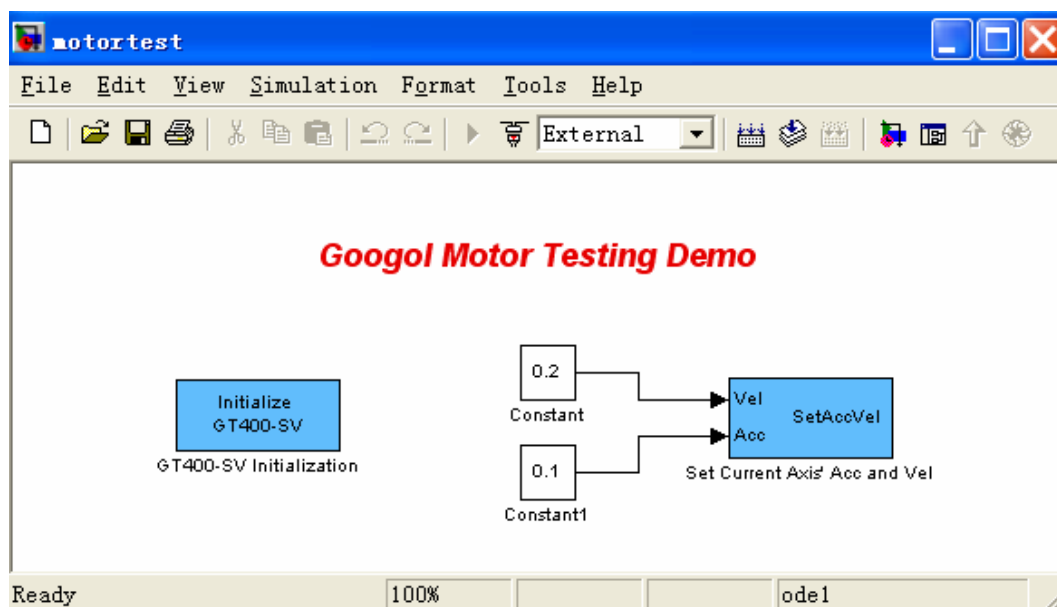


- 15) 打开电控箱电源;
- 16) 点击 “ ” 连接程序;
- 17) 点击 “ ” 运行程序;
- 18) 观察小车的运动轨迹, 记录实验结果。

修改最大速度和加速度两个参数, 观察小车的运动轨迹, 记录实验结果并完成实验报告。

参考模型:

进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products”, 打开 “Inverted Pendulum\Basic Experiments” 中的 “Servo Motor Control Experiment” 如下图所示:



第3章 直线倒立摆建模、仿真及实验

本实验的目的是让实验者掌握对实际系统进行建模的方法，熟悉利用 MATLAB 对系统模型进行仿真，利用学习的控制理论对系统进行控制器的设计，并对系统进行实际控制实验，对实验结果进行观察和分析，非常直观的感受控制器的控制作用。

3.1 直线一级倒立摆

直线一级倒立摆由直线运动模块和一级摆体组件组成，是最常见的倒立摆之一，见图 1-1。

3.1.1 直线一级倒立摆的物理模型

系统建模可以分为两种：机理建模和实验建模。实验建模就是通过在研究对象上加上一系列的研究者事先确定的输入信号，激励研究对象并通过传感器检测其可观测的输出，应用数学手段建立起系统的输入—输出关系。这里面包括输入信号的设计选取，输出信号的精确检测，数学算法的研究等等内容。机理建模就是在了解研究对象的运动规律基础上，通过物理、化学的知识和数学手段建立起系统内部的输入—状态关系。

对于倒立摆系统，由于其本身是自不稳定的系统，实验建模存在一定的困难。但是忽略掉一些次要的因素后，倒立摆系统就是一个典型的运动的刚体系统，可以在惯性坐标系内应用经典力学理论建立系统的动力学方程。下面我们采用其中的牛顿—欧拉方法和拉格朗日方法分别建立直线型一级倒立摆系统的数学模型。

3.1.1.1 微分方程的推导

3.1.1.1.1 牛顿力学方法

在忽略了空气阻力和各种摩擦之后，可将直线一级倒立摆系统抽象成小车和匀质杆组成的系统，如图 3-1 所示。

我们不妨做以下假设：

- M 小车质量
- m 摆杆质量
- b 小车摩擦系数
- l 摆杆转动轴心到杆质心的长度
- I 摆杆惯量

F 加在小车上的力
 x 小车位置

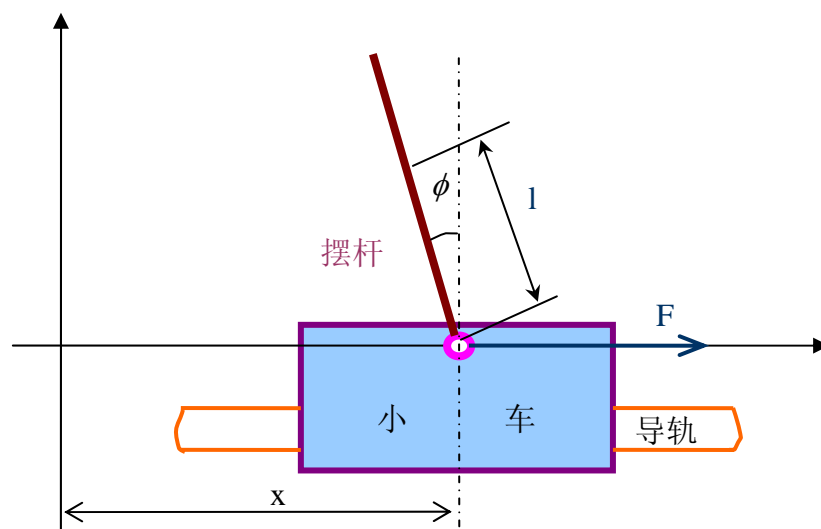


图 3-1 直线一级倒立摆模型

ϕ 摆杆与垂直向上方向的夹角

θ 摆杆与垂直向下方向的夹角（考虑到摆杆初始位置为竖直向下）

图是系统中小车和摆杆的受力分析图。其中， N 和 P 为小车与摆杆相互作用力的水平和垂直方向的分量。

注意：在实际倒立摆系统中检测和执行装置的正负方向已经完全确定，因而矢量方向定义如图所示，图示方向为矢量正方向。

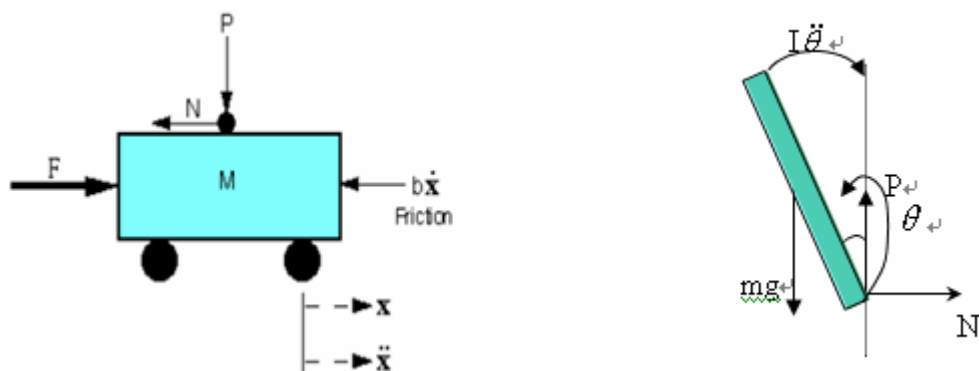


图 3-2 小车及摆杆受力分析

分析小车水平方向所受的合力，可以得到以下方程：

$$M\ddot{x} = F - b\dot{x} - N \quad (3-1)$$

由摆杆水平方向的受力进行分析可以得到下面等式：

$$N = m \frac{d^2}{dt^2} (x + l \sin \theta) \quad (3-2)$$

即:

$$N = m\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta \quad (3-3)$$

把这个等式代入式(3-1)中, 就得到系统的第一个运动方程:

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta = F \quad (3-4)$$

为了推出系统的第二个运动方程, 我们对摆杆垂直方向上的合力进行分析, 可以得到下面方程:

$$P - mg = m \frac{d^2}{dt^2} (l \cos \theta) \quad (3-5)$$

$$P - mg = -ml\ddot{\theta} \sin \theta - ml\dot{\theta}^2 \cos \theta \quad (3-6)$$

力矩平衡方程如下:

$$-Pl \sin \theta - Nl \cos \theta = I\ddot{\theta} \quad (3-7)$$

注意: 此方程中力矩的方向, 由于 $\theta = \pi + \phi$, $\cos \phi = -\cos \theta$, $\sin \phi = -\sin \theta$, 故等式前面有负号。

合并这两个方程, 约去 P 和 N , 得到第二个运动方程:

$$(I + ml^2)\ddot{\theta} + mgl \sin \theta = -ml\ddot{x} \cos \theta \quad (3-8)$$

设 $\theta = \pi + \phi$ (ϕ 是摆杆与垂直向上方向之间的夹角), 假设 ϕ 与 1 (单位是弧度) 相比很小, 即 $\phi \ll 1$, 则可以进行近似处理: $\cos \theta = -1$, $\sin \theta = -\phi$, $(\frac{d\theta}{dt})^2 = 0$ 。

用 u 来代表被控对象的输入力 F , 线性化后两个运动方程如下:

$$\begin{cases} (I + ml^2)\ddot{\phi} - mgl\phi = ml\ddot{x} \\ (M + m)\ddot{x} + b\dot{x} - ml\ddot{\phi} = u \end{cases} \quad (3-9)$$

对式(3-9)进行拉普拉斯变换, 得到

$$\begin{cases} (I + ml^2)\Phi(s)s^2 - mgl\Phi(s) = mlX(s)s^2 \\ (M + m)X(s)s^2 + bX(s)s - ml\Phi(s)s^2 = U(s) \end{cases} \quad (3-10)$$

注意: 推导传递函数时假设初始条件为 0。

由于输出为角度 ϕ , 求解方程组的第一个方程, 可以得到:

$$X(s) = [\frac{(I + ml^2)}{ml} - \frac{g}{s^2}]\Phi(s) \quad (3-11)$$

或
$$\frac{\Phi(s)}{X(s)} = \frac{m l s^2}{(I + m l^2) s^2 - m g l} \quad (3-12)$$

如果令 $v = \ddot{x}$ ，则有：

$$\frac{\Phi(s)}{V(s)} = \frac{m l}{(I + m l^2) s^2 - m g l} \quad (3-13)$$

把上式代入方程组的第二个方程，得到：

$$(M + m) \left[\frac{(I + m l^2)}{m l} - \frac{g}{s} \right] \Phi(s) s^2 + b \left[\frac{(I + m l^2)}{m l} + \frac{g}{s^2} \right] \Phi(s) s - m l \Phi(s) s^2 = U(s) \quad (3-14)$$

整理后得到传递函数：

$$\frac{\Phi(s)}{U(s)} = \frac{\frac{m l}{q} s^2}{s^4 + \frac{b(I + m l^2)}{q} s^3 - \frac{(M + m) m g l}{q} s^2 - \frac{b m g l}{q} s} \quad (3-15)$$

其中 $q = [(M + m)(I + m l^2) - (m l)^2]$

设系统状态空间方程为：

$$\begin{aligned} \dot{X} &= AX + Bu \\ y &= CX + Du \end{aligned} \quad (3-16)$$

方程组 对 $\ddot{x}, \ddot{\phi}$ 解代数方程，得到解如下：

$$\begin{cases} \dot{x} = \dot{x} \\ \ddot{x} = \frac{-(I + m l^2)b}{I(M + m) + M m l^2} \dot{x} + \frac{m^2 g l^2}{I(M + m) + M m l^2} \phi + \frac{(I + m l^2)}{I(M + m) + M m l^2} u \\ \dot{\phi} = \dot{\phi} \\ \ddot{\phi} = \frac{-m l b}{I(M + m) + M m l^2} \dot{x} + \frac{m g l (M + m)}{I(M + m) + M m l^2} \phi + \frac{m l}{I(M + m) + M m l^2} u \end{cases} \quad (3-17)$$

整理后得到系统状态空间方程：

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I + m l^2)b}{I(M + m) + M m l^2} & \frac{m^2 g l^2}{I(M + m) + M m l^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-m l b}{I(M + m) + M m l^2} & \frac{m g l (M + m)}{I(M + m) + M m l^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I + m l^2}{I(M + m) + M m l^2} \\ 0 \\ \frac{m l}{I(M + m) + M m l^2} \end{bmatrix} u$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (3-18)$$

由(3-9)的第一个方程为:

$$(I + ml^2)\ddot{\phi} - mgl\phi = m\ddot{x}$$

对于质量均匀分布的摆杆有:

$$I = \frac{1}{3}ml^2$$

于是可以得到:

$$\left(\frac{1}{3}ml^2 + ml^2\right)\ddot{\phi} - mgl\phi = m\ddot{x}$$

化简得到:

$$\ddot{\phi} = \frac{3g}{4l}\phi + \frac{3}{4l}\ddot{x} \quad (3-19)$$

设 $X = \{x, \dot{x}, \phi, \dot{\phi}\}$, $u' = \ddot{x}$ 则有:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{3g}{4l} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{3}{4l} \end{bmatrix} u'$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u' \quad (3-20)$$

另外, 也可以利用 MATLAB 中 tf2ss 命令对(3-13)式进行转化, 求得上述状态方程。

3.1.1.1.2 拉格朗日方法

下面采用拉格朗日方程建模。

拉格朗日方程为:

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q, \dot{q}) \quad (3-21)$$

其中 L 为拉格朗日算子, q 为系统的广义坐标, T 为系统的动能, V 为系统的势能。

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = f_i \quad (3-22)$$

其中 $i=1,2,3,\dots,n$, f_i 为系统在第 i 个广义坐标上的外力, 在一级倒立摆系统中, 系统的广义坐标有三个广义坐标, 分别为 x, θ_1 。

首先计算系统的动能:

$$T = T_M + T_m$$

其中 T_M, T_m 分别为小车的动能, 摆杆 1 的动能。

小车的动能:

$$T_M = \frac{1}{2} M \dot{x}^2$$

下面计算摆杆的动能:

$T_m = T_m' + T_m''$, 其中 T_m', T_m'' 分别为摆杆的平动动能和转动动能。

设以下变量:

xpend —— 摆杆质心横坐标;

ypend —— 摆杆质心纵坐标

有:

$$\begin{aligned} xpend &= x - l \sin \phi \\ ypend &= l \cos \phi \end{aligned}$$

摆杆的动能为:

$$T_m' = \frac{1}{2} m \left(\left(\frac{d(xpend)}{dt} \right)^2 + \left(\frac{d(ypend)}{dt} \right)^2 \right)$$

$$T_m'' = \frac{1}{2} J_p \dot{\theta}_2^2 = \frac{1}{6} m l^2 \dot{\phi}^2$$

于是有系统的总动能:

$$T_m = T_m' + T_m'' = \frac{1}{2} m \left(\left(\frac{d(xpend)}{dt} \right)^2 + \left(\frac{d(ypend)}{dt} \right)^2 \right) + \frac{1}{6} m l^2 \dot{\phi}^2$$

系统的势能为:

$$V = V_m = m \times g \times ypend = mgl \cos \phi$$

由于系统在 ϕ 广义坐标下只有摩擦力作用, 所以有:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\phi}} - \frac{\partial L}{\partial \phi} = b \dot{x}$$

对于直线一级倒立摆系统，系统状态变量为：

$$\left\{ x, \phi, \dot{x}, \dot{\phi} \right\}$$

为求解状态方程：

$$\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases}$$

需要求解 $\ddot{\phi}$ ，

因此设

$$\ddot{\phi} = f(x, \phi, \dot{x}, \dot{\phi}, \ddot{x})$$

将在平衡位置附近进行泰勒级数展开，并线性化，可以得到：

$$\ddot{\phi} = k_{11}x + k_{12}\phi + k_{13}\dot{x} + k_{14}\dot{\phi} + k_{15}\ddot{x}$$

$$\text{其中 } k_{11} = \left. \frac{\partial f}{\partial x} \right|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0} \quad k_{12} = \left. \frac{\partial f}{\partial \phi} \right|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0}$$

$$k_{13} = \left. \frac{\partial f}{\partial \dot{x}} \right|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0} \quad k_{14} = \left. \frac{\partial f}{\partial \dot{\phi}} \right|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0}$$

$$k_{15} = \left. \frac{\partial f}{\partial \ddot{x}} \right|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0}$$

在计算以上各式前，简单介绍一下 Mathematica（更多内容请参照网站：www.wolfram.com）。

Mathematica 是一种能够在很多方面进行技术计算的、新的符号计算机语言，自从 1988 年发布以来，在使用计算机进行数学计算的领域，产生了深远的影响，在本书中，主要使用了 Mathematica 的符号计算和方程式求解功能。

下面利用 Mathematica 对直线一级倒立摆的建模进行计算：（计算文件请参见“\GLIP2001\建模文件\MathematicaFile”中的“L1DIP.nb”文件。）

PRO 3-1 直线一级倒立摆建模程序

```

xpend = x[t] - l*Sin[phi[t]];
ypend = l*Cos[phi[t]];

tpend = 1/2*m*((D_t xpend)^2 + (D_t ypend)^2) + 1/6*m*l^2*(phi'[t])^2;
Simplify[tpend];
v = m*g*ypend;

lang = tpend - v;
Simplify[lang];

ldad = D_phi'[t] lang;
Simplify[ldad];
fa = D_t ldad - D_phi[t] lang;
Simplify[fa];

Solve[{fa == 0}, {phi''[t]}];
add = phi''[t] /. %;

k11 = D_x[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0
k12 = D_phi[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0
k13 = D_x'[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0
k14 = D_phi'[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0
k15 = D_x''[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0

g = 9.8;
l = 0.25;

Simplify[k12]
Simplify[k15]

```

运行程序（按下 Shift+Enter 组合键运行程序）得到：

$$k11 = 0$$

$$k12 = \frac{3g}{4l}$$

$$k13 = 0$$

$$k14 = 0$$

$$k15 = \frac{3}{4l}$$

设 $X = \{x, \dot{x}, \phi, \dot{\phi}\}$ ，系统状态空间方程为：

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

$$y = CX + Du$$

则有：

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{3g}{4l} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ \frac{3}{4l} \end{bmatrix} u' \quad (3-23)$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u'$$

可以看出，利用拉格朗日方法和牛顿力学方法得到的状态方程的是相同的，不同之处在于，输入 u' 为小车的加速度 \ddot{x} ，而输入 u 为外界给小车施加的力，对于不同的输入，系统的状态方程不一样，对比较简单的直线一级倒立摆，利用牛顿力学的方法计算比较方便和快捷，但对于多级倒立摆，利用拉格朗日方法编程计算会比较方便。

3.1.1.2 系统物理参数

实际系统的模型参数如下：

M	小车质量	1.096 Kg
m	摆杆质量	0.109 Kg
b	小车摩擦系数	0.1N/m/sec
l	摆杆转动轴心到杆质心的长度	0.25m
I	摆杆惯量	0.0034 kg*m*m



注 意

- 在进行实际系统的 MATLAB 仿真时，请用户自行检查系统参数是否与实际系统相符，否则请改用实际参数进行计算。

3.1.1.3 实际系统模型

把上述参数代入，可以得到系统的实际模型。

摆杆角度和小车位移的传递函数：

$$\frac{\Phi(s)}{X(s)} = \frac{0.02725s^2}{0.0102125s^2 - 0.26705} \quad (3-24)$$

摆杆角度和小车加速度之间的传递函数为：

$$\frac{\Phi(s)}{V(s)} = \frac{0.02725}{0.0102125s^2 - 0.26705} \quad (3-25)$$

摆杆角度和小车所受外界作用力的传递函数:

$$\frac{\Phi(s)}{U(s)} = \frac{2.35655s}{s^3 + 0.0883167s^2 - 27.9169s - 2.30942} \quad (3-26)$$

以外界作用力作为输入的系统状态方程:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.0883167 & 0.629317 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.235655 & 27.8285 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.883167 \\ 0 \\ 2.35655 \end{bmatrix} u$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (3-27)$$

以小车加速度作为输入的系统状态方程:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 29.4 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \end{bmatrix} u$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (3-28)$$

需要说明的是,在固高科技所有提供的控制器设计和程序中,采用的都是以小车的加速度作为系统的输入,如果用户需要采用力矩控制的方法,可以参考以上把外界作用力作为输入的各式。

3.1.1.4 系统可控性分析

系统的可控性分析原理可参考《现代控制工程》中第 11 章的控制系统的状态分析内容或其它相关资料。

对于连续时间系统:

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

$$y = CX + Du$$

系统状态完全可控的条件为:当且仅当向量组 $B, AB, \dots, A^{n-1}B$ 是线性无关的,

或 $n \times n$ 维矩阵 $[B:AB:\dots:A^{n-1}B]$ 的秩为 n 。

系统的输出可控性的条件为：当且仅当矩阵

$$[CB: CAB: CA^2B: \dots: CA^{n-1}B: D]$$

的秩等于输出向量 y 的维数。

应用以上原理对系统进行可控性分析，

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 29.4 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

代入上式，并在 MATLAB 中计算：

```
clear;
A=[ 0  1  0  0;
    0  0  0  0;
    0  0  0  1;
    0  0 29.4 0];
B=[ 0  1  0  3]';
C=[ 1  0  0  0;
    0  1  0  0];
D=[ 0 0 ]';
cona=[B A*B A^2*B A^3*B];
cona2=[C*B C*A*B C*A^2*B C*A^3*B D];
rank(cona)
rank(cona2)
```

或直接利用计算可控性矩阵的 *ctrb* 命令和计算可观性的矩阵 *obsv* 命令来计算。

```

Uc=ctrb(A,B);
Vo=obsv(A,C);

rank(Uc)
rank(Vo)

```

可以得到：

```

ans =
     4
ans =
     2

```

可以看出，系统的状态完全可控性矩阵的秩等于系统的状态变量维数，系统的输出完全可控性矩阵的秩等于系统输出向量 y 的维数，所以系统可控，因此可以对系统进行控制器的设计，使系统稳定。

对于以外界作用力作为输入的系统状态方程的可控性分析，读者可以按上述方法自行计算。

3.1.2 系统阶跃响应分析

上面已经得到系统的状态方程，先对其进行阶跃响应分析，在 MATLAB 中键入以下命令：

```

clear;
A=[ 0  1  0  0;
    0  0  0  0;
    0  0  0  1;
    0  0 29.4 0];
B=[ 0  1  0  3]';
C=[ 1  0  0  0;
    0  1  0  0];
D=[ 0 0 ]';

step(A, B ,C ,D)

```

得到如下计算结果：

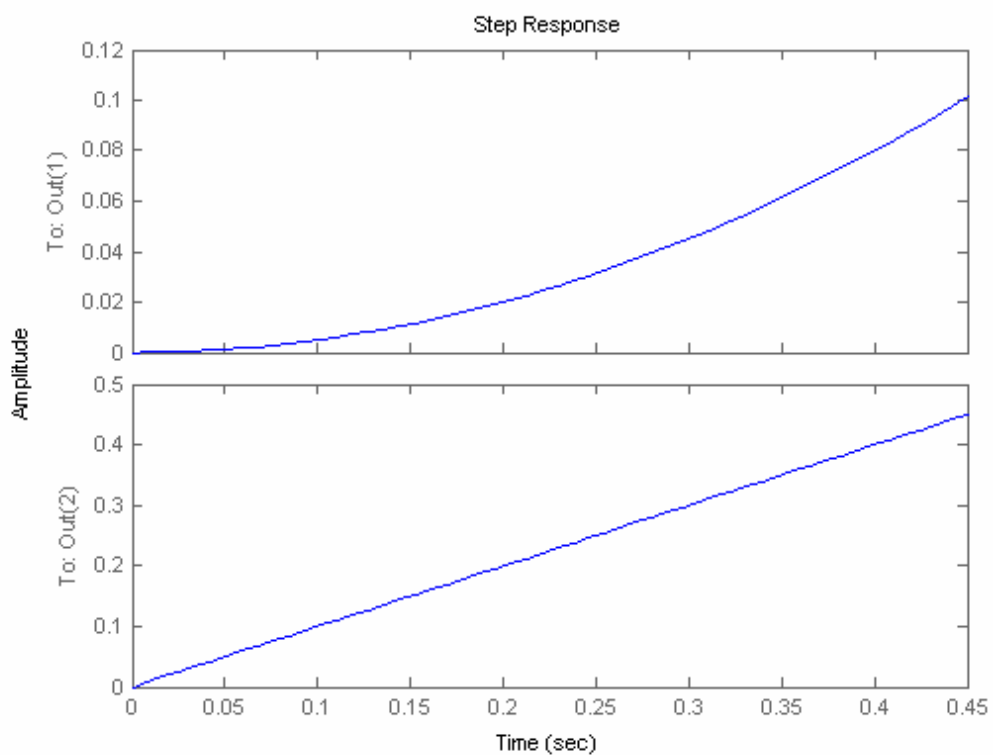


图 3-3 直线一级倒立摆单位阶跃响应仿真

可以看出，在单位阶跃响应作用下，小车位置和摆杆角度都是发散的。

3.1.3 直线一级倒立摆根轨迹控制实验

闭环系统瞬态响应的基本特性与闭环极点的位置紧密相关，如果系统具有可变的环路增益，则闭环极点的位置取决于所选择的环路增益，从设计的观点来看，对于有些系统，通过简单的增益调节就可以将闭环极点移到需要的位置，如果只调节增益不能满足所需要的性能时，就需要设计校正器，常见的校正器有超前校正、滞后校正以及超前滞后校正等。

3.1.3.1 根轨迹分析

前面我们已经得到了倒立摆系统的开环传递函数，输入为小车的加速度，输出为倒立摆系统摆杆的角度，被控对象的传递函数为：

$$\frac{\Phi(s)}{V(s)} = \frac{ml}{(I + ml^2)s^2 - mgl}$$

给系统施加脉冲扰动，输出量为摆杆的角度时，系统框图如下：

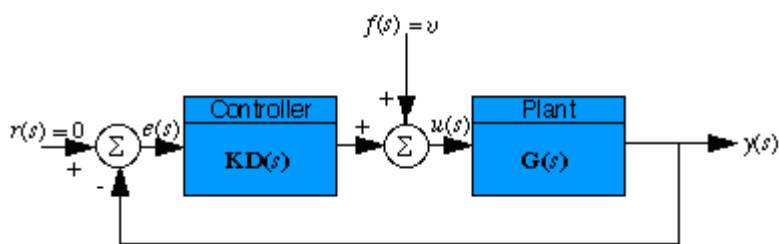


图 3-4 直线一级倒立摆闭环系统图（脉动干扰）

考虑到输入 $r(s) = 0$ ，结构图变换成：

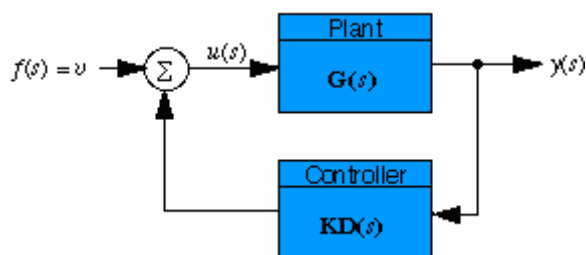


图 3-5 直线一级倒立摆闭环系统简化图（脉动干扰）

该系统的输出为：

$$\begin{aligned}
 Y(s) &= \frac{G(s)}{1 + KD(s)G(s)} U(s) = \frac{\frac{num}{den}}{1 + k \frac{(numlead)(numlag)(num)}{(denlead)(denlag)(den)}} \cdot U(s) \\
 &= \frac{num(denlead)(denlag)}{(denlead)(denlag)(den) + k(numlead)(numlag)(num)} U(s)
 \end{aligned}$$

其中 num ——被控对象传递函数的分子项；

den ——被控对象传递函数的分母项；

$numlead$ 、 $denlead$ ——控制器超前环节传递函数的分子项；

$numlag$ 、 $denlag$ ——控制器滞后环节传递函数的分子项和分母项；

k ——控制器增益

闭环传递函数可以由 Matlab 命令求出。

实际系统的开环传递函数为：

$$\frac{\Phi(s)}{V(s)} = \frac{0.02725}{0.0102125s^2 - 0.26705}$$

在 MATLAB 下新建一个文件，键入如下命令：

```
clear;
num=[0.02725 ];
den=[0.0102125 0 -0.26705];
rlocus(num,den)
z=roots(num);
n=roots(den):
```

得到结果如下：

```
z =
    0
    0
p =
    5.1136
   -5.1136
```

可以看出，系统有两个零点，有两个极点，并且有一个极点为正。

画出系统闭环传递函数的根轨迹如图 3-6，可以看出闭环传递函数的一个极点位于右半平面，并且有一条根轨迹起始于该极点，并沿着实轴向左跑到位于原点的零点处，这意味着无论增益如何变化，这条根轨迹总是位于右半平面，即系

统总是不稳定的。

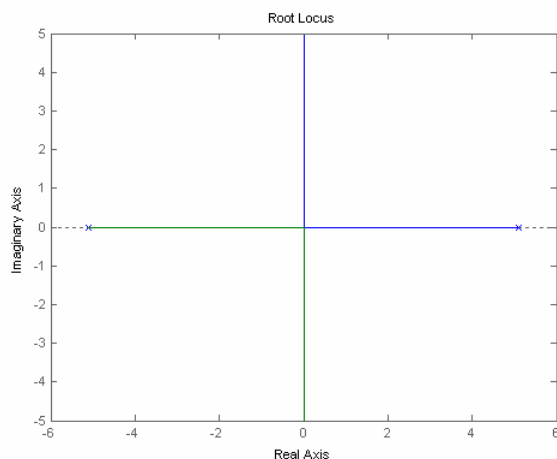


图 3-6 直线一级倒立摆开环根轨迹图

3.1.3.2 根轨迹校正及仿真

3.1.3.2.1 根轨迹校正

直线一级倒立摆的根轨迹校正可以转化为如下的问题：
对于传递函数为：

$$G(s) = \frac{0.02725}{0.0102125s^2 - 0.26705}$$

的系统，设计控制器，使得校正后系统的要求如下：

调整时间 $t_s = 0.5s(2\%)$ ；

最大超调量 $M_p \leq 10\%$

根轨迹设计步骤如下：

1) 确定闭环期望极点 s_d 的位置，由最大超调量

$$M_p = e^{-(\zeta / \sqrt{1-\zeta^2})\pi} \leq 10\%$$

可以得到： $\zeta = 0.591155$ ，近似取 $\zeta = 0.6$ 。

由 $\zeta = \cos(\theta)$ 可以得到：

$$\theta = 0.938306(\text{弧度})$$

其中 θ 为位于第二象限的极点和 o 点的连线与实轴负方向的夹角。

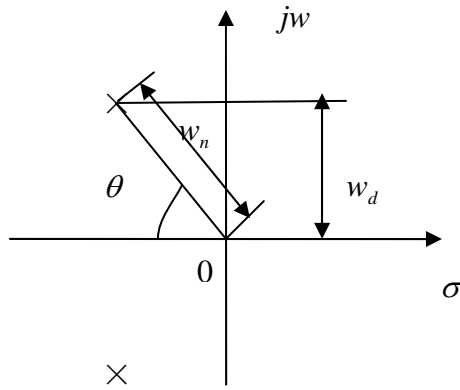


图 3-7 性能指标与根轨迹关系图

又由：

$$t_s = \frac{4}{\zeta \omega_n} \leq 0.5s$$

可以得到： $\omega_n = 13.5328$ ，于是可以得到期望的闭环极点为：

$$13.5328(-\cos\theta \pm j\sin\theta)$$

- 2) 未校正系统的根轨迹在实轴和虚轴上，不通过闭环期望极点，因此需要对系统进行超前校正，设控制器为：

$$K(s) = a \frac{Ts + 1}{aTs + 1} = \frac{s + z_c}{s + p_c} \quad (a \leq 1)$$

- 3) 计算超前校正装置应提供的相角，已知期望的闭环主导极点和系统原来的极点的相角和为：

$$G(s_d) = -\left[\tan^{-1}\left(\frac{13.5328\sin\theta}{13.5328\cos\theta - 5.1136}\right) - \tan^{-1}\left(\frac{13.5328\sin\theta}{13.5328\cos\theta + 5.1136}\right) + 6.28 \right] \\ \approx -4.27676$$

因此校正装置提供的相角为：

$$\phi = -3.14 - (-4.27676) = 1.13517$$

- 4) 设计超前校正装置，已知： $\theta = 0.938306$

对于最大的 a 值的 γ 角度可由下式计算得到：

$$\gamma = \frac{1}{2}(\pi - \theta - \phi)$$

所以有： $\gamma = \frac{1}{2}(\pi - \theta - \phi) = 0.534059$

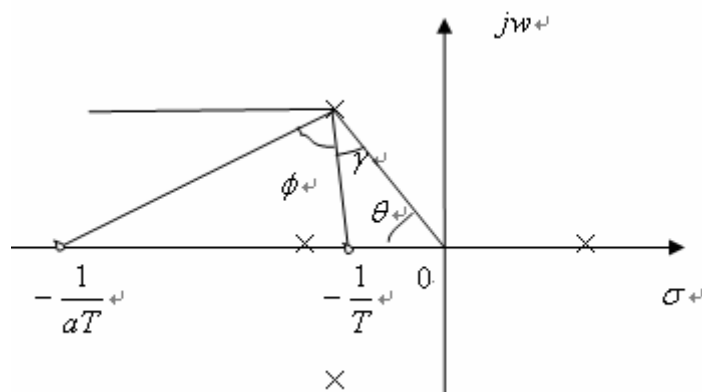


图 3-8 直线一级倒立摆根轨迹计算图

按最佳确定法作图规则，在上图中画出相应的直线，求出超前校正装置的零点和极点，分别为：

$$z_c = -6.92214; z_p = -26.4568$$

校正后系统的开环传递函数为：

$$Q = G(s)K(s) = \frac{K(s + 6.92214)}{(s + 26.4568)} \frac{0.02725}{0.0102125s^2 - 0.26705}$$

5) 由幅值条件 $|G(s_d)H(s_d)| = 1$ ，并设反馈为单位反馈，所以有 $k = 141.137$ ；

6) 于是我们得到了系统的控制器：

$$G(s) = \frac{141.137(s + 6.92214)}{s + 26.4568}$$

7) 上述过程手动计算比较复杂，可以采用编程程序自动计算得到：

在 Mathematica 中编写如下所示的程序，计算以上步骤：

计算文件请参见“\GLIP2001\建模文件\MathematicaFile”中的“L1DIPRLocusCompute.nb”文件。

PRO 3-2 直线一级倒立摆根轨迹计算 Mathematics 程序

```

OverStep = 0.1; (*设置超调量*)
AdjustTime = 0.5; (*设置调整时间*)
pospole = 5.1136; (*系统原来极点*)
negpole = -5.1136;
Mp =  $e^{-\left(\xi/\sqrt{1-\xi^2}\right)\pi}$ ; (*超调量计算公式*)
Print[" $\xi$ "] (*求解系统阻尼*)
Solve[{(Mp - OverStep) == 0}, { $\xi$ }]
damp =  $\xi$  /. %
Print[" $\theta$ "]
 $\theta = \text{ArcCos}[\text{damp}]$  (*期望闭环极点的幅角*)

Ts =  $\frac{4}{\text{damp} * \omega_n}$ ; (*系统调整时间计算公式*)

Print[" $\omega$ "] (*求解系统无阻尼自然振动频率*)
Solve[{(Ts - AdjustTime) == 0}, { $\omega_n$ }]
 $\omega = \omega_n$  /. %

polesre =  $-\omega * \text{Cos}[\theta]$ ;
polesim =  $\omega * \text{Sin}[\theta]$ ;
z = polesre + I * polesim; (*期望闭环极点以复数表示, 为方便下面计算*)

Print["Gsd"]
Gsd =  $-(\text{Arg}[z - \text{pospole}] + \text{Arg}[z - \text{negpole}])$  (*相角计算*)

Print[" $\phi$ "]
 $\phi = -\pi - \text{Gsd}$  (*校正装置应该提供的相角*)
Print[" $\gamma$ "]
 $\gamma = (\pi - \phi - \theta) / 2$  (*按最佳确定法计算幅角*)

Print["zerocoord"]
zerocoord = polesre - polesim *  $\text{Tan}[\gamma - (\pi / 2 - \theta)]$  (*计算校正装置的零点*)
Print["polecoord"]
polecoord = polesre - polesim *  $\text{Tan}[\gamma - (\pi / 2 - \theta) + \phi]$ 
(*计算校正装置的极点*)

fk = Abs $\left[k * \frac{z - \text{zerocoord}}{z - \text{polecoord}} * \frac{0.02725}{0.0102125 * z^2 - 0.26705}\right]$ ;
Print["K"]
Solve[{fk - 1 == 0}, {k}]
(*根据幅值条件计算反馈系数K*)

```

3.1.3.2 MATLAB 计算仿真

在 MATLAB 中编写如下的 m 文件,对系统进行仿真。

PRO 3-3 直线一级倒立摆根轨迹计算 MATLAB 程序

```
clear;
num=[0.02725];           % Transfer function of the Linear 1 stage
den=[0.0102125 0 -0.26705]; % inverted penulum
numlead=-6.92214;        % Controller Zeros
denlead=-26.4568;        % Controller Poles

[Z,P,K]=tf2zp(num,den);
Za=[Z;numlead];          % Add Zeros to the LIP system
Pa=[P;denlead];          % Add Poles to the LIP system

[num2,den2]=zp2tf(Za,Pa,K);
sys=tf(num2,den2);
rlocus(sys);
KK=141.137                % compensated system
sys2=zpk(Za,Pa,KK*K);
sysc=sys2/(1+sys2);
t=0:0.005:5;
step(sysc,t)              % Response of the Close-loop system
```

运行即可以得到以上的计算结果,校正后系统的跟轨迹如下图所示:

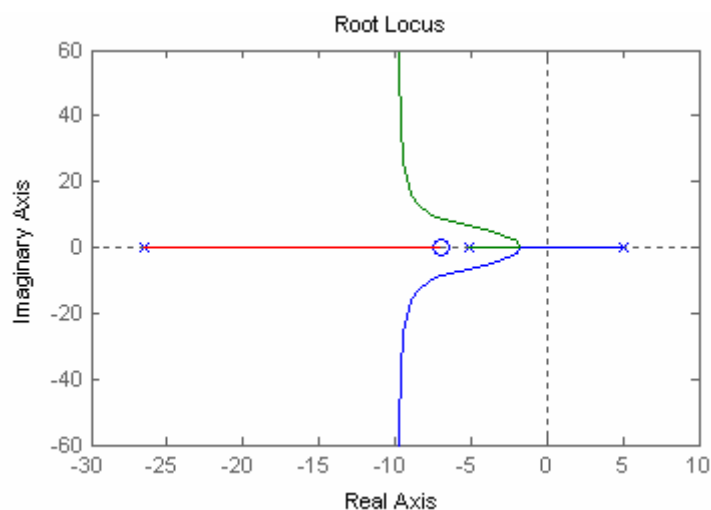


图 3-9 校正后的根轨迹图

从图中可以看出,系统的三条根轨迹都有位于左半平面的部分,选取适当的 K 就可以稳定系统。

系统的阶跃响应如下所示:

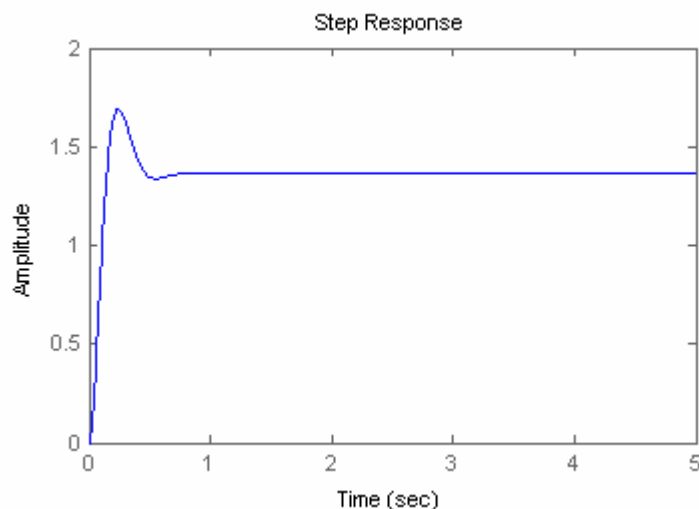


图 3-10 校正后的阶跃响应

可以看出，系统有较好的稳定性，但系统存在一定的稳态误差，并且超调过大，为使系统瞬态响应满足要求，可以采用以下的方法：

第一种方法：增加阻尼 ζ ，重复上面的设计方法（此时，以上编写的程序修改即为方便），重新设计，直到系统的响应满足要求。请读者自行设计，这里不再描述其过程。

第二种方法：在保持 ϕ 角不便的情况下，将校正装置的零点向左侧偏移，以减少闭环零点和极点的影响，实验者可以按此方法重新进行设计。

对于有经验的使用者，可以采用直接对系统增加零点和极点的方法：

为使位于右半平面的根轨迹进入左边平面，因此增加一个左半平面的零点，假设为-7，得到系统新的根轨迹如下：

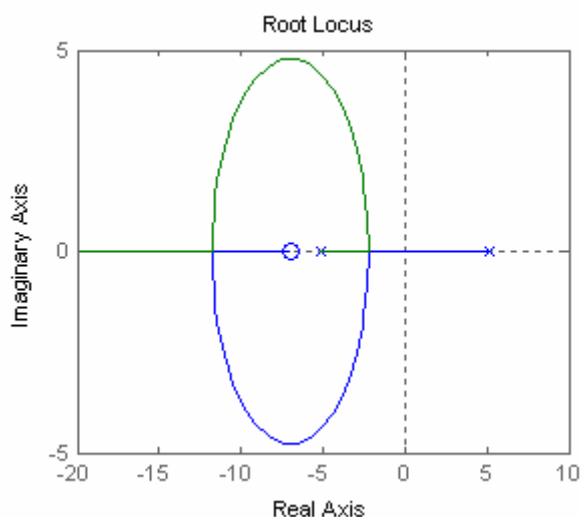


图 3-11 手动校正后的根轨迹图（增加零点）

可以看出，增加零点后，系统的两条根轨迹都进入了左半平面，系统可以稳定，但在实际系统中，传递函数分子的阶次必须比分母的阶次低，因此我们增加一个远离零点的极点，增加极点后（例如增加极点-30），系统的根轨迹如下：

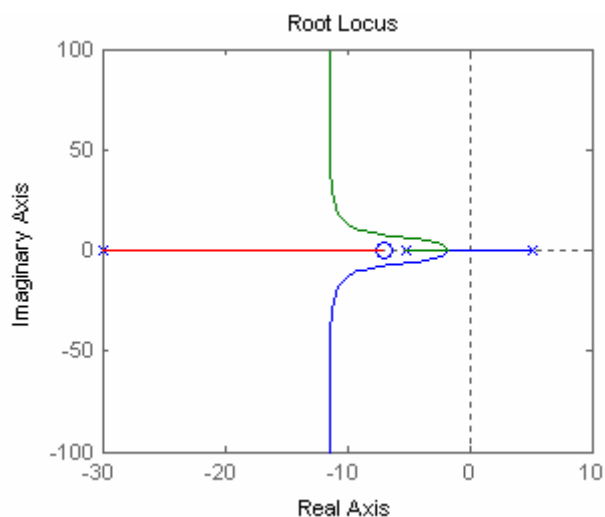


图 3-12 手动校正后的根轨迹图（增加零点）

可以看出，系统的根轨迹都有位于左半平面的部分，选取适当的增益(计算结果：236.1682)，可以得到一个稳定的闭环控制系统。

闭环系统的脉冲响应如下图所示：

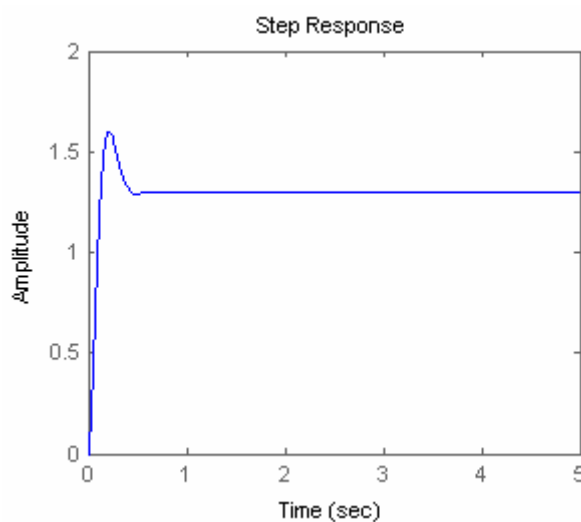


图 3-13 手动校正后的阶跃响应曲线

可以看出，系统在 1s 的时间内可以稳定，响应比较迅速，超调比较小。

上述根轨迹设计 MATLAB 程序如下：

（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ Root Locus Experiments” 中的 “Root Locus Control M Files”）

PRO 3-4 直线一级倒立摆根轨迹校正 MATLAB 程序 2

```

%%%%%%%%%%
%   Googol Linear 1 stage Inverted Pendulum Transfer Function %
%%%%%%%%%%
clear;
num=[0.02725];           % Transfer function of the Linear 1 stage
den=[0.0102125 0 -0.26705]; % inverted penulum
numlead=-7;              % Controller Zeros
denlead=-30;             % Controller Poles

subplot(2,2,1)
rlocus(num,den)

[Z,P,K]=tf2zp(num,den);
Za=[Z;numlead];         % Add Zeros to the LIP system

sysaddzero=zpk(Za,P,K);
subplot(2,2,2)
rlocus(sysaddzero)

Pa=[P;denlead];         % Add Poles to the LIP system

[num2,den2]=zp2tf(Za,Pa,K);
sys=tf(num2,den2);

subplot(2,2,3)
rlocus(sys);
[KK,poles]=rlocfind(sys) % Find the K and Poles of the
%KK=141.137              % compensated system
sys2=zpk(Za,Pa,KK*K);    % Open loop transfer function
% of the compensated system

subplot(2,2,3)
axis([-60 10 -10 10])
rlocus(sys2)             % Draw the root locus of the
% compensated system

sysc=sys2/(1+sys2);      % Close-loop transfer function
t=0:0.005:5;
subplot(2,2,4);
step(sysc,t)             % Response of the Close-loop system

```

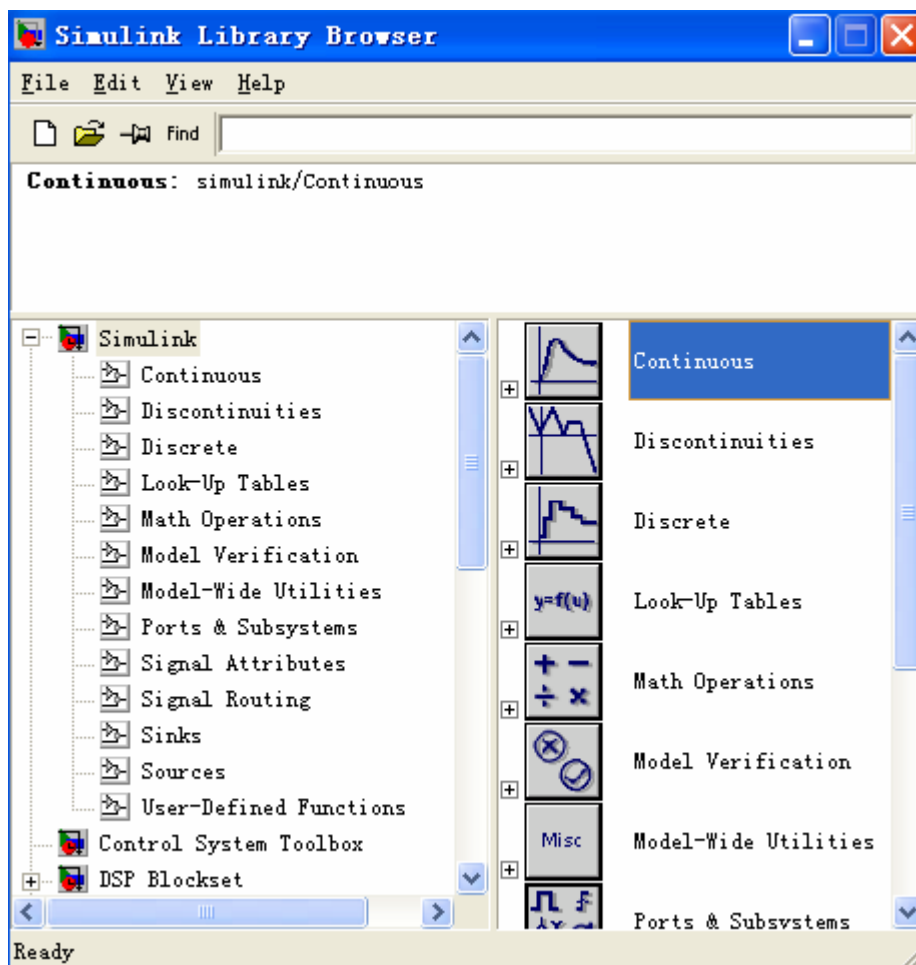
改变控制器的极点和零点，可以得到不同的控制效果，请实验者自行设计并分析结果的差异。

3.1.3.2 MATLAB Simulink 仿真

MATLAB 提供了一个强大的图形化仿真工具 Simulink，下面在 Simulink 中建立直线一级倒立摆的模型，（对于初次使用 MATLAB Simulink 实验者，请在进行此实验之前熟悉 Simulink 相关的知识和使用方法），这里详细介绍一下模型的建立方法：

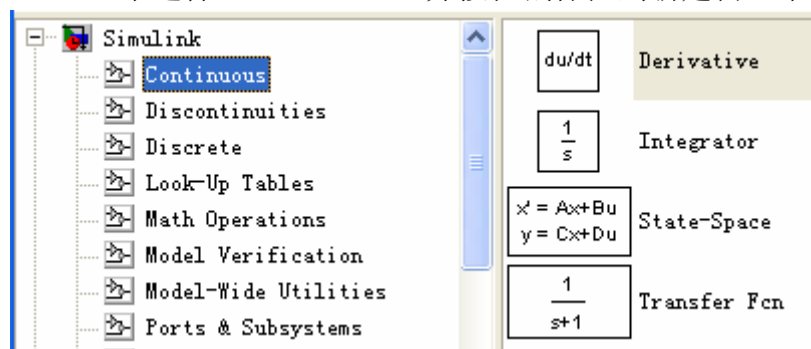
1) 打开 MATLAB Simulink 窗口。

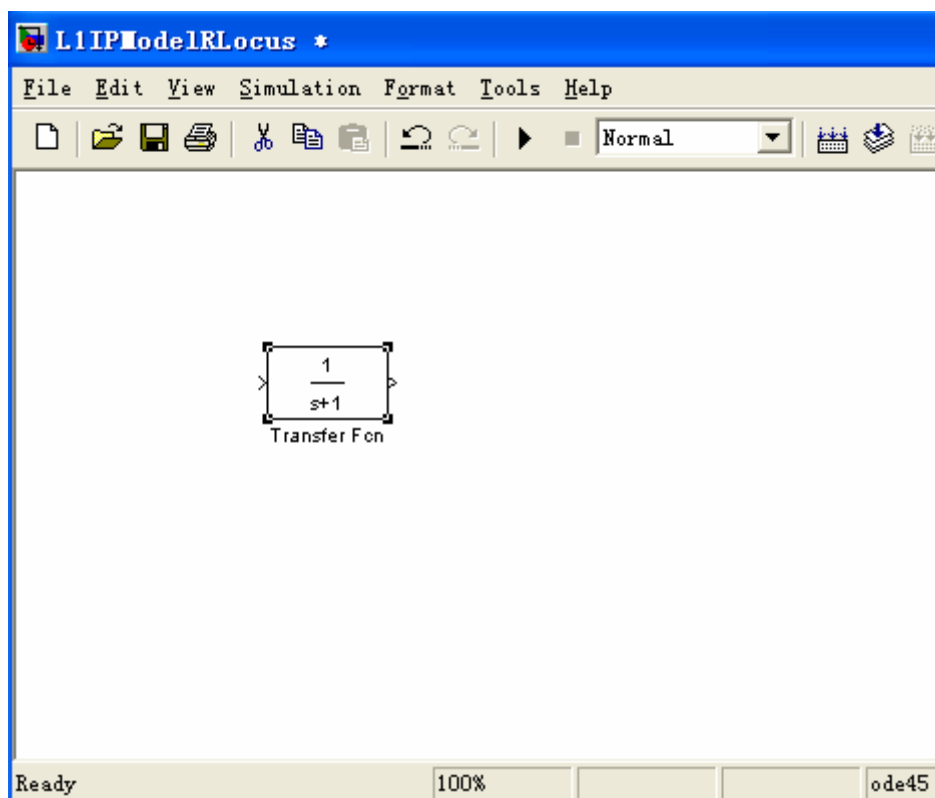
点击 MATLAB 窗口中 “” 图标进入 Simulink 环境，Simulink 窗口如下：



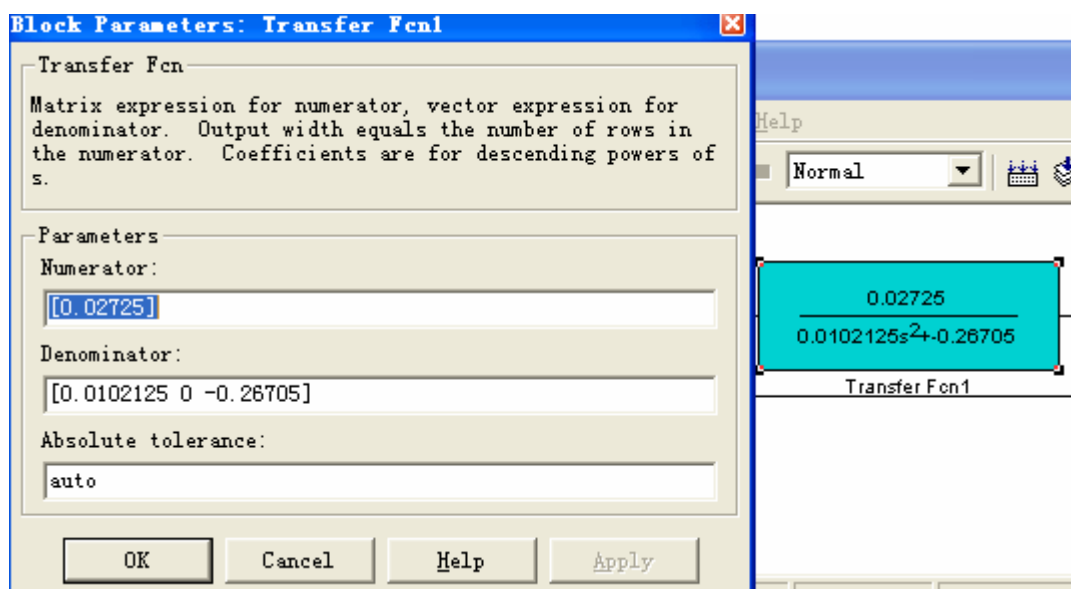
2) 点击 “” 新建一个模型，并命名，如 “L1IPModelRLocus”。

从 “Continuous” 中选择 “Transfer Fcn” 并按住鼠标拖到新建窗口中，

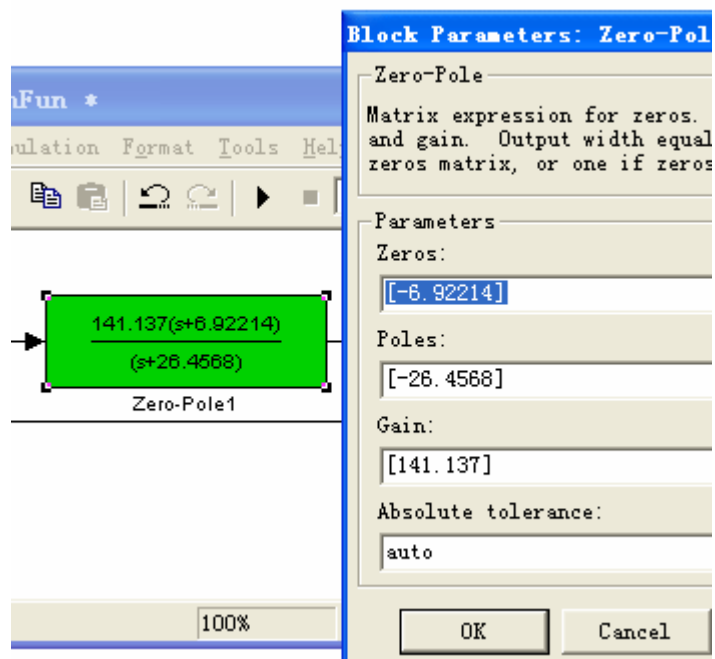




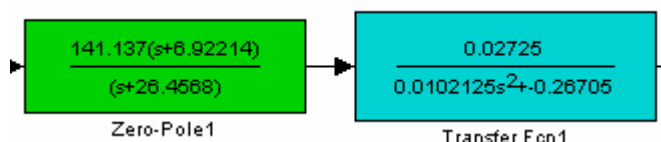
- 3) 将上面的传递函数模块改名为“L1IP Transfer Fcn”，点击鼠标右键，选取“BackGround Color”为“Cyan”，并双击模块，打开参数设置窗口如下，



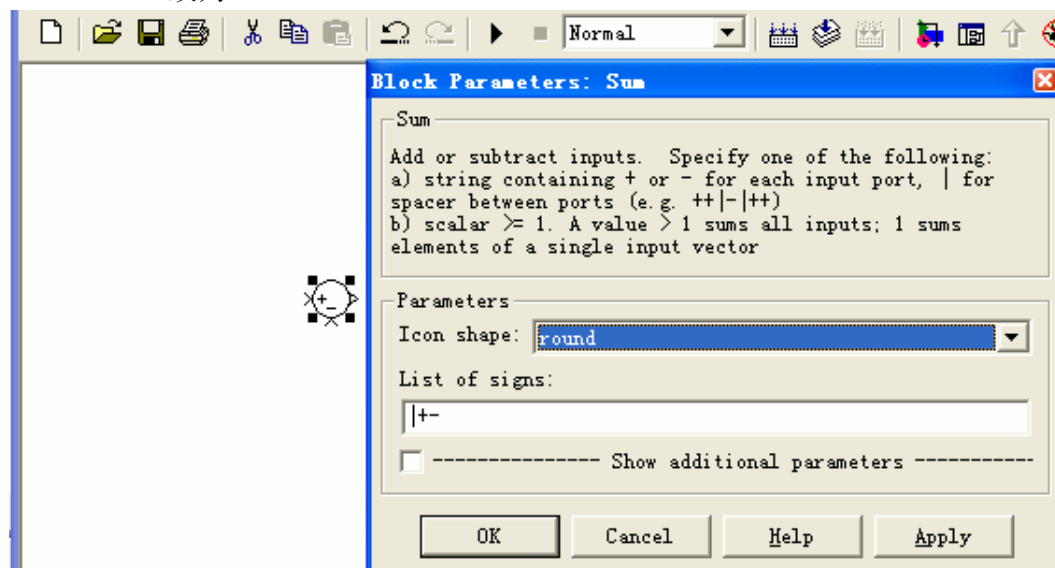
- 4) 同样从 Simulink 模型库“Continuous”中拖一个“Zero-Pole”模块到窗口中作为控制器，双击模块，设定零点、极点和上面程序计算得到的增益 KK 值。



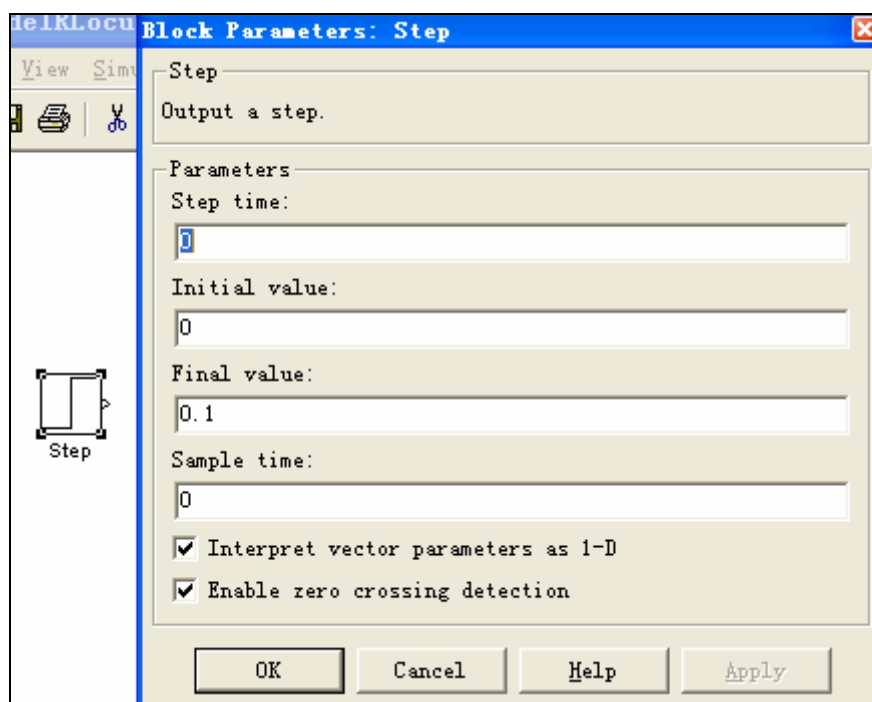
- 5) 连接“Controller”模块和“L1IP L1IP Transfer Fcn”模块。



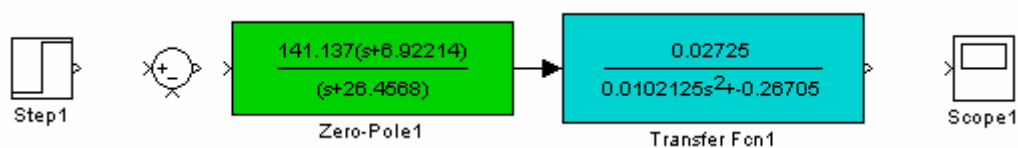
- 6) 从 Simulink 模型库“Math Operations”中拖一个“Sum”模块到“L1IPModelRLocus”窗口中，并双击模块改为如下图所示（把其中的“++”改为“+-”）：



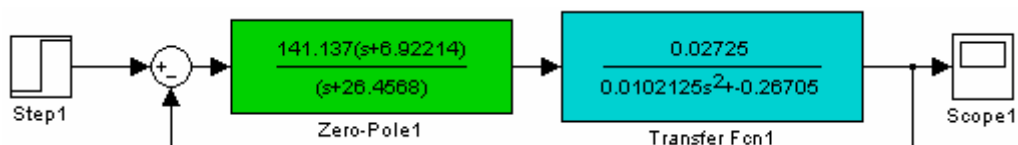
- 7) 从 Simulink 模型库“Sources”中拖一个“Step”信号模块到“L1IPModelRLocus”窗口中，并双击模块设置阶跃信号参数：



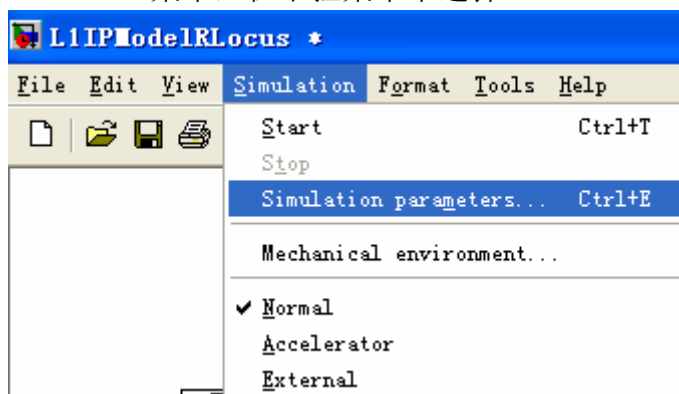
- 8) 从 Simulink 模型库 “Sinks” 中拖一个 “Scope” 信号模块到 “L1IPModelRLocus” 窗口中，如下图所示：



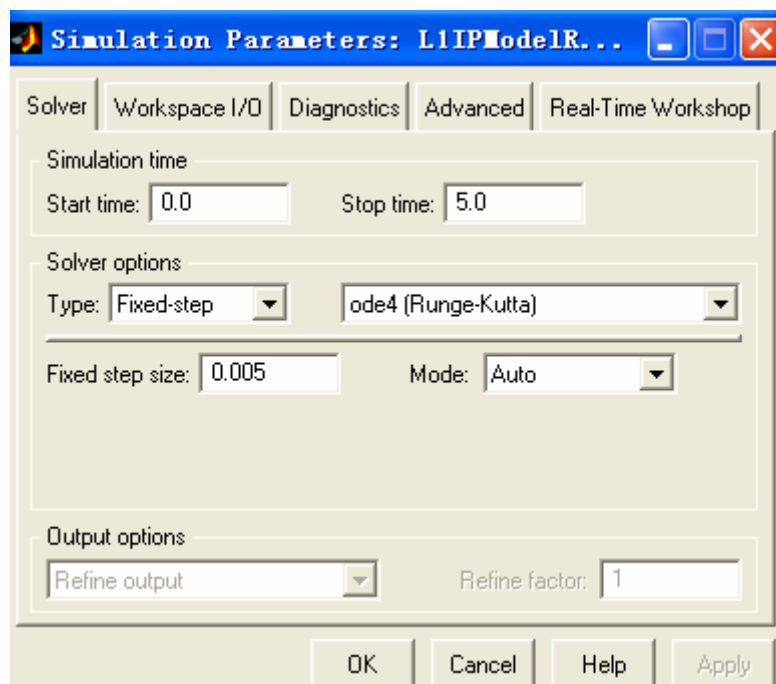
- 9) 连接各个模块如下图所示，构成一个闭环控制系统。



- 10) 点击 “Simulation” 菜单，在下拉菜单中选择 “Simulation Parameters”，



打开如下窗口：



在上面窗口中设置“Simulation time”以及“Solver options”等选项。设置仿真步长为 0.005 秒。

11) 点击▶运行仿真，双击“Scope”模块观察仿真结果：

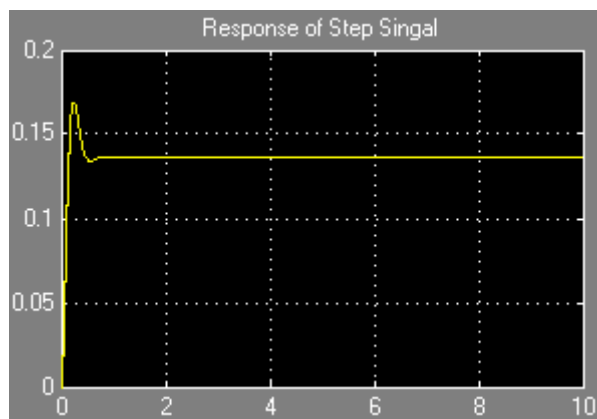

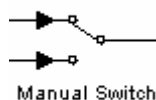


图 3-14 直线一级倒立摆的根轨迹校正仿真结果（一阶控制器）

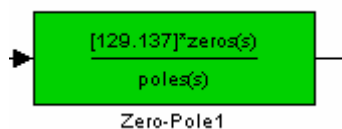
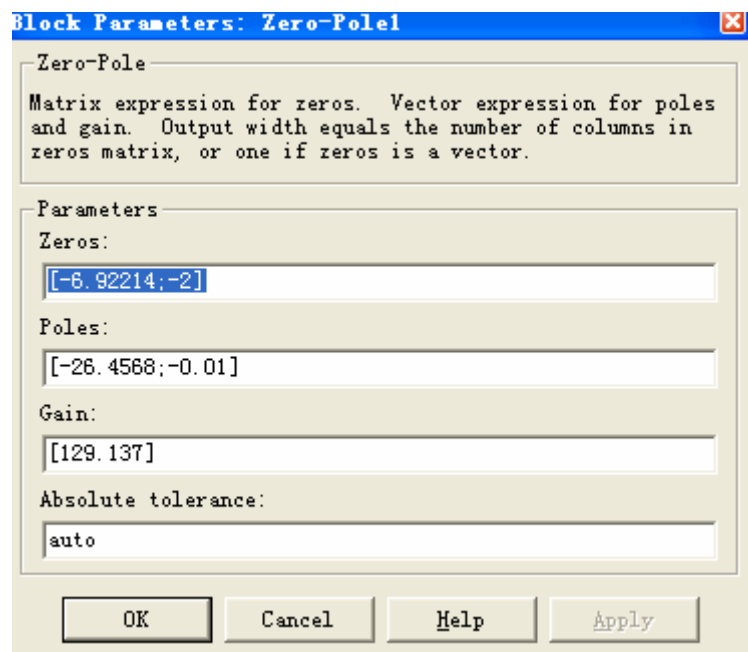
如果曲线超出界面范围，可以点击“”观察全图。

12) 可以看出，系统能较好的跟踪阶跃信号，但是存在一定的稳态误差，修改控制器的零点和极点，可以得到不同的控制效果，请在多次改变参数后，选取仿真结果最好的参数。例如修改控制为二阶的超前滞后控制器，给控制器再增加一个极点和零点，具体的设计方法请参见相关教材。

13) 在“Simulink\Signal Routing”中拉一个“Manual Switch”模块到窗口中，



14) 复制一个控制器模块到窗口中并修改参数：



15) 最后整理根轨迹仿真模块如下图所示:

(参考模型: 进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\Root Locus Experiments” 中的 “Root Locus Control Simulink”)

Googol Linear 1-Stage Inverted Pendulum Root Locus Control Simulink

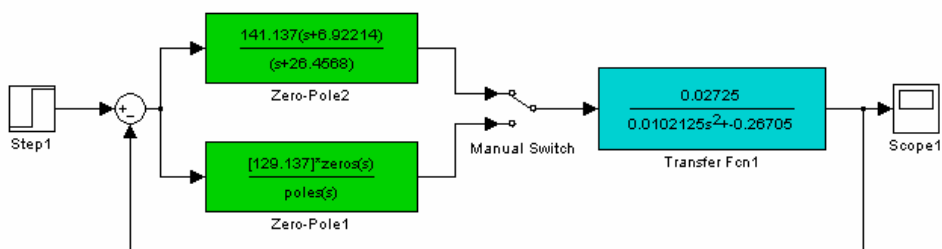


图 3-15 直线一级倒立摆的根轨迹仿真模型

双击 “Manual Switch” 打到下边, 点击 “▶” 运行得到仿真结果如下:

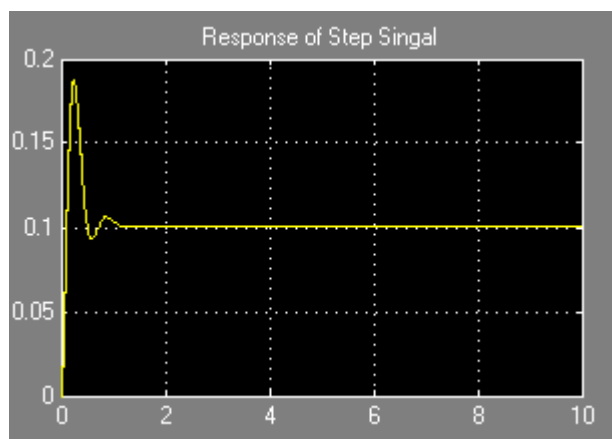


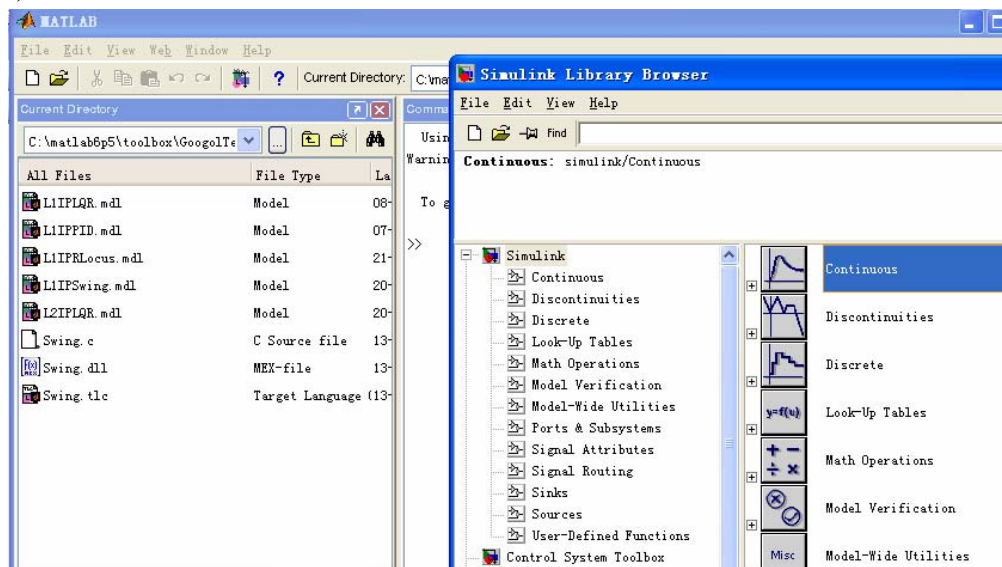
图 3-16 直线一级倒立摆的根轨迹校正仿真结果（二阶控制器）

可以看出，系统稳态误差相对较少，但是超调增大，请实验者分析原因并改进控制器。

3.1.3.3 根轨迹校正实时控制实验

在进行实验前请先安装固高科技 MATLAB 实时控制软件，其方法请参见软件的使用说明书。

1) 打开 MATLAB 以及 Simulink，如下图所示：



2) 在 Simulink 中打开“Googol Educational Products”并选择“Inverted Pendulum”，然后选择“Linear Inverted Pendulum”，在右边的界面中选择“Linear 1 Stage Inverted Pendulum Root Locus Control”

3) 打开直线一级倒立摆根轨迹控制界面：

（进入 MATLAB Simulink 实时控制工具箱“Googol Education Products”打开“Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\Root Locus Experiments”中的“Root Locus Control Demo”）

Googol Linear 1-Stage Inverted Pendulum Root Locus Control Demo

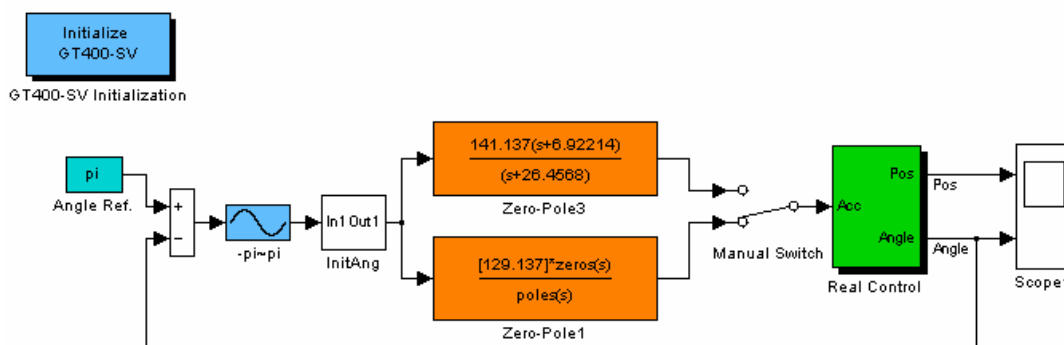





图 3-17 直线一级倒立摆的根轨迹实时控制模块

- 4) 点击 “” 编译程序，编译成功后在 MATLAB 命令窗口中有提示信息（如果没有修改控制界面结构，在编译一次后，不需再进行此步骤）：

```

>> exportLib
-export:Enable
-export:GetBoards
    Creating library ..\L1IPRLocus.lib and object ..\L1IPRLocus.exp
### Created Real-Time Windows Target module L1IPRLocus.rwd.
### Successful completion of Real-Time Workshop build procedure for model: L1IPRLocus
>>
  
```

如果不成功，请按照固高科技 *MATLAB* 实时控制软件使用说明书查找原因。

- 5) 点击 “” 连接程序；
- 6) 点击 “” 运行程序；
- 7) 程序运行后，可以听到电机上伺服的声音，



注 意

- 在电机上伺服后，请所有实验人员在倒立摆的运动范围外，出现异常时关闭电控箱电源，并在试验前检查倒立摆系统的状况。

- 8) 手动提起倒立摆到竖直位置，程序进入自动控制。
- 9) 双击 “Scope” 观察实验数据：

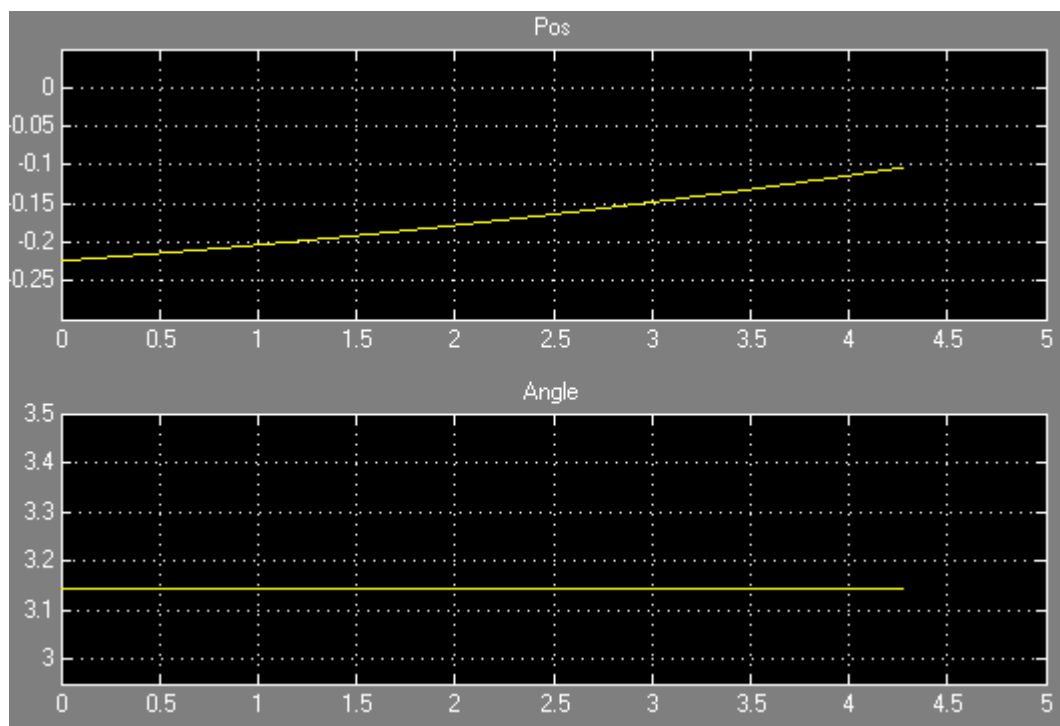


图 3-18 直线一级倒立摆的根轨迹控制结果（平衡）

10) 双击“Manual Switch”切换控制器，观察实验数据并记录，可以看到在给定倒立摆干扰的情况下系统的响应：

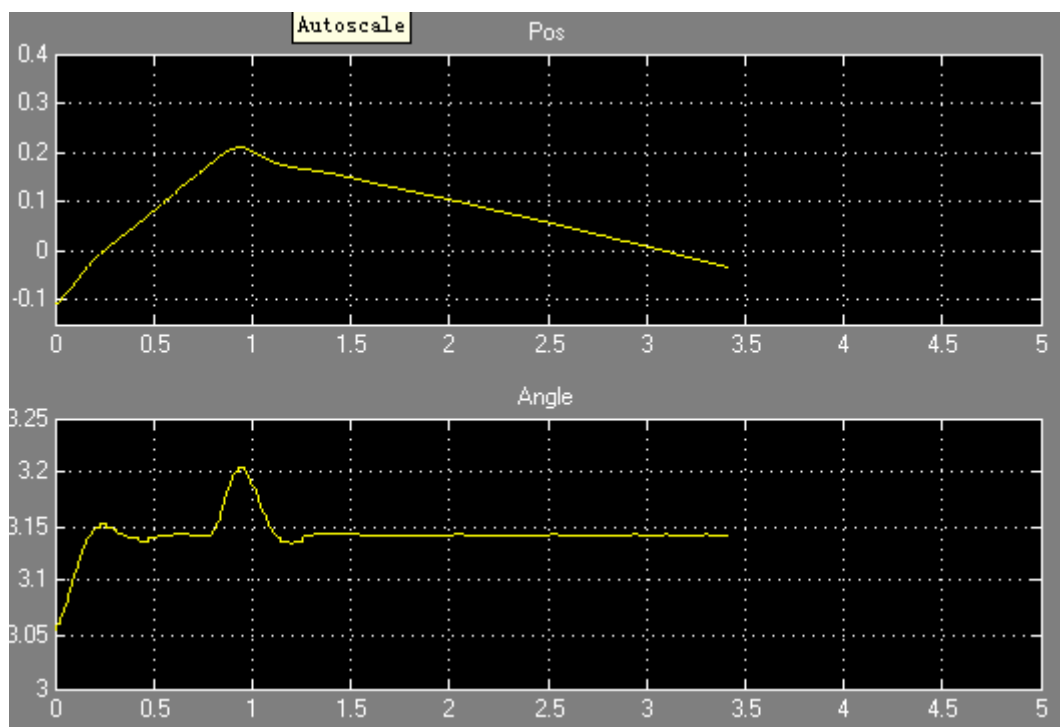
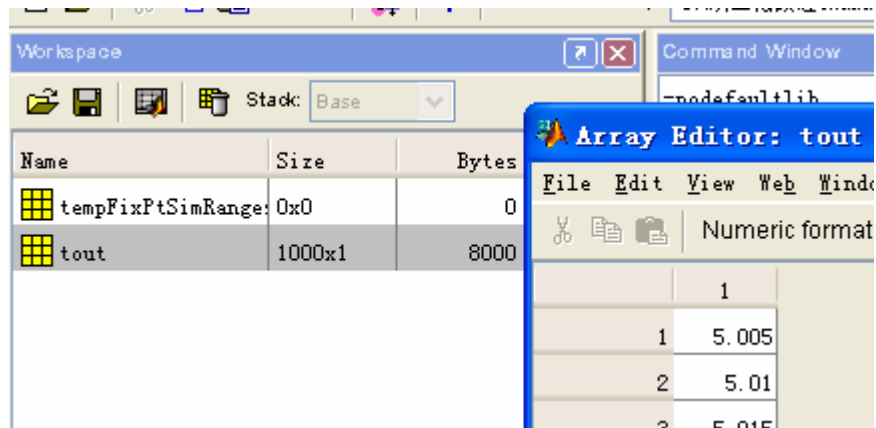


图 3-19 直线一级倒立摆的根轨迹控制结果（抵抗干扰）

11) 用户可以把实验数据保存为文件或时输出到 MATLAB 工作空间“Workspace”：其方法请参考 MATLAB 相关书籍。



- 12) 分析实验数据和仿真结果的差别。
- 13) 读者也可以采用根轨迹的其它方法对系统进行校正。

3.1.3.4 实验结果及实验报告

请将计算步骤，仿真和实验结果记录并完成实验报告。
实验报告样本如下：

表 3-1 直线一级倒立摆根轨迹实时控制实验报告

直线一级倒立摆根轨迹实时控制实验报告			
学生姓名:		学 号:	
系 别:		日 期:	
课程名称:			
实验目的			
实验设备			
实验原理			
仿真步骤 及结果			
实验结果 记录			
实验结果 分析			
教师评语			

3.1.4 直线一级倒立摆频率响应控制实验

系统对正弦输入信号的响应，称为频率响应。在频率响应方法中，我们在一定范围内改变输入信号的频率，研究其产生的响应。

频率响应可以采用以下三种比较方便的方法进行分析，一种为伯德图或对数坐标图，伯德图采用两幅分离的图来表示，一幅表示幅值和频率的关系，一幅表示相角和频率的关系；一种是极坐标图，极坐标图表示的是当 ω 从0变化到无穷大时，向量 $|G(j\omega)| / \angle G(j\omega)$ 的轨迹，极坐标图也常称为奈奎斯特图，奈奎斯特稳定判据使我们有可能根据系统的开环频率响应特性信息，研究线性闭环系统的绝对稳定性和相对稳定性。

3.1.4.1 频率响应分析

前面我们已经得到了直线一级倒立摆的物理模型，实际系统的开环传递函数为：

$$\frac{\Phi(s)}{V(s)} = \frac{0.02725}{0.0102125s^2 - 0.26705}$$

其中输入为小车的加速度 $V(s)$ ，输出为摆杆的角度 $\Phi(s)$ 。

在 MATLAB 下绘制系统的 Bode 图和奈奎斯特图。

绘制 Bode 图的命令为：

`Bode(sys)`

绘制奈奎斯特图的命令为：

`Nyquist(sys)`

在 MATLAB 中键入以下命令：

```
clear;
num=[0.02725];
den=[0.0102125 0 -0.26705];

z=roots(num);
p=roots(den);

subplot(2,1,1)
bode(num,den)
subplot(2,1,2)
nyquist(num,den)
```

得到如下图所示的结果：

```

z =
Empty matrix: 0-by-1

p =
    5.1136
   -5.1136

```

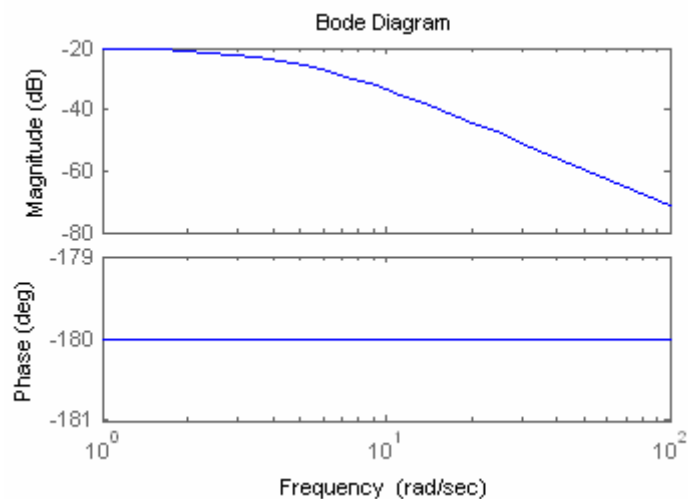


图 3-20 直线一级倒立摆的 Bode 图

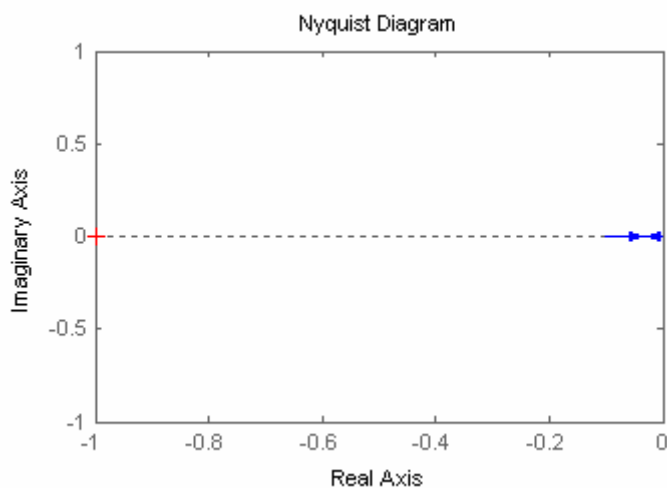


图 3-21 直线一级倒立摆的奈奎斯特图

可以得到, 系统没有零点, 但存在两个极点, 其中一个极点位于右半 s 平面, 根据奈奎斯特稳定判据, 闭环系统稳定的充分必要条件是: 当 ω 从 $-\infty$ 到 $+\infty$ 变化时, 开环传递函数 $G(j\omega)$ 沿逆时针方向包围 -1 点 p 圈, 其中 p 为开环传递函数在右半 S 平面内的极点数。对于直线一级倒立摆, 由图 3-21 我们可以看出, 开环传递函数在 S 右半平面有一个极点, 因此 $G(j\omega)$ 需要沿逆时针方向包围 -1 点

一圈。可以看出，系统的奈奎斯特图并没有逆时针绕-1点一圈，因此系统不稳定，需要设计控制器来镇定系统。

3.1.4.2 频率响应设计及仿真

直线一级倒立摆的频率响应设计可以表示为如下问题：

考虑一个单位负反馈系统，其开环传递函数为：

$$G(s) = \frac{\Phi(s)}{V(s)} = \frac{0.02725}{0.0102125s^2 - 0.26705}$$

设计控制器 $G_c(s)$ ，使得系统的静态位置误差常数为 10，相位裕量为 50° ，增益裕量等于或大于 10 分贝。

根据要求，控制器设计如下：

1) 选择控制器，上面我们已经得到了系统的 Bode 图，可以看出，给系统增加一个超前校正就可以满足设计要求，设超前校正装置为：

$$G_c(s) = K_c \alpha \frac{Ts + 1}{\alpha Ts + 1} = K_c \frac{s + \frac{1}{T}}{s + \frac{1}{\alpha T}}$$

已校正系统具有开环传递函数 $G_c(s)G(s)$

设

$$G_1(s) = KG(s) = \frac{0.02725 \times K}{0.0102125s^2 - 0.26705}$$

式中 $K = K_c \alpha$ 。

2) 根据稳态误差要求计算增益 K ，

$$K_p = \lim_{s \rightarrow 0} G_c(s)G(s) = \lim_{s \rightarrow 0} K_c \frac{(s + \frac{1}{T})}{(s + \frac{1}{\alpha T})} \times \frac{0.02725}{0.0102125s^2 - 0.26705} = 10$$

可以得到：

$$K_c \alpha = 98 = K$$

于是有：

$$G_1(s) = \frac{0.02725 \times 98}{0.0102125s^2 - 0.26705}$$

3) 在 MATLAB 中画出 $G_1(s)$ 的 Bode 图：

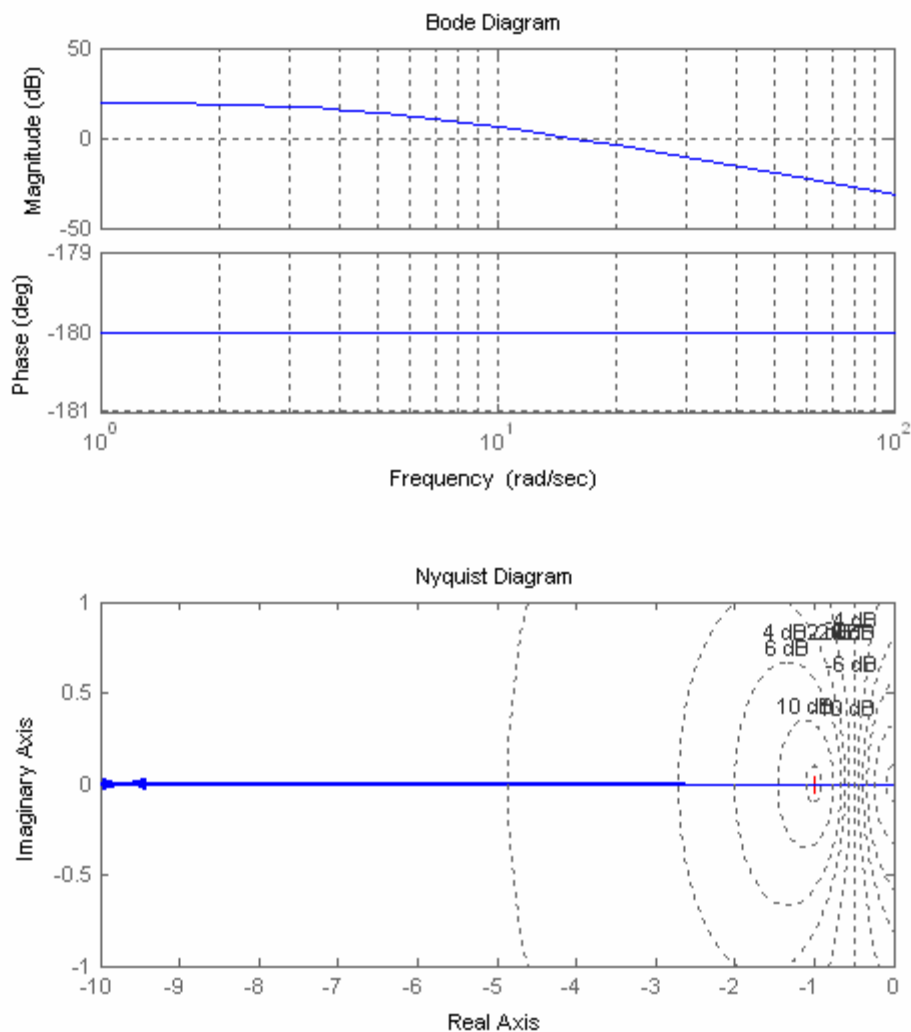


图 3-22 添加增益后的直线一级倒立摆的 Bode 图和 Nyquist 图

4) 可以看出，系统的相位裕量为 0° ，根据设计要求，系统的相位裕量为 50° ，因此需要增加的相位裕量为 50° ，增加超前校正装置会改变 Bode 图的幅值曲线，这时增益交界频率会向右移动，必须对增益交界频率增加所造成的 $G_1(j\omega)$ 的相位滞后增量进行补偿，因此，假设需要的最大相位超前量 ϕ_m 近似等于 55° 。

因为

$$\sin \phi_m = \frac{1-\alpha}{1+\alpha}$$

计算可以得到: $\alpha = 0.0994$

5) 确定了衰减系统, 就可以确定超前校正装置的转角频率 $\omega = 1/T$ 和 $\omega = 1/(\alpha T)$, 可以看出, 最大相位超前角 ϕ_m 发生在两个转角频率的几何中心上, 即 $\omega = 1/(\sqrt{\alpha}T)$, 在 $\omega = 1/(\sqrt{\alpha}T)$ 点上, 由于包含 $(Ts+1)/(\alpha Ts+1)$ 项, 所以幅值的变化为:

$$\left| \frac{1+j\omega T}{1+j\omega\alpha T} \right|_{\omega=1/(\sqrt{\alpha}T)} = \left| \frac{1+j\frac{1}{\sqrt{\alpha}}}{1+j\sqrt{\alpha}} \right| = \frac{1}{\sqrt{\alpha}}$$

$$\text{又 } \frac{1}{\sqrt{\alpha}} = \frac{1}{\sqrt{0.0994}} = 10.0261 \text{ 分贝}$$

并且 $|G_1(j\omega)| = -10.0261$ 分贝对应于 $\omega = 28.5 \text{ rad/s}$, 我们选择此频率作为新的增益交界频率 ω_c , 这一频率相应于 $\omega = 1/(\sqrt{\alpha}T)$, 即 $\omega_c = 1/(\sqrt{\alpha}T)$, 于是

$$\frac{1}{T} = \sqrt{\alpha}\omega_c = 8.9854$$

$$\frac{1}{\alpha T} = \frac{\omega_c}{\sqrt{\alpha}} = 90.3965$$

6) 于是校正装置确定为:

$$G_c(s) = K_c \alpha \frac{Ts+1}{\alpha Ts+1} = K_c \frac{s+8.9854}{s+90.3965}$$

$$K_c = \frac{K}{\alpha} = 985.9155$$

7) 增加校正后系统的根轨迹和奈奎斯特图如下:

(进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\Frequency Response Experiments” 中的 “Frequency Response Control M Files”)

PRO 3-5 直线一级倒立摆的频率响应校正 MATLAB 程序

```
%%%%%%%%%%%%%%  
%   Googol Linear 1 stage Inverted Pendulum Frequency Response %  
%%%%%%%%%%%%%%  
  
clear;  
num=98*[0.02725];  
den=[0.0102125 0 -0.26705];  
subplot(2,1,1)  
bode(num,den)  
subplot(2,1,2)  
nyquist(num,den)  
  
z=roots(num);  
p=roots(den);  
  
za=[z;-8.9854];  
pa=[p;-90.3965];  
k=985.9155;  
sys=zpk(za,pa,k);  
  
figure  
subplot(2,1,1)  
bode(sys)  
subplot(2,1,2)  
nyquist(sys)  
  
figure  
sysc=sys/(1+sys);  
t=0:0.005:5;  
impulse(sysc,t)
```

从 Bode 图中可以看出,系统具有要求的相角裕度和幅值裕度,从奈魁斯特图中可以看出,曲线绕-1 点逆时针一圈,因此校正后的系统稳定。

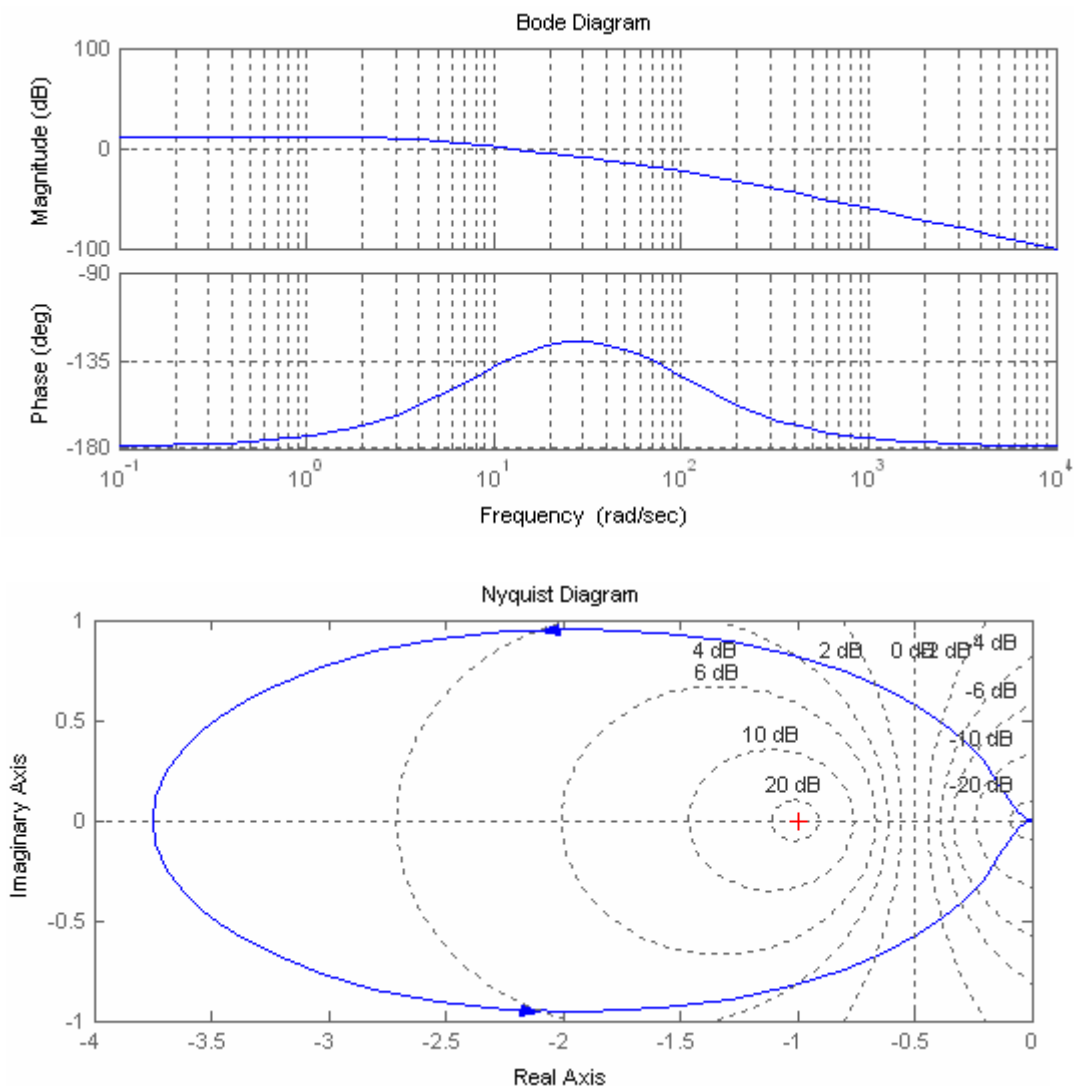


图 3-23 添加控制器后的直线一级倒立摆 Bode 图和 Nyquist 图（一阶控制器）

得到系统的单位阶跃响应如下：

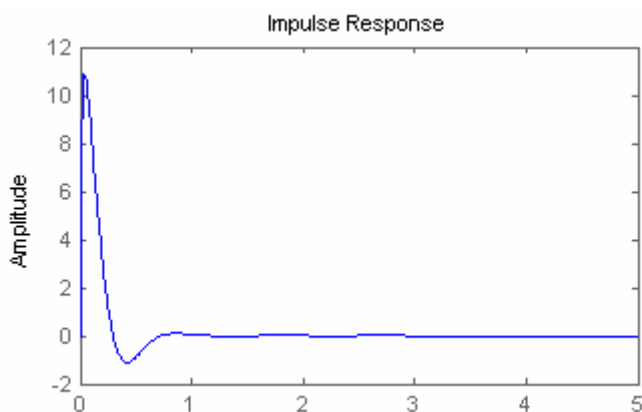


图 3-24 利用频率响应方法校正后系统的单位阶跃响应（一阶控制器）

可以看出，系统在遇到干扰后，在 1 秒内可以达到新的平衡，但是超调量比较大。

8) 打开“L1dofFreq.mdl”，在 MATLAB Simulink 下对系统进行仿真（本例和以下的例子都不再详细说明每步的操作方法，详细的步骤请参见前一章内容）。

（进入 MATLAB Simulink 实时控制工具箱“Googol Education Products”打开“Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\Frequency Response Experiments”中的“Frequency Response Control Simulink”）

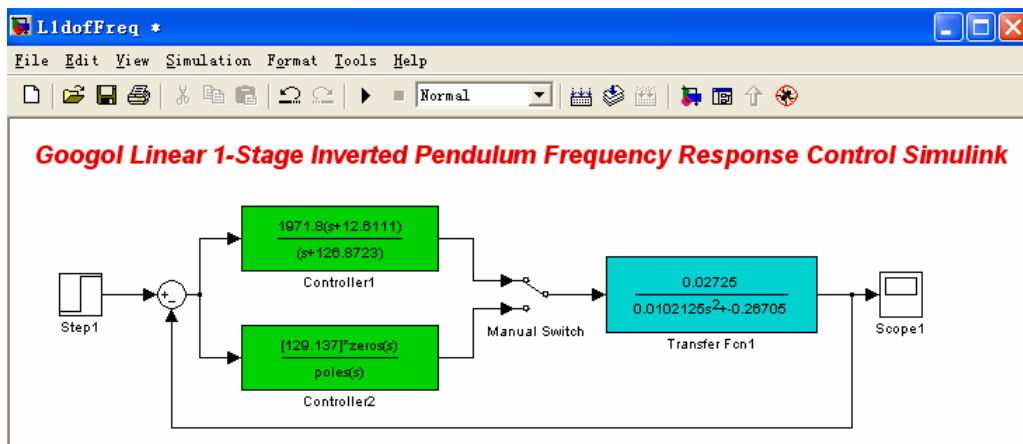
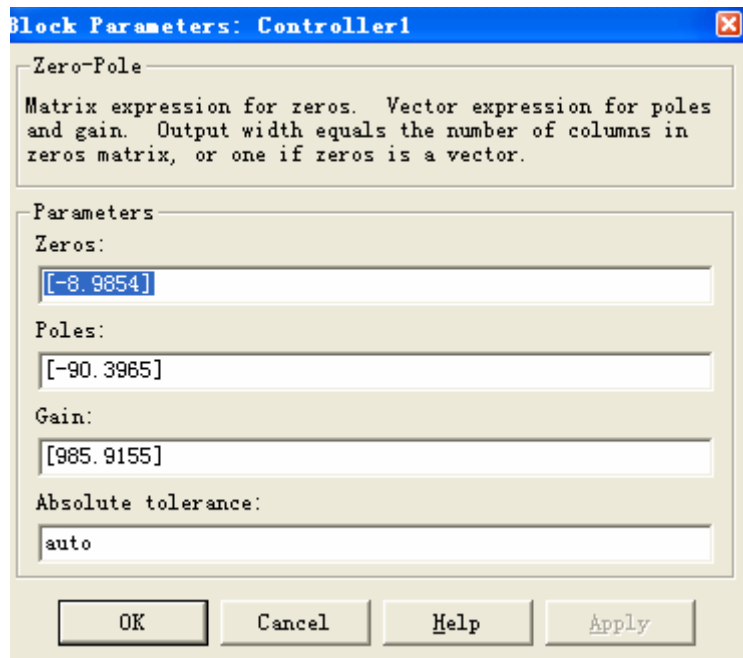
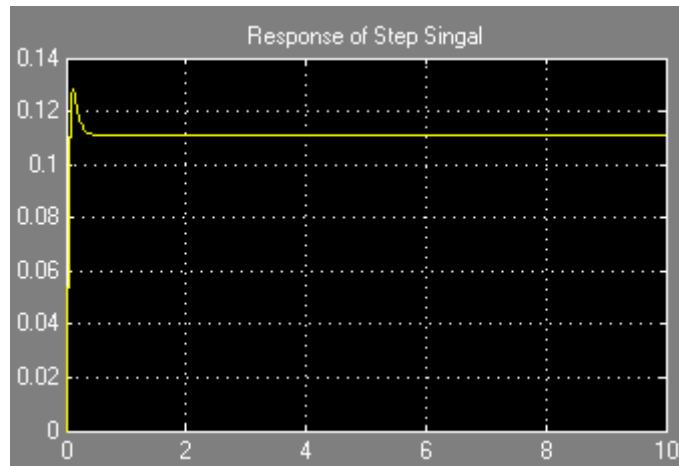


图 3-25 直线一级倒立摆的频率响应校正仿真程序

双击“Controller1”设置校正器参数：



点击“▶”得到以下仿真结果：



9) 可以看出,系统存在一定的稳态误差,为使系统获得快速响应特性,又可以得到良好的静态精度,我们采用滞后-超前校正(通过应用滞后-超前校正,低频增益增大,稳态精度提高,又可以增加系统的带宽和稳定性裕量),设滞后-超前控制器为:

$$G_c(s) = K_c \frac{(s + \frac{1}{T_1})(s + \frac{1}{T_2})}{(s + \frac{\beta}{T_1})(s + \frac{1}{\beta T_2})}$$

10) 请读者参考相关教材设计滞后-超前控制器。设控制器为:

$$G_c(s) = K_c \frac{(s + \frac{1}{T_1})(s + \frac{1}{T_2})}{(s + \frac{\beta}{T_1})(s + \frac{1}{\beta T_2})} = 980 \times \frac{s + 8.9854}{s + 90.3965} \times \frac{s + 2}{s + 0.1988}$$

可以得到静态误差系数:

$$\begin{aligned} K_p &= \lim_{s \rightarrow 0} G_c(s)G(s) \\ &= \lim_{s \rightarrow 0} 980 \times \frac{s + 8.9854}{s + 90.3965} \times \frac{s + 2}{s + 0.1988} \times \frac{0.02725}{0.0102125s^2 - 0.26705} \\ &= 100.6 \end{aligned}$$

比超前校正提高了很多,因为-2零点和-0.1988极点比较接近,所以对相角裕度影响等不是很大,滞后-超前校正后的系统 Bode 图和奈魁斯特图如下所示:

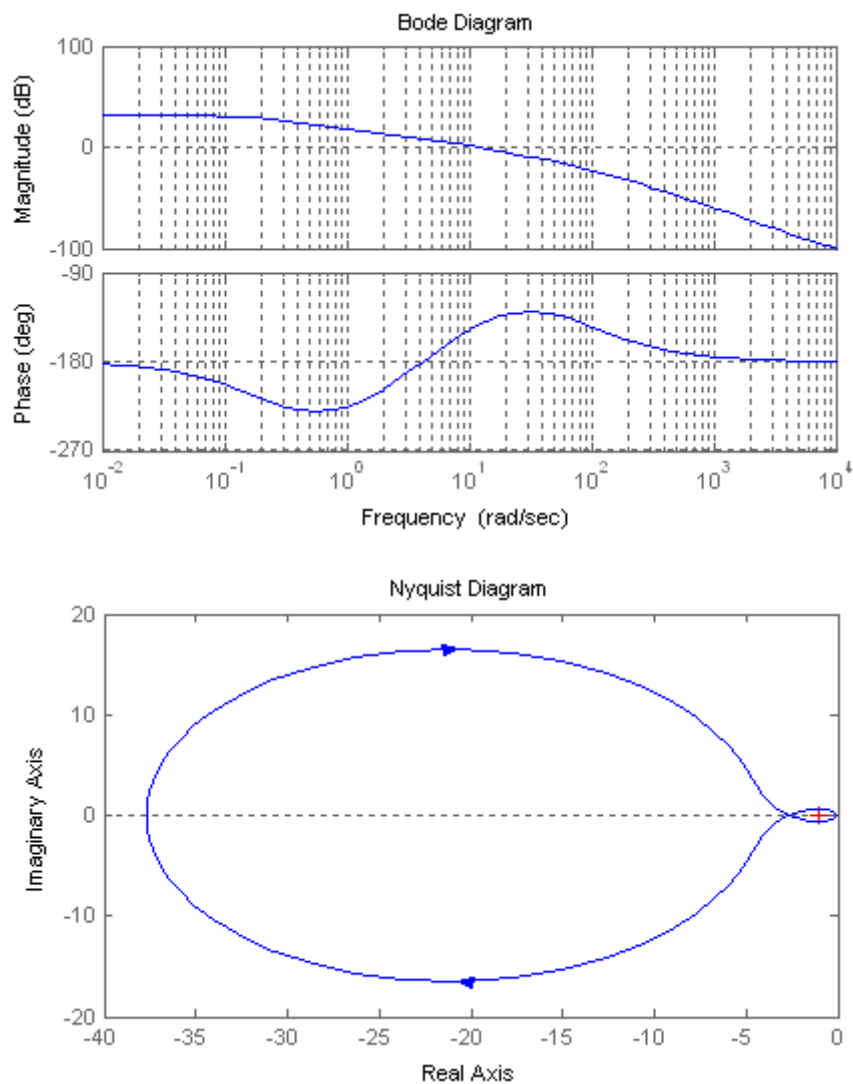
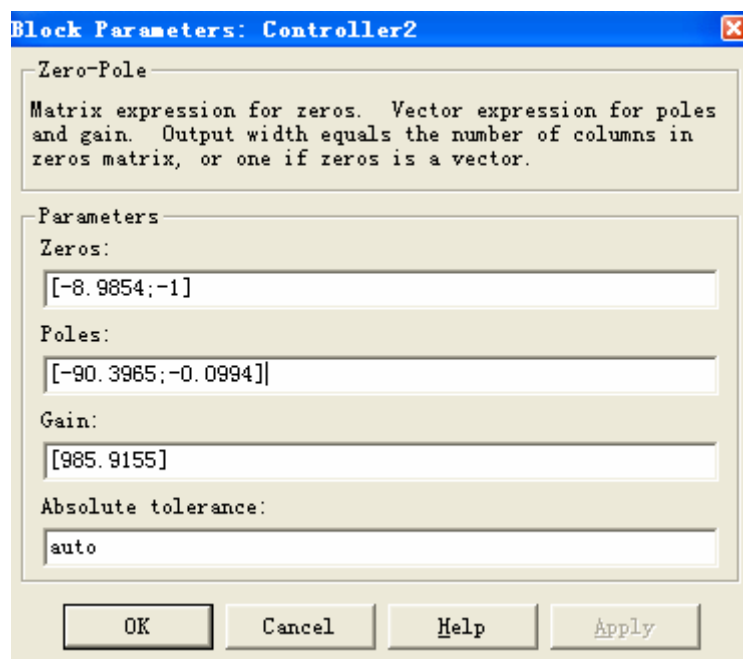


图 3-26 利用频率响应方法校正后的 Bode 图和 Nyquist 图（二阶控制器）

设“Controller2”为：



运行仿真结果，可以得到：

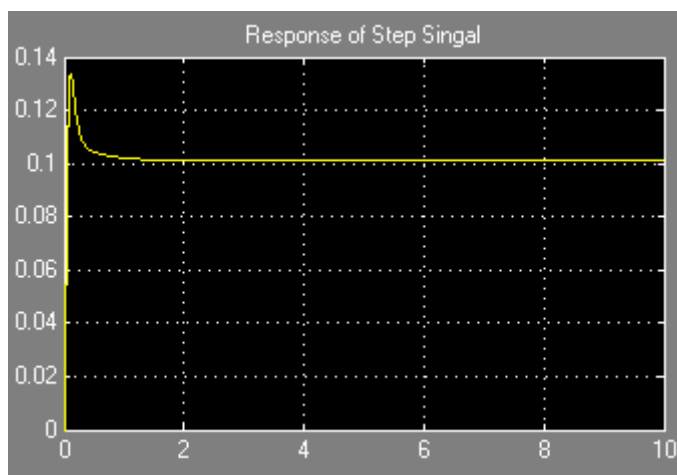


图 3-27 频率响应校正后阶跃响应仿真结果（二阶控制器）

可以很明显的看出，系统的稳态误差较少。

3.1.4.3 直线一级倒立摆频率响应校正法实验

1) 进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\Frequency Response Experiments” 中的 “Frequency Response Control Demo”（详细的超作步骤请参考前一张根轨迹校正控制实验）：

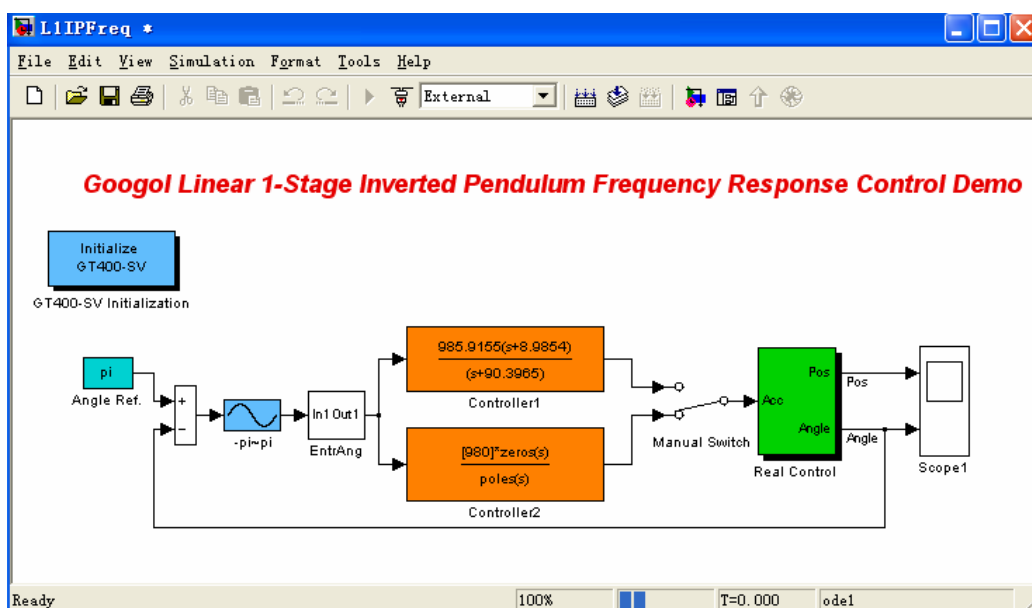
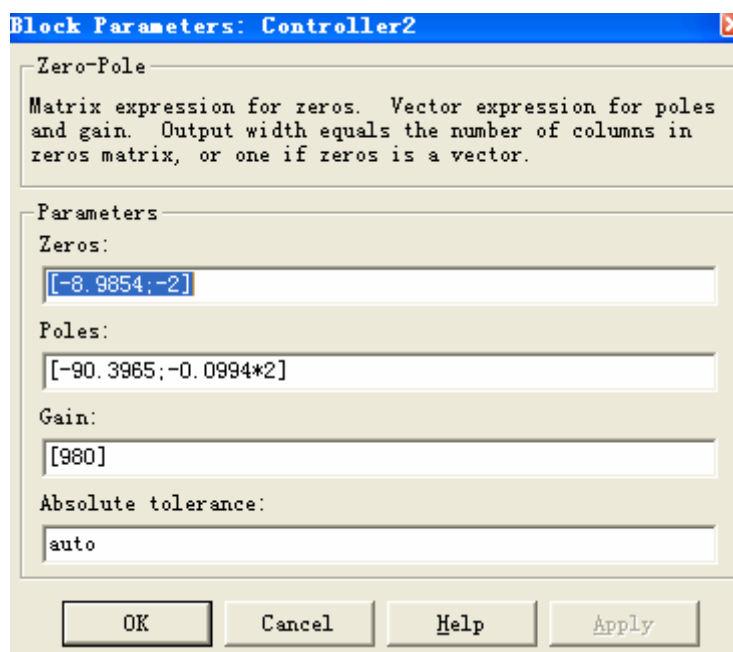





图 3-28 直线一级倒立摆的频率响应校正实时控制程序

- 2) 点击“Manual Switch”选择控制器，选择控制器“Contrller1”或是“Controller2”
- 3) 双击 Controller2，设置上面计算和仿真得到的参数。



- 4) 点击“”编译程序，再编译成功后点击“”连接程序。
- 5) 打开电控箱电源。
- 6) 点击“”运行程序。
- 7) 在听到电机上伺服的声音后，手动缓慢的提起摆杆到竖直向上的位置，在程序进入自动控制后松开手，因为频率响应法只控摆杆的角度，并不控制

小车的位置，所以当小车运动到一端时需要用工具挡一下，以免碰到限位开关，停止控制。

8) 双击“Scope”观察运行结果：

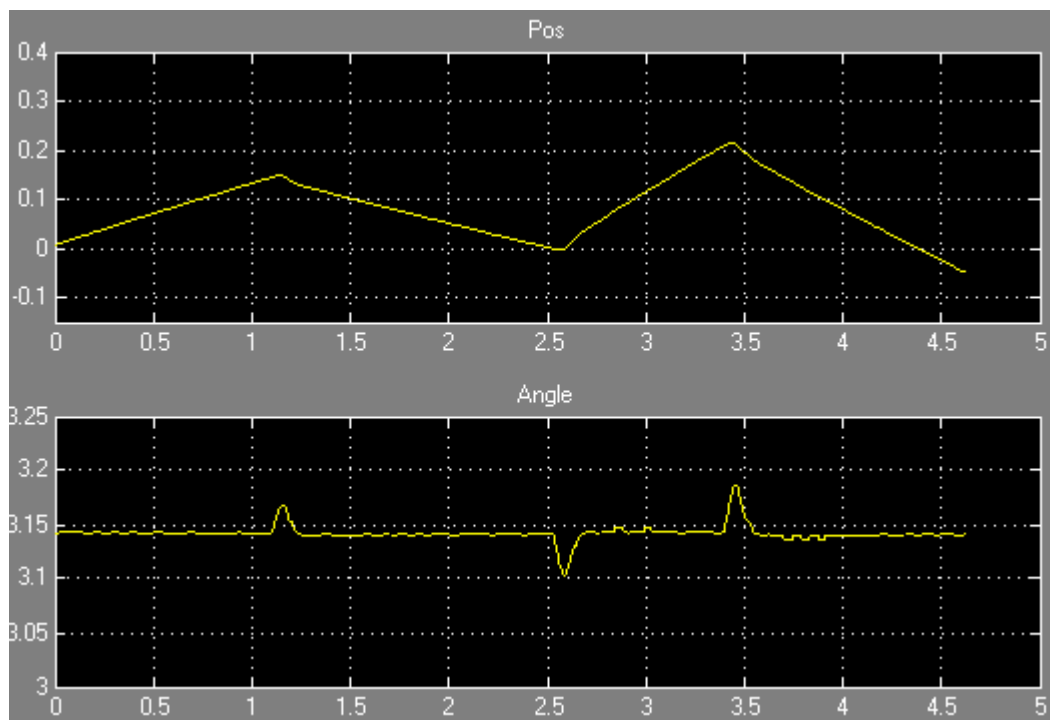


图 3-29 频率响应校正实时控制结果（一阶控制器）

请读者仔细分析系统实际的稳定时间（上图所示约为 0.25 秒）和设计指标的关系。

9) 根据不同的指标计算得到不同的控制器参数，在修改参数后观察控制效果。

10) 点击“Manual Switch”切换控制器后，系统的控制效果如下：

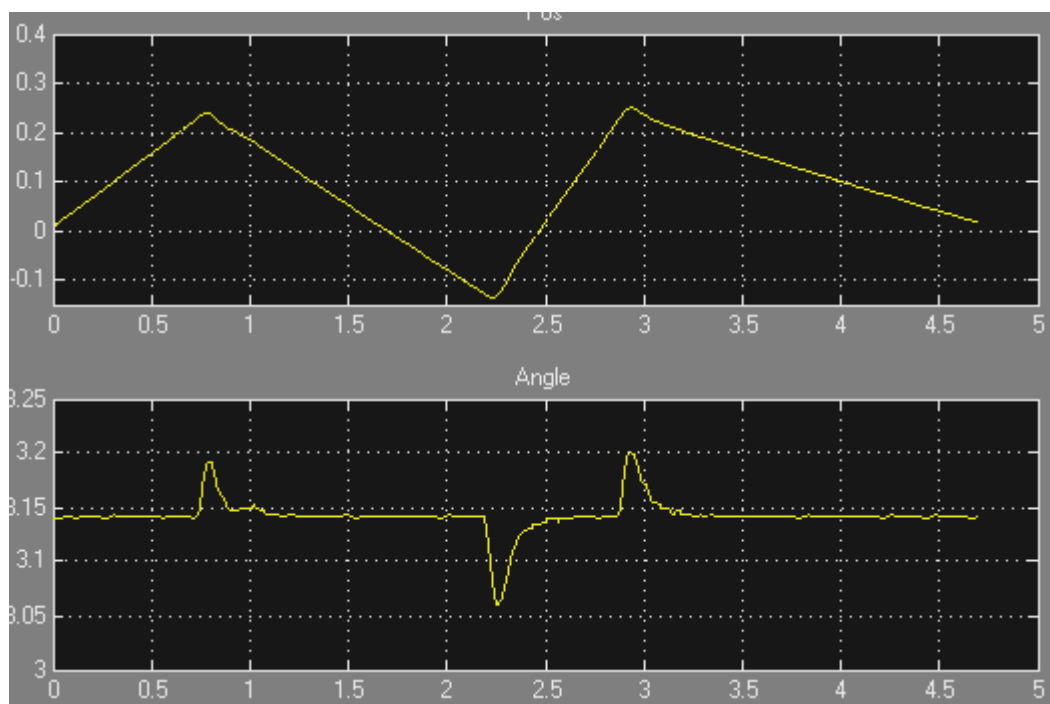


图 3-30 频率响应校正实时控制结果（二阶控制器）

请读者仔细分析这个控制器和前面控制器的控制效果差别（稳定时间约为 0.5 秒）。

3.1.4.4 实验结果及实验报告

记录实验结果并完成实验报告。

3.1.5 直线一级倒立摆 PID 控制实验

本实验的目的是让实验者理解并掌握 PID 控制的原理和方法，并应用于直线一级倒立摆的控制，PID 控制并不需要对系统进行精确的分析，因此我们采用实验的方法对系统进行控制器参数的设置。

3.1.5.1 PID 控制分析

经典控制理论的研究对象主要是单输入单输出的系统，控制器设计时一般需要有关被控对象的较精确模型。PID 控制器因其结构简单，容易调节，且不需要对系统建立精确的模型，在控制上应用较广。

首先，对于倒立摆系统输出量为摆杆的角度，它的平衡位置为垂直向上的情况。系统控制结构框图如下：

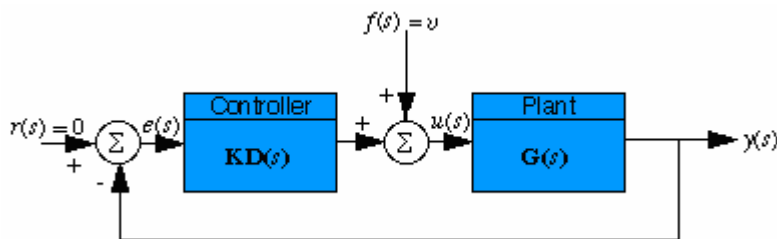


图 3-31 直线一级倒立摆闭环系统图

图中 $KD(s)$ 是控制器传递函数， $G(s)$ 是被控对象传递函数。

考虑到输入 $r(s) = 0$ ，结构图可以很容易的变换成：

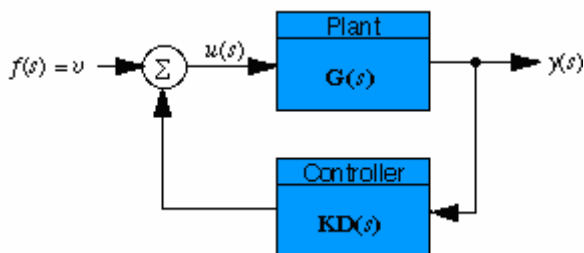


图 3-32 直线一级倒立摆闭环系统简化图

该系统的输出为：

$$\begin{aligned}
 y(s) &= \frac{G(s)}{1 + KD(s)G(s)} F(s) = \frac{\frac{num}{den}}{1 + \frac{(numPID)(num)}{(denPID)(den)}} F(s) \\
 &= \frac{num(denPID)}{(denPID)(den) + (numPID)(num)} F(s)
 \end{aligned}$$

其中 num ——被控对象传递函数的分子项

den ——被控对象传递函数的分母项

$numPID$ ——PID 控制器传递函数的分子项

$denPID$ ——PID 控制器传递函数的分母项

通过分析上式就可以得到系统的各项性能。

由(3-13)可以得到摆杆角度和小车加速度的传递函数：

$$\frac{\Phi(s)}{V(s)} = \frac{ml}{(I + ml^2)s^2 - mgl}$$

PID 控制器的传递函数为：

$$KD(s) = K_D s + K_p + \frac{K_I}{s} = \frac{K_D s^2 + K_p s + K_I}{s} = \frac{numPID}{denPID}$$

需仔细调节 PID 控制器的参数，以得到满意的控制效果。

前面的讨论只考虑了摆杆角度，那么，在控制的过程中，小车位置如何变化呢？

小车位置输出为：

$$X(s) = V(s)s^2$$

通过对控制量 v 双重积分即可以得到小车位置。

3.1.5.2 PID 控制参数设定及仿真

PID 参数设定法则可以参考《现代控制工程》第十章 PID 控制与鲁棒控制，对于 PID 控制参数，我们采用以下的方法进行设定。

由实际系统的物理模型：

$$\frac{\Phi(s)}{V(s)} = \frac{0.02725}{0.0102125s^2 - 0.26705}$$

在 Simulink 中建立如图所示的直线一级倒立摆模型：

（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打

开“Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ PID Experiments”中的“PID Control Simulink”))。

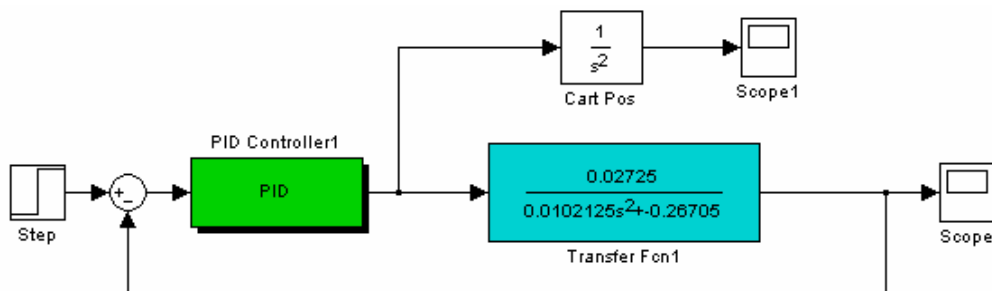


图 3-33 直线一级倒立摆 PID 控制 MATLAB 仿真模型

其中 PID Controller 为封装 (Mask) 后的 PID 控制器, 双击模块打开参数设置窗口,

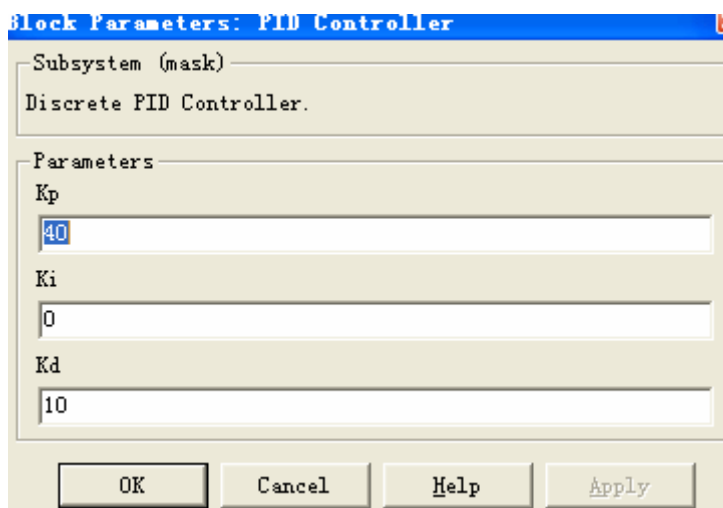


图 3-34 PID 参数设置窗口

先设置 PID 控制器为 P 控制器, 令 $K_p = 9, K_i = 0, K_d = 0$, 得到以下仿真结果:

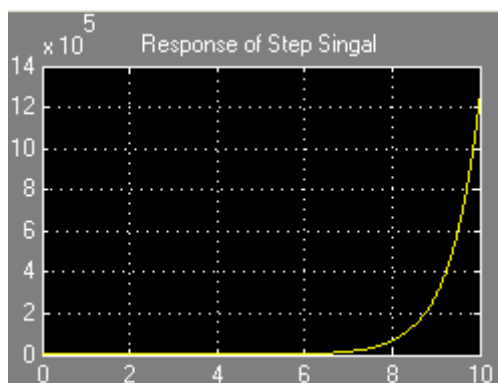
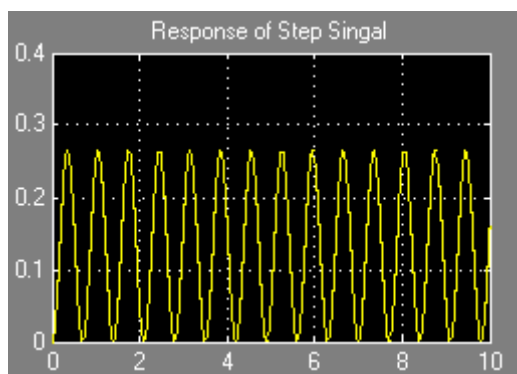


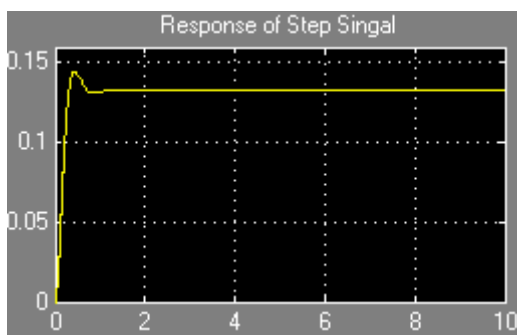
图 3-35 直线一级倒立摆 P 控制仿真结果图 ($K_p=9$)

从图中可以看出, 控制曲线不收敛, 因此增大控制量, $K_p = 40, K_i = 0, K_d = 0$ 得到以下仿真结果:

图 3-36 直线一级倒立摆 P 控制仿真结果图 ($K_p=40$)

从图中可以看出，闭环控制系统持续振荡，周期约为 0.7s。为消除系统的振荡，增加微分控制参数 K_D ，

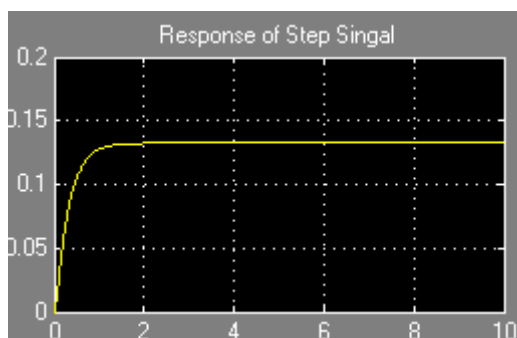
令 $K_p = 40, K_i = 0, K_D = 4$ ，得到仿真结果如下：

图 3-37 直线一级倒立摆 PD 控制仿真结果图 ($K_p=40, K_d=4$)

从图中可以看出，系统稳定时间过长，大约为 4 秒，且在两个振荡周期后才能稳定，因此再增加微分控制参数 K_D ，

令： $K_p = 40, K_i = 0, K_D = 10$

仿真得到如下结果：

图 3-38 直线一级倒立摆 PD 控制仿真结果图 ($K_p=40, K_d=10$)

从上图可以看出，系统在 1.5 秒后达到平衡，但是存在一定的稳态误差。为消除稳态误差，我们增加积分参数 K_i ，

令： $K_p = 40, K_i = 20, K_d = 10$

得到以下仿真结果：

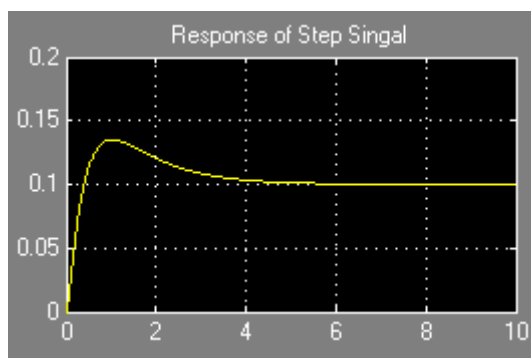


图 3-39 直线一级倒立摆 PID 控制仿真结果图 ($K_p=40, K_i=20, K_d=4$)

从上面仿真结果可以看出，系统可以较好的稳定，但由于积分因素的影响，稳定时间明显增大。

双击“Scope1”，得到小车的位置输出曲线为：

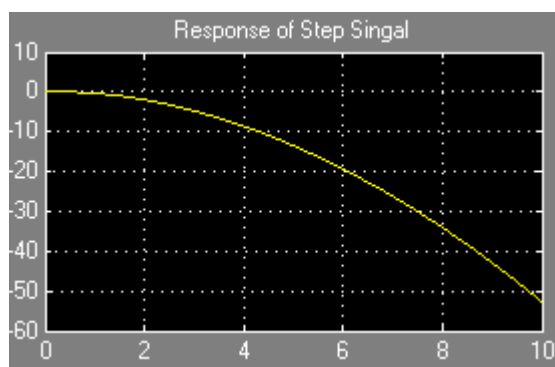


图 3-40 直线一级倒立摆 PD 控制仿真结果图（小车位置曲线）

可以看出，由于 PID 控制器为单输入单输出系统，所以只能控制摆杆的角度，并不能控制小车的位置，所以小车会往一个方向运动。

也可以采用编写 M 文件的方法进行仿真。

（进入 MATLAB Simulink 实时控制工具箱“Googol Education Products”打开“Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ PID Experiments”中的“PID Control M Files”）

PRO 3-6 直线一级倒立摆 PID 控制 MATLAB 仿真程序

```

%%%%%%%%%%%%%%
%   Googol Linear 1 stage Inverted Pendulum PID Control %
%%%%%%%%%%%%%%

clear;
num=[0.02725];
den=[0.0102125 0 -0.26705];

kd=10    %pid close loop system pendant response for impluse signal
k=40
ki=10
numPID= [ kd  k  ki ];
denPID= [ 1  0 ];
numc= conv ( num, denPID )
denc= polyadd ( conv(denPID, den ), conv( numPID, num ) )

t = 0 : 0.005 : 5;
figure(1);
impulse ( numc , denc , t )

```

运行后得到如下的仿真结果：

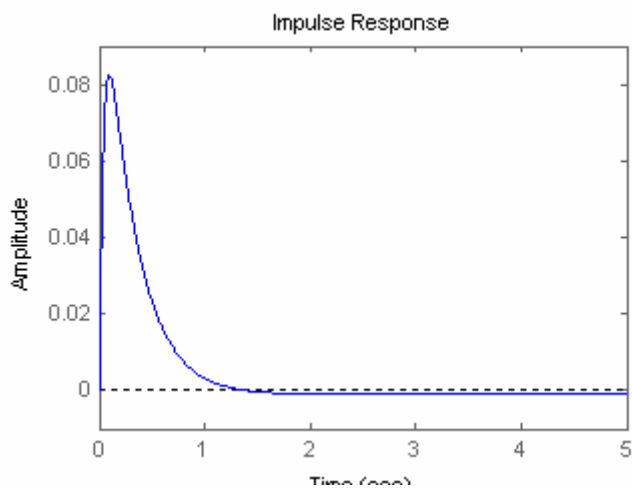


图 3-41 直线一级倒立摆 PID 控制 MATLAB 仿真结果（脉冲干扰）

3.1.5.3PID 控制实验

实时控制实验在 MATALB Simulink 环境下进行，用户在实验前请仔细阅读使用手册。



注意

- 在进行 MATLAB 实时控制实验时，请用户检查倒立摆系统机械结构和电气接线有无危险因素存在，在保障实验安全的情况下进行实验。

3.1.5.3.1 MATLAB 版实验软件下的实验步骤

- 1) 打开直线一级倒立摆 PID 控制界面入下图所示：
(进入 MATLAB Simulink 实时控制工具箱“Googol Education Products”打开“*Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\PID Experiments*”中的“PID Control Demo”)

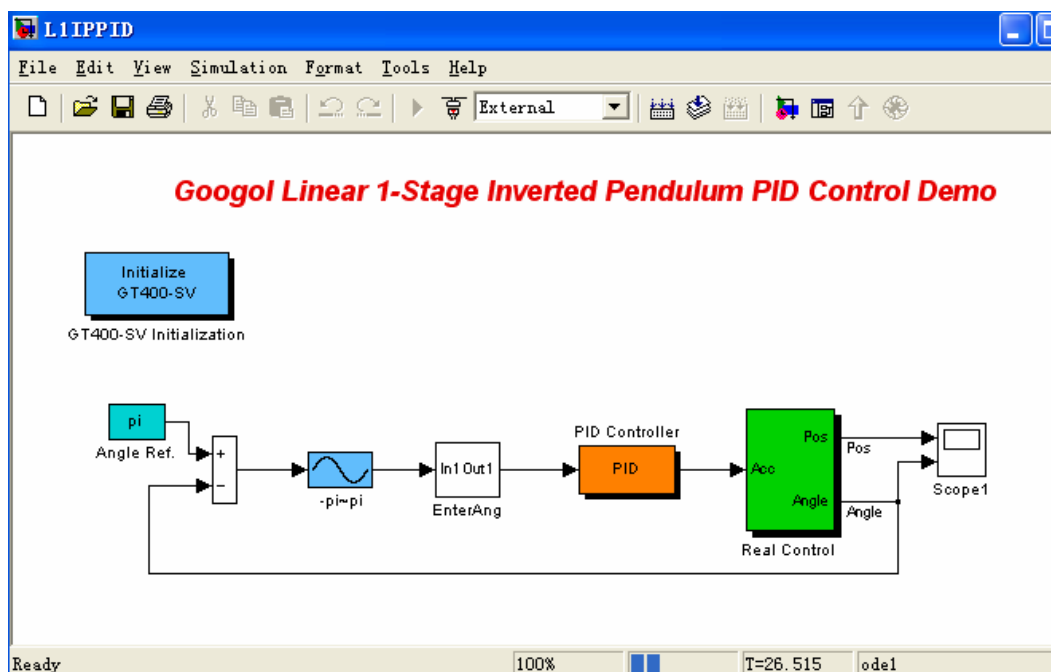
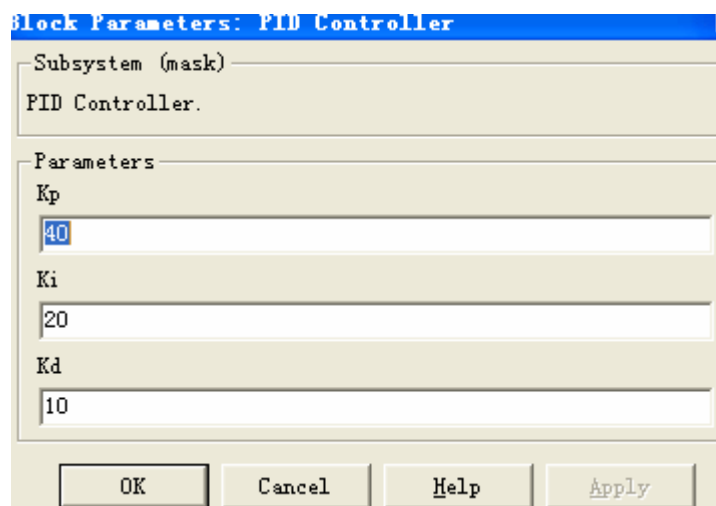


图 3-42 直线一级倒立摆 MATLAB 实时控制界面

- 2) 双击“PID”模块进入 PID 参数设置，如下图所示：



把仿真得到的参数输入 PID 控制器，点击“OK”保存参数。

- 3) 点击 编译程序，完成后点击 使计算机和倒立摆建立连接。
- 4) 点击 运行程序，检查电机是否上伺服，如果没有上伺服，请参见直线倒立摆使用手册相关章节。缓慢提起倒立摆的摆杆到竖直向上的位置，在程序进入自动控制后松开，当小车运动到正负限位的位置时，用工具挡一下摆杆，使小车反向运动。
- 5) 实验结果如下图所示：

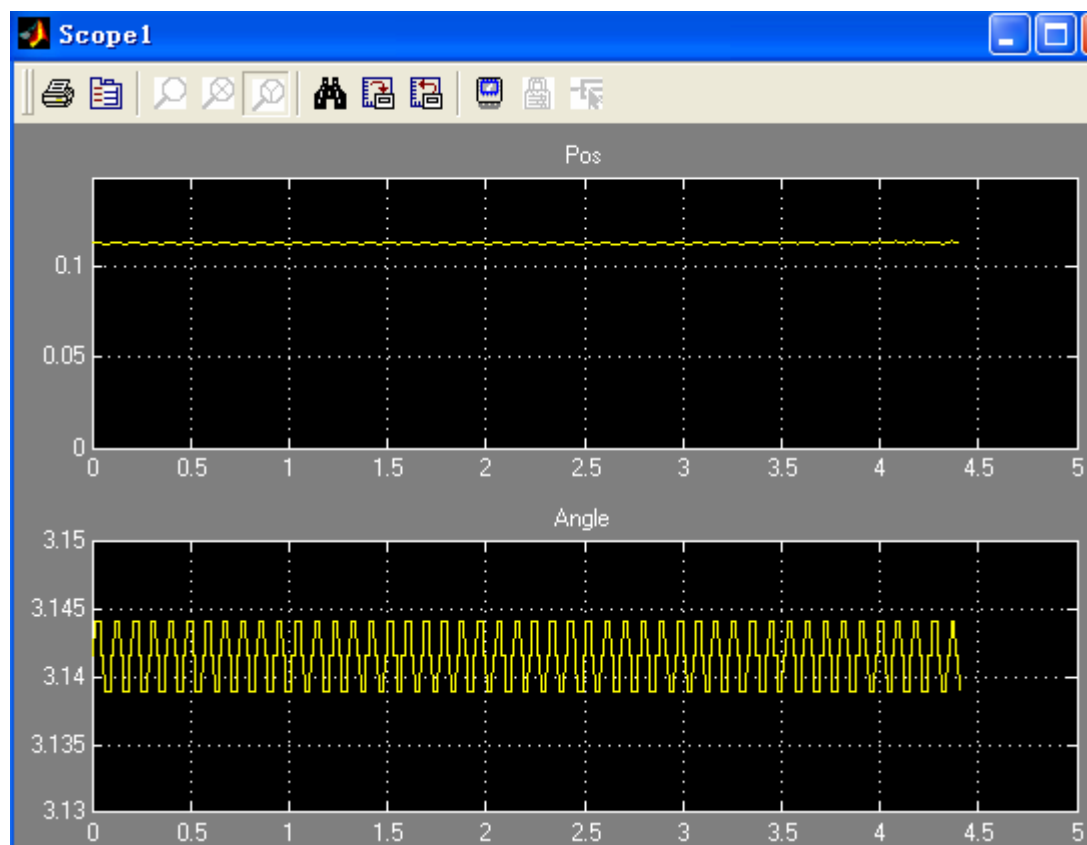


图 3-43 直线一级倒立摆 PID 控制实验结果 1

从图中可以看出，倒立摆可以实现较好的稳定性，摆杆的角度在 3.14（弧度）左右。同仿真结果，PID 控制器并不能对小车的位置进行控制，小车会沿滑杆有

稍微的移动。在给定干扰的情况下,小车位置和摆杆角度的变化曲线如下图所示:

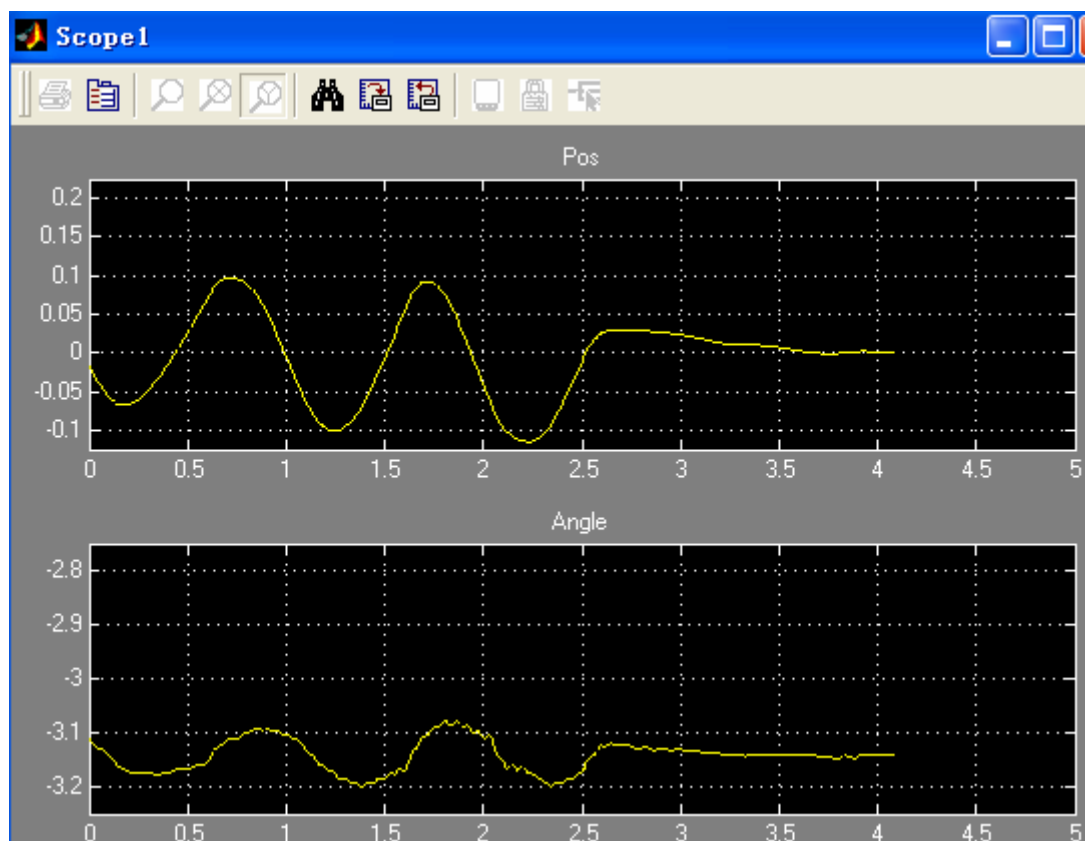
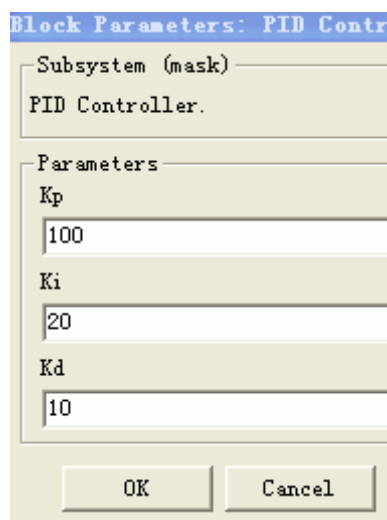


图 3-44 直线一级倒立摆 PID 控制实验结果 2 (施加干扰)

可以看出,系统可以较好的抵换外界干扰,在干扰停止作用后,系统能很快回到平衡位置。

修改 PID 控制参数,例如:



观察控制结果的变化,可以看出,系统的调整时间减少,但是在平衡的时候会出现小幅的振荡。

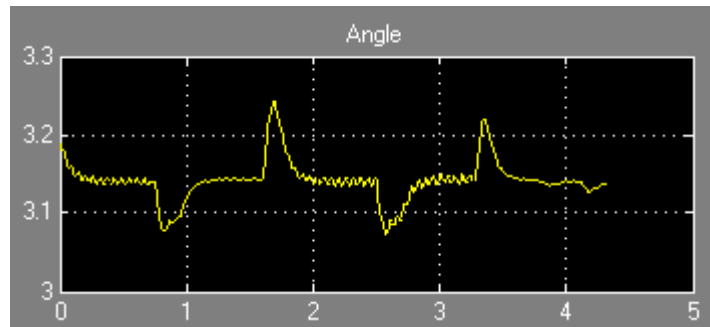


图 3-45 直线一级倒立摆 PID 控制实验结果 3（改变 PID 控制参数）

3.1.5.4 实验结果与实验报告

请将计算步骤，仿真和实验结果记录并完成实验报告：

3.1.6 状态空间极点配置控制实验

经典控制理论的研究对象主要是单输入单输出的系统，控制器设计时一般需要有关被控对象的较精确模型，现代控制理论主要是依据现代数学工具，将经典控制理论的概念扩展到多输入多输出系统。极点配置法通过设计状态反馈控制器将多变量系统的闭环系统极点配置在期望的位置上，从而使系统满足瞬态和稳态性能指标。前面我们已经得到了倒立摆系统的比较精确的动力学模型，下面我们针对直线型一级倒立摆系统应用极点配置法设计控制器。

3.1.6.1 状态空间分析

对于控制系统

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

式中

X 为状态向量（ n 维）

u 控制向量（纯量）

A $n \times n$ 维常数矩阵

B $n \times 1$ 维常数矩阵

选择控制信号为：

$$u = -KX$$

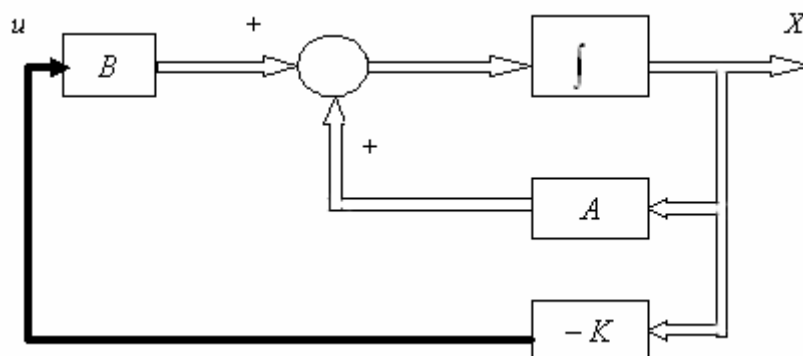


图 3-46 状态反馈闭环控制原理图

求解上式，得到

$$\dot{x}(t) = (A - BK)x(t)$$

方程的解为：

$$x(t) = e^{(A-BK)t} x(0)$$

可以看出，如果系统状态完全可控， K 选择适当，对于任意的初始状态，当 t 趋于无穷时，都可以使 $x(t)$ 趋于 0。

极点配置的设计步骤：

1) 检验系统的可控性条件。

2) 从矩阵 A 的特征多项式

$$|sI - A| = s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n$$

来确定 a_1, a_2, \cdots, a_n 的值。

3) 确定使状态方程变为可控标准型的变换矩阵 T :

$$T = MW$$

其中 M 为可控性矩阵,

$$M = [B:AB:\cdots:A^{n-1}B]$$

$$W = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & & 1 & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ a_1 & 1 & \cdots & 0 & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}$$

4) 利用所期望的特征值, 写出期望的多项式

$$(s - \mu_1)(s - \mu_2) \cdots (s - \mu_n) = s^n + \alpha_1 s^{n-1} + \cdots + \alpha_{n-1} s + \alpha_n$$

并确定 $\alpha_1, \alpha_2, \cdots, \alpha_n$ 的值。

5) 需要的状态反馈增益矩阵 K 由以下方程确定:

$$K = [\alpha_n - a_n : \alpha_{n-1} - a_{n-1} : \cdots : \alpha_2 - a_2 : \alpha_1 - a_1] T^{-1}$$

3.1.6.2 极点配置及仿真

前面我们已经得到了直线一级倒立摆的状态空间模型, 以小车加速度作为输入的系统状态方程为:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 29.4 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \end{bmatrix} u$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

于是有：

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 29.4 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

直线一级倒立摆的极点配置转化为：

对于如上所述的系统，设计控制器，要求系统具有较短的调整时间（约 3 秒）和合适的阻尼（阻尼比 $\zeta = 0.5$ ）。

下面采用四种不同的方法计算反馈矩阵 K 。

方法一：按极点配置步骤进行计算。

1) 检验系统可控性，由 3.1.1.4 系统可控性分析可以得到，系统的状态完全可控性矩阵的秩等于系统的状态维数(4)，系统的输出完全可控性矩阵的秩等于系统输出向量 y 的维数(2)，所以系统可控。

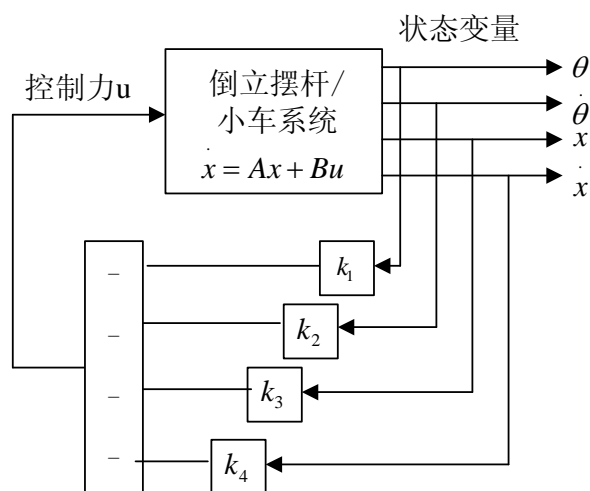


图 3-47 倒立摆极点配置原理图

2) 计算特征值

根据要求, 并留有一定的裕量(设调整时间为 2 秒), 我们选取期望的闭环极点 $s = \mu_i$ ($i = 1, 2, 3, 4$), 其中:

$$\mu_1 = -10, \mu_2 = -10, \mu_3 = -2 + j2\sqrt{3}, \mu_4 = -2 - j2\sqrt{3}$$

其中, μ_3, μ_4 使一对具有 $\zeta = 0.5, \omega_n = 4$ 的主导闭环极点, μ_1, μ_2 位于主导闭环极点的左边, 因此其影响较小, 因此期望的特征方程为:

$$\begin{aligned}(s - \mu_1)(s - \mu_2)(s - \mu_3)(s - \mu_4) &= (s + 10)(s + 10)(s + 2 - 2\sqrt{3}j)(s + 2 + 2\sqrt{3}j) \\ &= s^4 + 24s^3 + 196s^2 + 720s + 1600\end{aligned}$$

因此可以得到:

$$\alpha_1 = 24, \alpha_2 = 196, \alpha_3 = 720, \alpha_4 = 1600$$

由系统的特征方程:

$$\begin{aligned}|sI - A| &= \begin{vmatrix} s & -1 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & -1 \\ 0 & 0 & -29.4 & s \end{vmatrix} \\ &= s^4 - 29.4s^2\end{aligned}$$

因此有

$$a_1 = 0, a_2 = -29.4, a_3 = 0, a_4 = 0$$

系统的反馈增益矩阵为:

$$K = [\alpha_4 - a_4 : \alpha_3 - a_3 : \alpha_2 - a_2 : \alpha_1 - a_1]T^{-1}$$

3) 确定使状态方程变为可控标准型的变换矩阵 T:

$$T = MW$$

式中:

$$\begin{aligned}M &= [B : AB : A^2B : A^3B] \\ &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 88.2 \\ 3 & 0 & 88.2 & 0 \end{bmatrix}\end{aligned}$$

$$W = \begin{bmatrix} a_3 & a_2 & a_1 & 1 \\ a_2 & a_1 & 1 & 0 \\ a_1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -29.4 & 0 & 1 \\ -29.4 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

于是可以得到:

$$T = MW = \begin{bmatrix} -29.4 & 0 & 1 & 0 \\ 0 & -29.4 & 0 & 1 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

$$T^{-1} = \begin{bmatrix} -0.034 & 0 & 0.0113 & 0 \\ 0 & -0.034 & 0 & 0.0113 \\ 0 & 0 & 0.3333 & 0 \\ 0 & 0 & 0 & 0.3333 \end{bmatrix}$$

4) 于是有状态反馈增益矩阵 K 为:

$$K = [\alpha_4 - a_4 : \alpha_3 - a_3 : \alpha_2 - a_2 : \alpha_1 - a_1] T^{-1}$$

$$= [1600 - 0 : 720 - 0 : 196 + 29.4 : 24 - 0] \begin{bmatrix} -0.034 & 0 & 0.0113 & 0 \\ 0 & -0.034 & 0 & 0.0113 \\ 0 & 0 & 0.3333 & 0 \\ 0 & 0 & 0 & 0.3333 \end{bmatrix}$$

$$= [-54.4218 \quad -24.4898 \quad 93.2739 \quad 16.1633]$$

得到控制量为:

$$\mu = -KX = 54.4218x + 24.4898\dot{x} - 93.2739\phi - 16.1633\dot{\phi}$$

以上计算可以采用 MATLAB 编程计算。

运行得到以下结果:

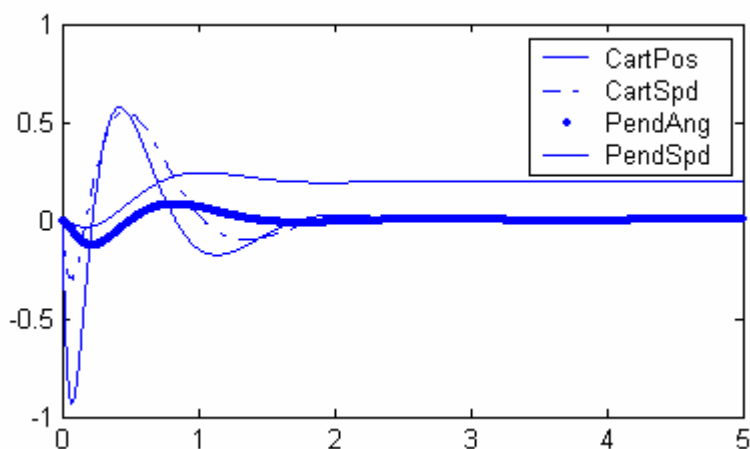


图 3-48 极点配置仿真结果

可以看出, 在给定系统干扰后, 倒立摆可以在 2 秒内很好的回到平衡位置, 满足

设计要求。

PRO 3-7 直线一级倒立摆状态空间极点配置 MATLAB 程序 1

```

%%%%%%%%%%%%%%
Googol Linear 1 stage Inverted Pendulum Poles Placement Method1 %
%%%%%%%%%%%%%%
clear;

A=[ 0  1  0  0; 0  0  0  0; 0  0  0  1; 0  0 29.4 0];

B=[ 0  1  0  3]';

C=[ 1  0  0  0; 0  0  1  0];
D=[ 0 0 ]';

J=[ -10  0  0  0; 0  -10  0  0; 0  0  -2*sqrt(3)*i  0;
    0  0  0 -2+2*sqrt(3)*i];

pa=poly(A);pj=poly(J);

M=[B A*B A^2*B A^3*B];

W=[ pa(4)  pa(3)  pa(2)  1; pa(3)  pa(2)  1  0;
    pa(2)  1  0  0; 1  0  0  0];

T=M*W;
K=[pj(5)-pa(5)  pj(4)-pa(4)  pj(3)-pa(3)  pj(2)-pa(2)]*inv(T)

Ac = [(A-B*K)];
Bc = [B]; Cc = [C]; Dc = [D];

T=0:0.005:5;
U=0.2*ones(size(T));
Cn=[1 0 0 0];
Nbar=rscale(A,B,Cn,0,K);
Bcn=[Nbar*B];
[Y,X]=lsim(Ac,Bcn,Cc,Dc,U,T);

plot(T,X(:,1),'-'); hold on;
plot(T,X(:,2),'-'); hold on;
plot(T,X(:,3),'-'); hold on;
plot(T,X(:,4),'-')
legend('CartPos','CartSpd','PendAng','PendSpd')

```

(进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ Poles Experiments” 中的 “Poles Control M File1”)

方法二：读者还可以通过下面的方法进行极点配置计算：

矩阵 $(A-BK)$ 的特征值是方程式 $|Is-(A-BK)|=0$ 的根：

$$\begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & a & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ b \end{bmatrix} \times [k_1 \quad k_2 \quad k_3 \quad k_4] = 0$$

这是 s 的四次代数方程式，可表示为

$$s^4 + (k_2 + bk_4)s^3 + (-a + k_1 + bk_3)s^2 - ak_2s - ak_1 = 0$$

适当选择反馈系数 k_1, k_2, k_3, k_4 系统的特征根可以取得所希望的值。

把四个特征根 $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ 设为四次代数方程式的根，则有

$$\begin{aligned} & s^4 - (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4)s^3 \\ & + (\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_4 + \lambda_4\lambda_1 + \lambda_1\lambda_3 + \lambda_2\lambda_4)s^2 \\ & - (\lambda_1\lambda_2\lambda_3 + \lambda_2\lambda_3\lambda_4 + \lambda_1\lambda_3\lambda_4 + \lambda_4\lambda_1\lambda_2)s \\ & + \lambda_1\lambda_2\lambda_3\lambda_4 = 0 \end{aligned}$$

比较两式有下列联立方程式

$$\begin{aligned} k_2 + bk_4 &= -(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) \\ -a + k_1 + bk_3 &= \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_4 + \lambda_4\lambda_1 + \lambda_1\lambda_3 + \lambda_2\lambda_4 \\ -ak_2 &= -(\lambda_1\lambda_2\lambda_3 + \lambda_2\lambda_3\lambda_4 + \lambda_1\lambda_3\lambda_4 + \lambda_4\lambda_1\lambda_2) \\ -ak_1 &= \lambda_1\lambda_2\lambda_3\lambda_4 \end{aligned}$$

如果给出的 $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ 是实数或共轭复数，则联立方程式的右边全部为

实数。据此可求解出实数 k_1, k_2, k_3, k_4

当将特征根指定为下列两组共轭复数时

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4 = -2 \pm 2\sqrt{3}j, -10, -10$$

又 $a = 29.4, b = 3$

利用方程式可列出关于 k_1, k_2, k_3, k_4 的方程组：

$$\begin{aligned} k_2 + 3k_4 &= 24 \\ -29.4 + k_1 + 3k_3 &= 196 \\ -29.4k_2 &= 720 \\ -29.4k_1 &= 1600 \end{aligned}$$

求解后得

$$k_1 = -54.4218 \quad k_2 = -24.4898 \quad k_3 = 93.2739 \quad k_4 = 16.1633$$

即施加在小车水平方向的控制力 u ：

$$\mu = -KX = 54.4218x + 24.4898\dot{x} - 93.2739\phi - 16.1633\dot{\phi}$$

可以看出，和方法一的计算结果一样。

PRO 3-8 直线一级倒立摆状态空间极点配置 MATLAB 程序 2

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Googol Linear 1 stage Inverted Pendulum Poles Placement Method2 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear;
syms a b k1 k2 k3 k4;

A=[ 0  1  0  0;
    0  0  0  0;
    0  0  0  1;
    0  0  a  0];
B=[ 0  1  0  b]';

SS=[ s  0  0  0;
     0  s  0  0;
     0  0  s  0;
     0  0  0  s];
K=[k1 k2 k3 k4];

J=[ -10  0      0      0;
     0  -10     0      0;
     0  0  -2*sqrt(3)*i  0;
     0  0      0 -2+2*sqrt(3)*i];

ans=A-B*K;
P=poly(ans)
PJ=poly(J)

```

（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ Poles Experiments” 中的 “Poles Control M File2”）

方法三：利用爱克曼公式计算。

爱克曼方程所确定的反馈增益矩阵为：

$$K = [0 \quad 0 \quad \cdots \quad 0 \quad 1] [B : AB : \cdots : A^{n-1}B]^{-1} \phi(A)$$

其中 $\phi(A) = A^3 + \alpha_1 A^2 + \alpha_2 A + \alpha_3 I$

利用 MATLAB 可以方便的计算，程序如下：

PRO 3-9 直线一级倒立摆状态空间极点配置 MATLAB 程序 3 (爱克曼公式)

```

clear;

A=[ 0  1  0  0;
    0  0  0  0;
    0  0  0  1;
    0  0 29.4 0];
B=[ 0  1  0  3]';

M=[B A*B A^2*B A^3*B];

J=[ -10  0      0      0;
     0 -10      0      0;
     0  0 -2-2*sqrt(3)*i  0;
     0  0      0 -2+2*sqrt(3)*i];

phi=polyvalm(poly(J),A);
K=[ 0 0 0 1]*inv(M)*phi

```

(进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ Poles Experiments” 中的 “Poles Control M File3”)

运行可以得到:

```

K =

    -54.4218   -24.4898    93.2739    16.1633

```

可以看出, 计算结果和前面两种方法一致。

方法四:

可以直接利用 MATLAB 的极点配置函数

$$[K, \text{PREC}, \text{MESSAGE}] = \text{PLACE}(A, B, P)$$

来计算。

(进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ Poles Experiments” 中的 “Poles Control M File4”)

PRO 3-10 直线一级倒立摆状态空间极点配置 MATLAB 程序 4（爱克曼公式）

```

clear;

A=[ 0  1  0  0;
    0  0  0  0;
    0  0  0  1;
    0  0  29.4  0];
B=[ 0  1  0  3]';

P=[-10-0.0001*j,-10+0.0001*j,-2-2*sqrt(3)*j,-2+2*sqrt(3)*j];

K=place(A,B,P);

```

为匹配 place() 函数，把-10,-10 两个极点改成了 $-10 - 0.0001j$, $-10 + 0.0001j$, 因为增加的虚部很小，可以忽略不计，运行得到如下结果：

```

K =

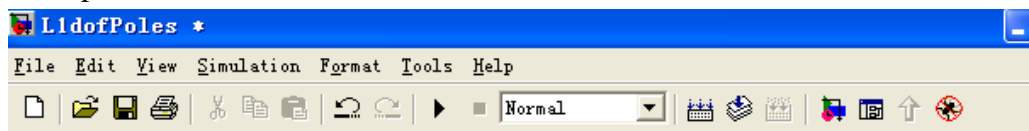
-54.4218  -24.4898   93.2739   16.1633

```

可以看出，以上四种方法计算结果都保持一致。

下面对以上的计算结果在 MATLAB Simulink 中进行仿真，打开直线一级倒立摆的仿真模型：

（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ Poles Experiments” 中的 “Poles Control Simulink”）



Googol Linear 1-Stage Inverted Pendulum Poles Control Simulink

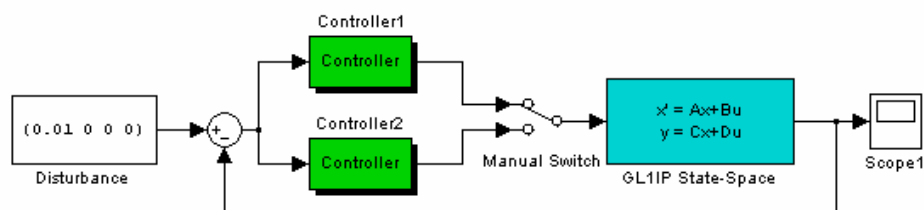
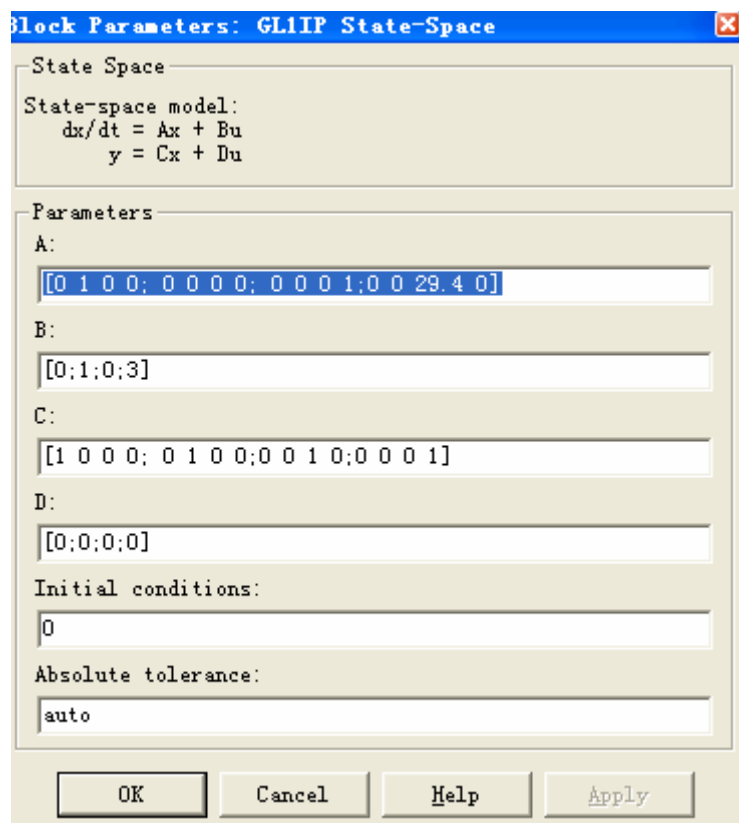
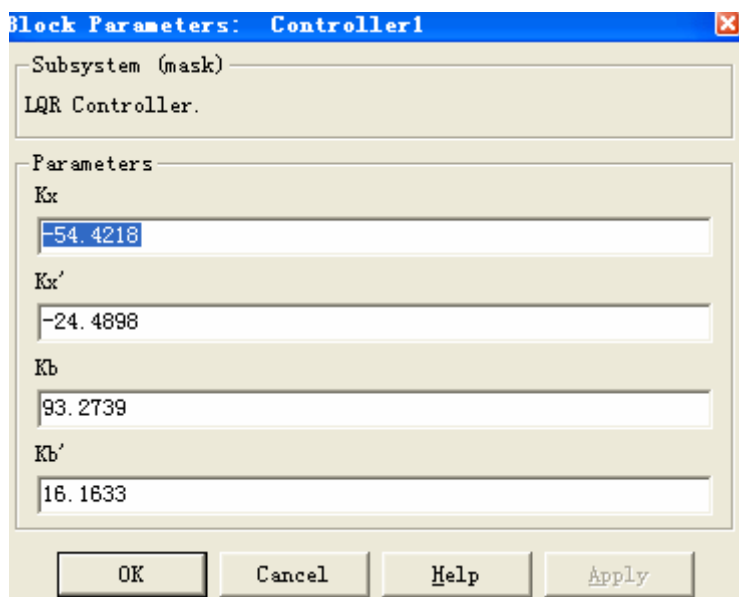


图 3-49 直线一级倒立摆状态空间极点配置 MATLAB Simulink 仿真

其中 “GL1IP State-Space” 为直线一级倒立摆的状态空间模型，双击打开如下窗口：



双击“Controller1”模块，打开状态反馈矩阵 K 设置窗口：



把计算得到的 K 值输入上面的窗口。

运行仿真，得到以下结果：

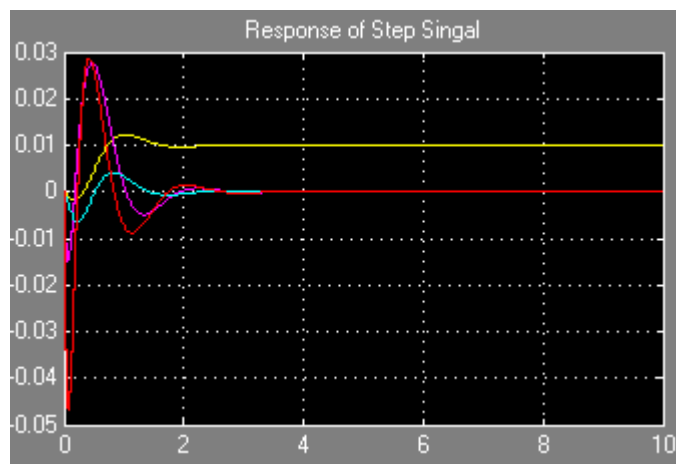
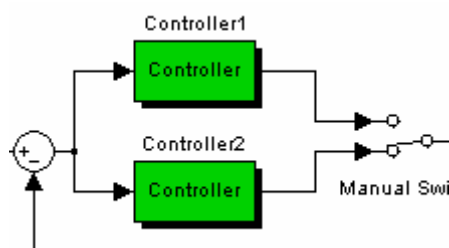


图 3-50 直线一级倒立摆状态空间极点配置 MATLAB Simulink 仿真结果

可以看出，在存在干扰的情况下，系统在 3 秒内基本上可以恢复到新的平衡位置，

读者可以修改期望的性能指标，进行新的极点配置，在“Controller2”模块中设置新的控制参数，并点击“Manual Switch”把控制信号切换到“Controller2”



3.1.6.3 极点配置控制实验

实验步骤如下：

- 1) 进入 MATLAB Simulink 中 “[\\matlab6p5\toolbox\GoogolTech\InvertedPendulum\Linear Inverted Pendulum](http://matlab6p5.toolbox/GoogolTech/InvertedPendulum/Linear%20Inverted%20Pendulum),” 目录，打开直线一级倒立摆状态空间极点配置控制程序如下：

（进入 MATLAB Simulink 实时控制工具箱“Googol Education Products”打开“Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\Poles Experiments”中的“Poles Control Demo”）

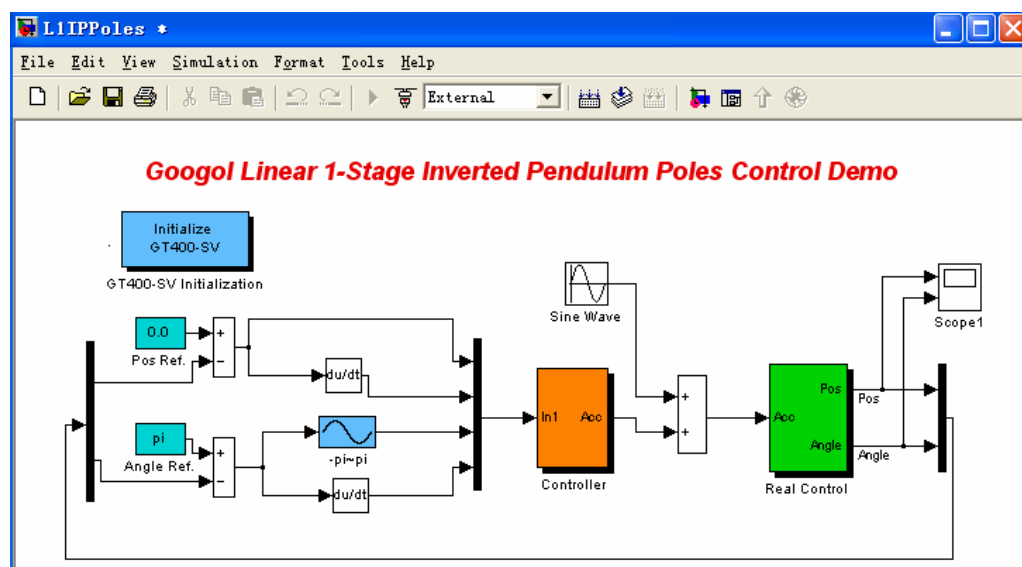
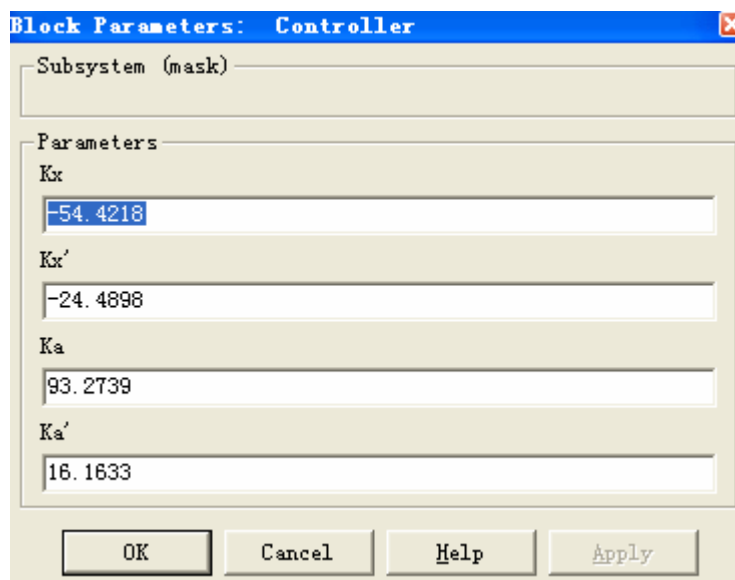


图 3-51 直线一级倒立摆状态空间极点配置实时控制程序

- 2) 点击“Controller”模块设置控制器参数，把前面仿真结果较好的参数输入到模块中：



点击“OK”完成设定。

- 3) 点击 编译程序，完成后点击 使计算机和倒立摆建立连接。
- 4) 点击 运行程序，检查电机是否上伺服，如果没有上伺服，请参见直线倒立摆使用手册相关章节。提起倒立摆的摆杆到竖直向上的位置，在程序进入自动控制后松开。
- 5) 双击“Scope”观察实验结果如下图所示：

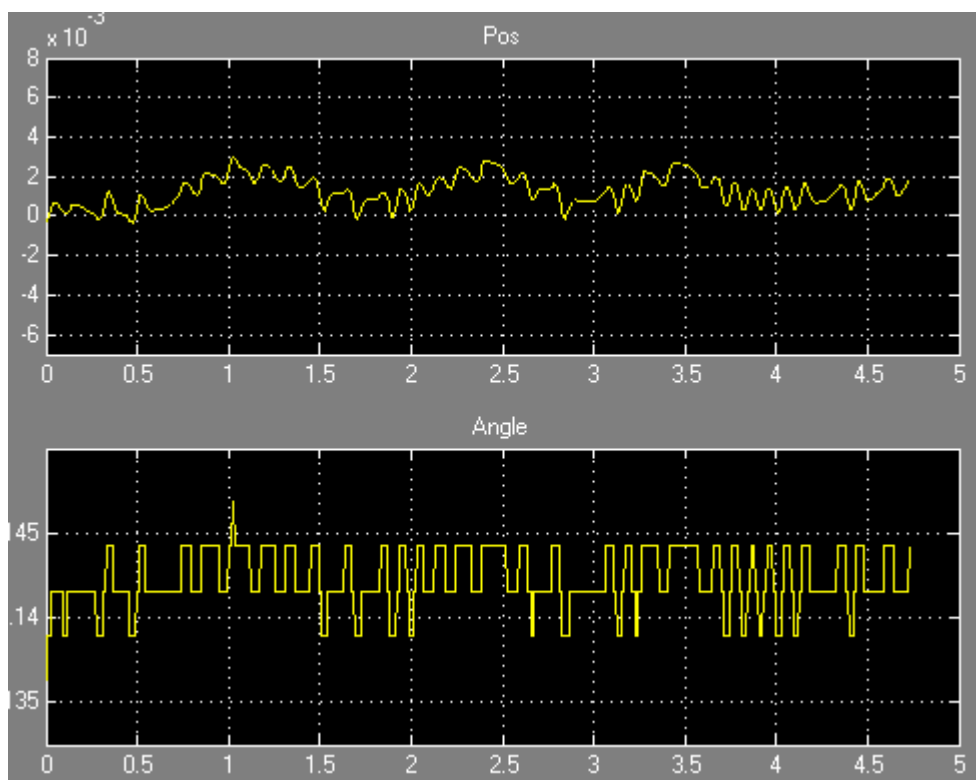


图 3-52 直线一级倒立摆状态空间极点配置实时控制结果（平衡）

可以看出，系统可以在很小的振动范围内保持平衡，小车振动幅值约为 $4 \times 10^{-3} \text{ m}$ ，摆杆振动的幅值约为 0.05 弧度，注意，不同的控制参数会有不同的控制结果。

在给定倒立摆干扰后，系统如响应如下图所示：

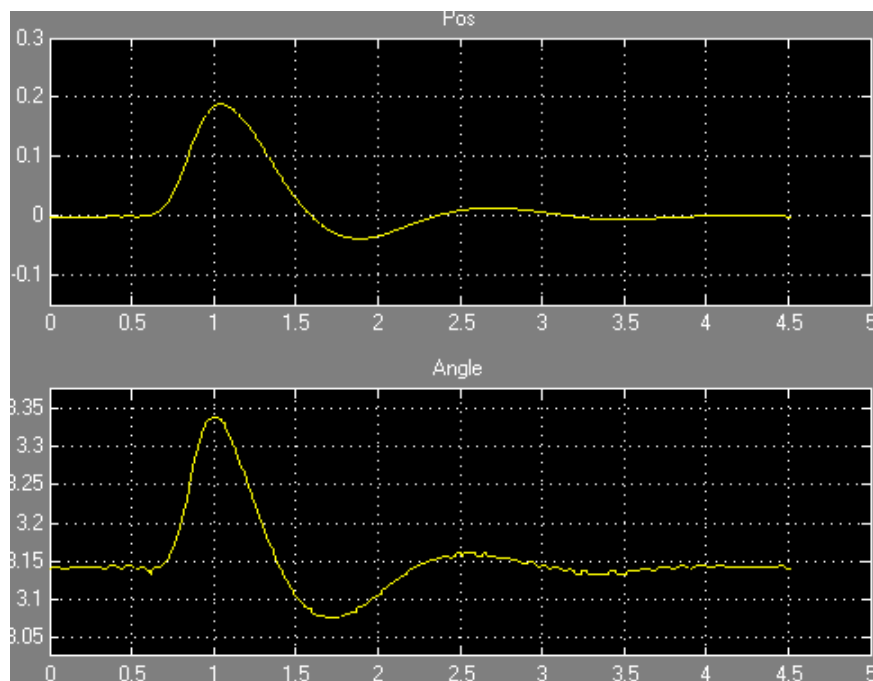


图 3-53 直线一级倒立摆状态空间极点配置实时控制结果（施加干扰）

从上图可以看出，系统稳定时间约为 3 秒，达到设计要求。

3.1.6.4 实验结果及实验报告

上机实验并记录实验结果，完成实验报告。

提示：用户可以在极点配置实验的基础上添加状态观测器，对系统进行设计和仿真以及实验。

3.1.7 线性二次最优控制 LQR 控制实验

本实验的主要内容是让实验者了解并掌握线性二次最优控制 LQR 控制的原理和方法，并对直线一级倒立摆进行控制实验。

3.1.7.1 线性二次最优控制 LQR 基本原理及分析

线性二次最优控制 LQR 基本原理为，由系统方程：

$$\dot{X} = AX + Bu \quad (4-1)$$

确定下列最佳控制向量的矩阵 K：

$$u(t) = -K^*x(t) \quad (4-2)$$

使得性能指标达到最小值：

$$J = \int_0^{\infty} (X^* Q X + u^* R u) dt \quad (4-3)$$

式中 Q ——正定(或正半定)厄米特或实对称阵

R ——为正定厄米特或实对称阵

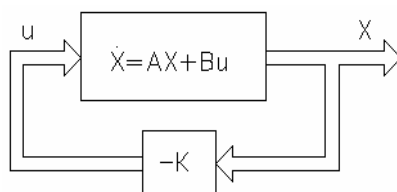


图 3-54 最优控制 LQR 控制原理图

方程右端第二项是考虑到控制能量的损耗而引进的，矩阵 Q 和 R 确定了误差和能量损耗的相对重要性。并且假设控制向量 $u(t)$ 是无约束的。

对线性系统：

$$\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases}$$

根据期望性能指标选取 Q 和 R ，利用 MATLAB 命令 `lqr` 就可以得到反馈矩阵 K 的值。

$$K = \text{lqr}(A, B, Q, R)$$

改变矩阵 Q 的值，可以得到不同的响应效果， Q 的值越大(在一定的范围之内)，系统抵抗干扰的能力越强，调整时间越短。但是 Q 不能过大，其影响将在实验结果分析中阐述。

关于线性二次最优控制 LQR 的详细原理请参见现代控制理论的相关书籍。

3.1.7.2 LQR 控制参数调节及仿真

前面我们已经得到了直线一级倒立摆系统的比较精确的动力学模型，下面我们针对直线型一级倒立摆系统应用 LQR 法设计与调节控制器，控制摆杆保持竖

直向上平衡的同时，跟踪小车的位置。

前面我们已经得到了直线一级倒立摆系统的系统状态方程。：

$$\begin{bmatrix} \dot{x} \\ \dot{\dot{x}} \\ \dot{\phi} \\ \dot{\dot{\phi}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 29.4 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 3 \end{bmatrix} u$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

应用线性反馈控制器，控制系统结构如下图。图中 R 是施加在小车上的阶跃输入，四个状态量 $x, \dot{x}, \phi, \dot{\phi}$ 分别代表小车位移、小车速度、摆杆角度和摆杆角速度，输出 $y = [x, \phi]'$ 包括小车位置和摆杆角度。设计控制器使得当给系统施加一个阶跃输入时，摆杆会摆动，然后仍然回到垂直位置，小车可以到达新的指定位置。

假设全状态反馈可以实现（四个状态量都可测），找出确定反馈控制规律的向量 K 。在 Matlab 中得到最优控制器对应的 K 。Lqr 函数允许你选择两个参数—— R 和 Q ，这两个参数用来平衡输入量和状态量的权重。最简单的情况是假设 $R=1$ ， $Q=C'*C$ 。当然，也可以通过改变 Q 矩阵中的非零元素来调节控制器以得到期望的响应。

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

其中， $Q_{1,1}$ 代表小车位置的权重，而 $Q_{3,3}$ 是摆杆角度的权重，输入的权重 R 是 1。

下面来求矩阵 K ，Matlab 语句为 $K = lqr(A, B, Q, R)$ 。下面在 MATLAB 中编程计算：

（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ LQR Experiments” 中的 “LQR Control MFiles”）

PRO 3-11 直线一级倒立摆 LQR 控制 MATLAB 计算程序

```

%%%%%%%%%%%%%%
%   Googol Linear 1 stage Inverted Pendulum LQR Control   %
%%%%%%%%%%%%%%
clear;
A=[ 0   1   0   0;
    0   0   0   0;
    0   0   0   1;
    0   0 29.4 0];
B=[ 0   1   0   3]';
C=[ 1   0   0   0;
    0   0   1   0];
D=[ 0 0 ]';

Q11=1000; Q33=200;
Q=[Q11 0 0 0;
    0 0 0 0;
    0 0 Q33 0;
    0 0 0 0];
R = 1;
K = lqr(A,B,Q,R)

Ac = [(A-B*K)]; Bc = [B]; Cc = [C]; Dc = [D];

T=0:0.005:5;
U=0.2*ones(size(T));
Cn=[1 0 0 0];
Nbar=rscale(A,B,Cn,0,K);
Bcn=[Nbar*B];
[Y,X]=lsim(Ac,Bcn,Cc,Dc,U,T);

plot(T,X(:,1),'-');hold on;
plot(T,X(:,2),'-');hold on;
plot(T,X(:,3),'-');hold on;
plot(T,X(:,4),'-');
legend('CartPos','CartSpd','PendAng','PendSpd')

```

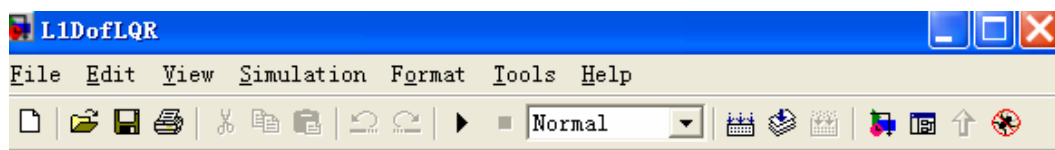
令 $Q_{1,1} = 1$, $Q_{3,3} = 1$ 求得

$$K = [-1 \quad -1.7855 \quad 25.422 \quad 4.6849]$$

在 Simulink 中建立直线一级倒立摆的模型如下图所示:

(进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ LQR

Experiments” 中的 “LQR Control Simulink”)



Googol Linear 1-Stage Inverted Pendulum LQR Control Simulink

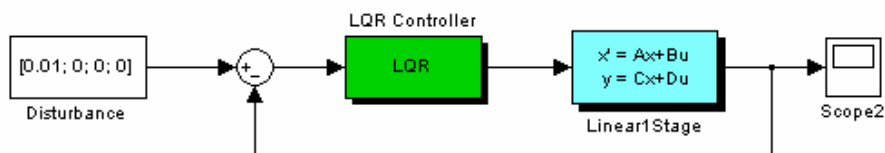
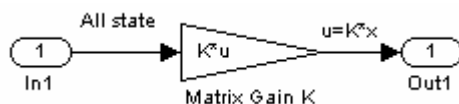
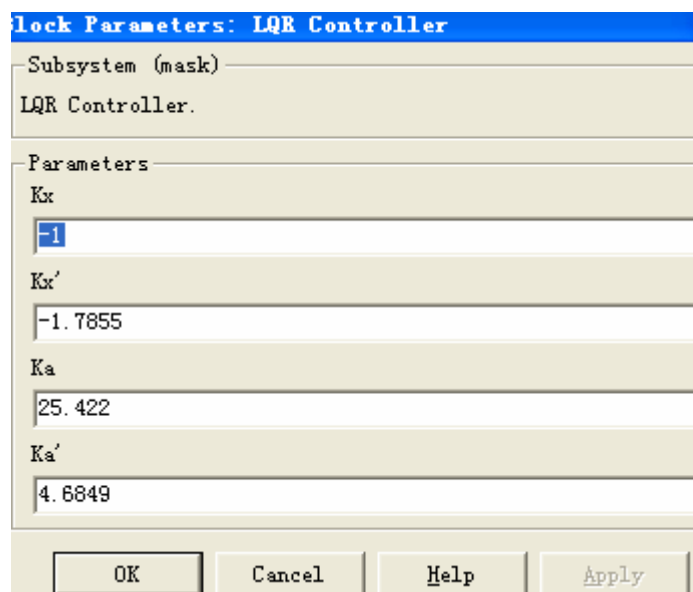


图 3-55 直线一级倒立摆 LQR 控制仿真模型

“LQR Controller” 为一封装好的模块，在其上单击鼠标右键，选择 “Look under mask” 打开 LQR Controller 结构如下：



双击 “Matrix gain K” 即可输入控制参数：



点击▶执行仿真，得到如下仿真结果：

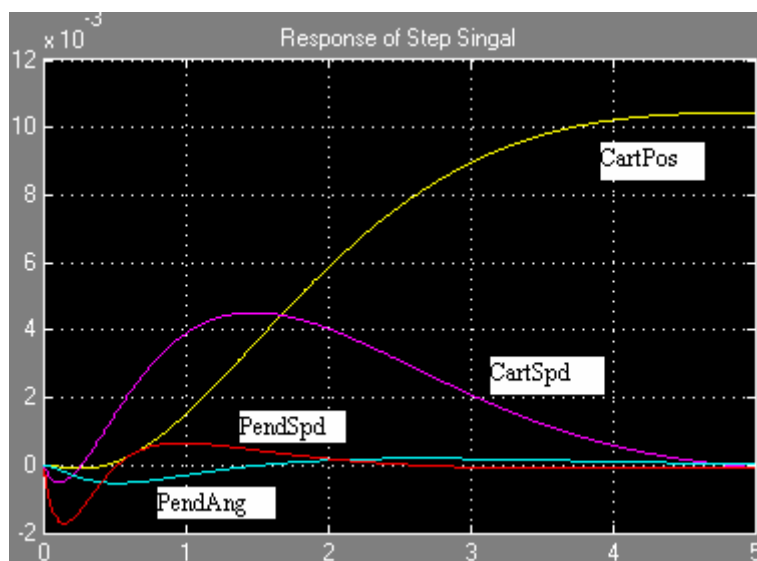


图 3-56 直线一级倒立摆 LQR 控制仿真结果 1

LQR 控制的阶跃响应如上图所示, 其中 CartPos 为小车的位置曲线, CartSpd 为小车的速度曲线, PendAng 为摆杆角度曲线, PendSpd 为摆杆角速度曲线, 从图中可以看出, 闭环控制系统响应的超调量很小, 但稳定时间和上升时间偏大, 我们可以通过增大控制量来缩短稳定时间和上升时间。

可以发现, Q 矩阵中, 增加 Q_{11} 使稳定时间和上升时间变短, 并且使摆杆的角度变化减小。这里取 $Q_{1,1}=1000$, $Q_{3,3}=200$,

则 $K = [-31.623 \quad -20.151 \quad 72.718 \quad 13.155]$

输入参数, 运行得到响应曲线如下:

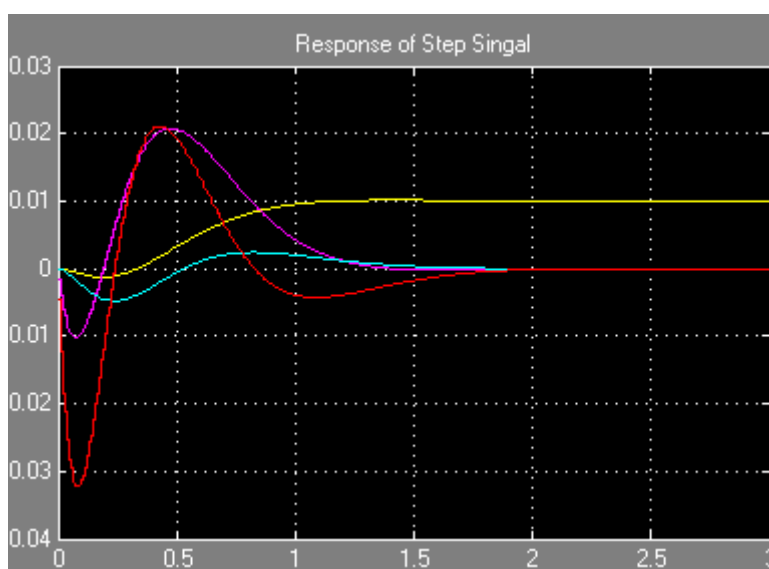


图 3-57 直线一级倒立摆 LQR 控制仿真结果 2

从图中可以看出, 系统响应时间有明显的改善, 增大 $Q_{1,1}$ 和 $Q_{3,3}$, 系统的响应还会更快, 但是对于实际离散控制系统, 过大的控制量会引起系统振荡。

3.1.7.3 直线一级倒立摆 LQR 控制实验

实时控制实验在 MATLAB Simulink 环境下进行，用户在实验前请仔细阅读使用手册。



注意

- 在进行 MATLAB 实时控制实验时，请用户检查倒立摆系统机械结构和电气接线有无危险因素存在，在保障实验安全的情况下进行实验。

操作步骤：

- 1) 打开直线一级倒立摆 LQR 实时控制模块，
（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Experiment\ LQR Experiments” 中的 “LQR Control Demo”）

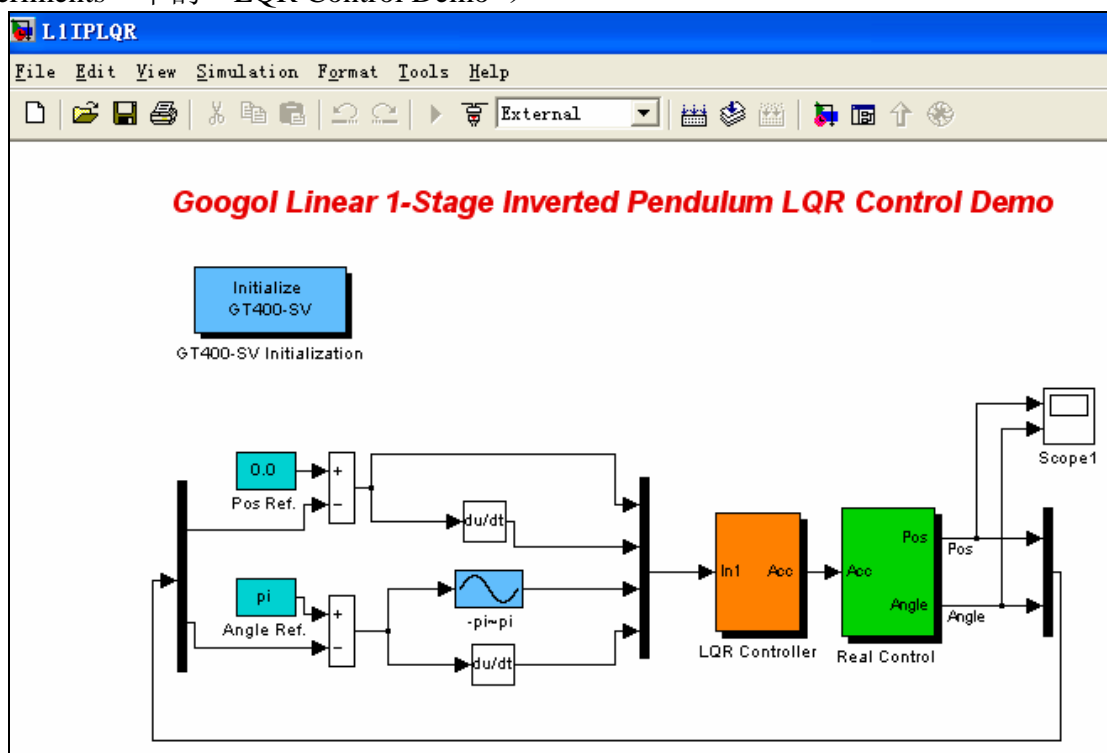


图 3-58 直线一级倒立摆 LQR 控制实时控制程序

其中 “LQR Controller” 为 LQR 控制器模块，“Real Control” 为实时控制模块，双击 “LQR Controller” 模块打开 LQR 控制器参数设置窗口如下：

Block Parameters: LQR Controller

Subsystem (mask)

Parameters

Kx

-31.623

Kx'

-20.151

Ka

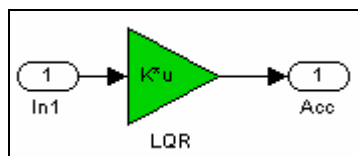
72.718

Ka'

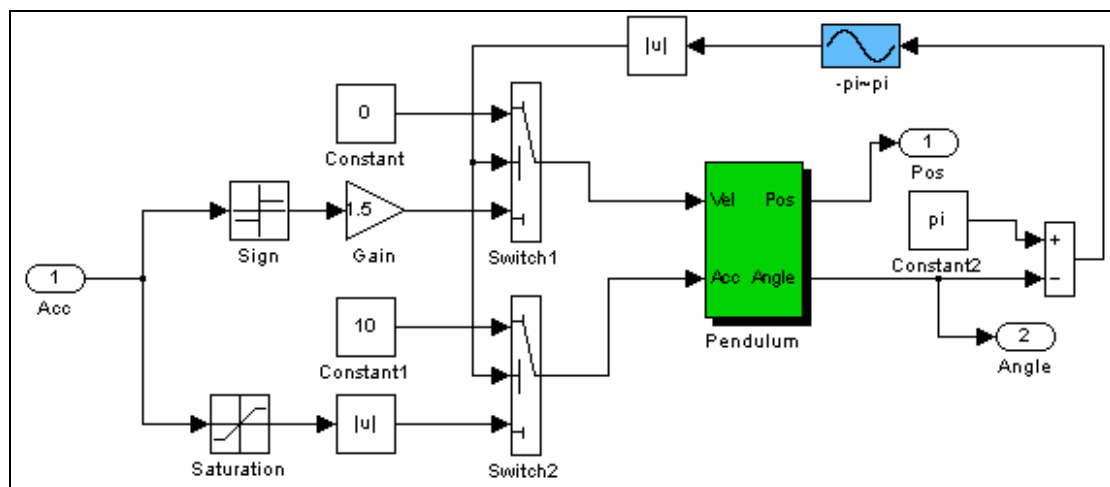
13.155

OK Cancel Help Apply

在“LQR Controller”模块上点击鼠标右键选择“Look under mask”打开模型如下：

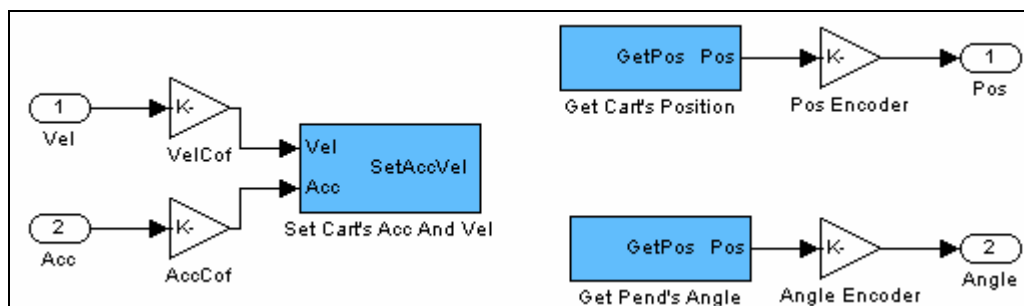


双击“Real Control”模块打开实时控制模块如下图:






其中“Pendulum”模块为倒立摆系统输入输出模块,输入为小车的速度“Vel”和“Acc”,输出为小车的位置“Pos”和摆杆的角度“Angle”。

双击“Pendulum”模块打开其内部结构:



其中“Set Cart's Acc and Vel”模块的作用是设置小车运动的速度和加速度，“Get Cart's Position”模块的作用是读取小车当前的实际位置，“Get Pend's Angle”的作用是读取摆杆当前的实际角度。更为详细的说明请参见直线倒立摆使用手册。

- 2) 点击  编译程序，编译成功后点击  连接，再点击  运行程序，在确认点击上伺服后，缓慢的提起摆杆到竖直向上的位置，程序进入自动控制后松开摆杆。
- 3) 实验运行结果如下图所示：

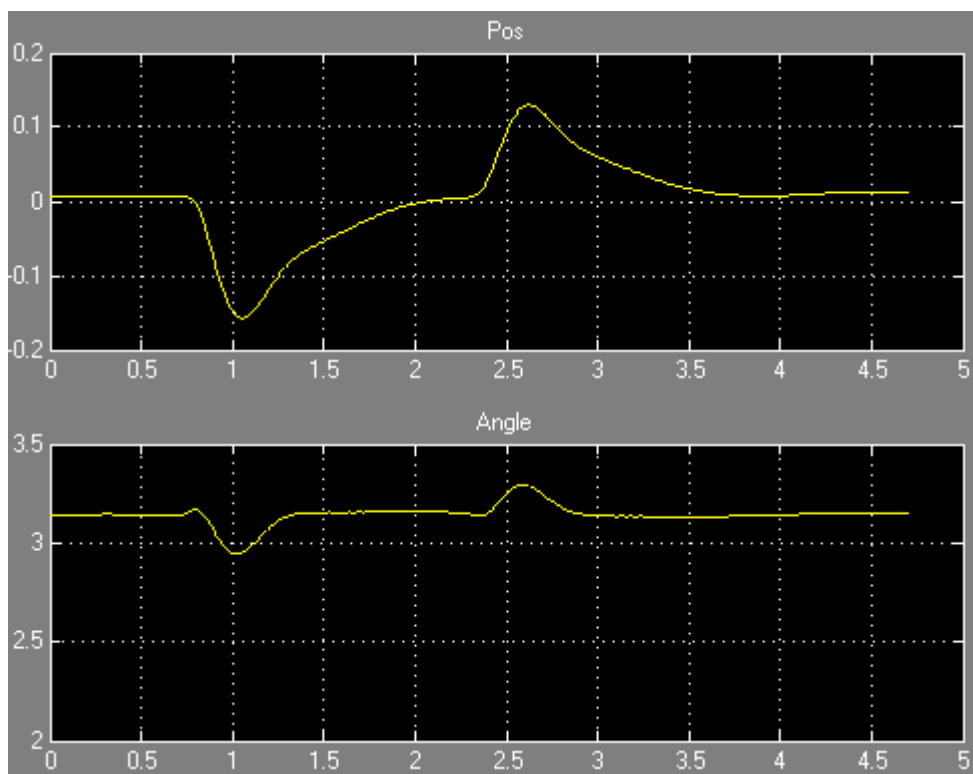


图 3-59 直线一级倒立摆 LQR 控制实时控制结果

其中图片上半部分为小车的位置曲线，下半部分为摆杆角度的变化曲线，从图中可以看出，在给定外界干扰后，小车可以在 1.5 秒内回到平衡位置。达到了较好的控制效果。

- 4) 改变 Q 矩阵的值 Q_{11} 和 Q_{33} ，再把仿真得到的 LQR 控制参数输入实时控制程序，运行实时控制程序，观察控制效果的变化。



注意

- 在进行 MATLAB 实时控制实验时，请用户不要过大的改变控制器参数，过大的参数会使系统有失控的危险，请在保障实验安全的情况下进行实验。

3.1.7.4 实验结果及实验报告

请将计算步骤，仿真和实验结果记录并完成实验报告。

第4章 直线一级顺摆建模和实验

直线一级倒立摆的摆杆在没有外力作用下，会保持静止下垂的状态，当受到外力作用后，摆杆的运动状态和钟摆类似，如果不存在摩擦力的作用，摆杆将持续摆动，很多情况下，我们并不希望出现这种持续振荡的情况，例如吊车在吊动物体的时候，我们希望物体能过很快地停止到指定的位置。下面我们对直线一级顺摆进行建模分析，并对其进行仿真和控制。

4.1 直线一级顺摆的建模与分析

4.1.1 直线一级顺摆的建模

同直线一级倒立摆的物理模型相似，可以采用牛顿力学和拉格朗日方法进行建模和分析，对于牛顿力学方法，这里不再进行分析和计算，读者可以参考直线一级倒立摆的物理模型对其进行建模，下面采用拉格朗日方法对直线一级顺摆进行建模。

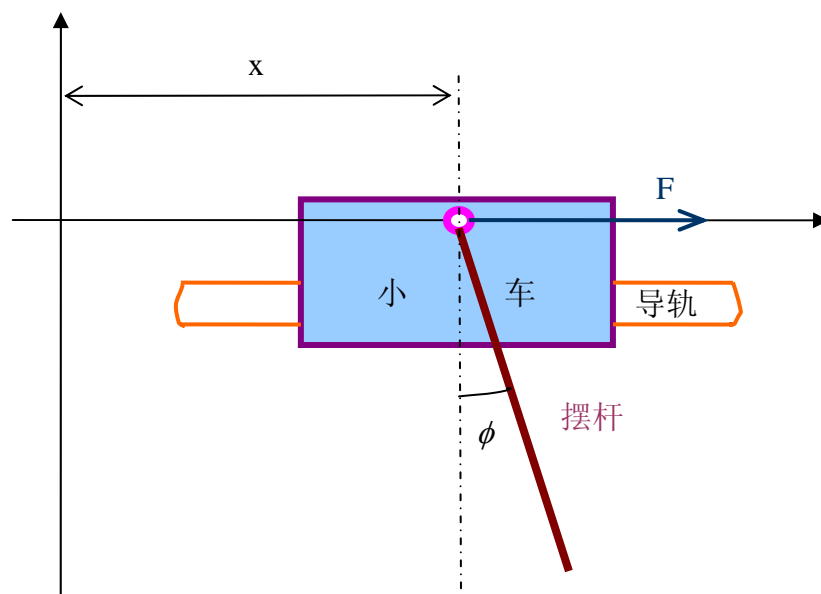


图 4-1 直线一级顺摆物理模型图

拉格朗日方程为：

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q, \dot{q}) \quad (4-1)$$

其中 L 为拉格朗日算子， q 为系统的广义坐标， T 为系统的动能， V 为系统的势能。

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = f_i \quad (4-2)$$

其中 $i=1,2,3,\dots,n$, f_i 为系统在第 i 个广义坐标上的外力, 在一级顺摆系统中, 系统的广义坐标有三个广义坐标, 分别为 x, ϕ 。

首先计算系统的动能:

$$T = T_M + T_m$$

其中 T_M, T_m 分别为小车的动能, 摆杆 1 的动能。

小车的动能:

$$T_M = \frac{1}{2} M \dot{x}^2$$

下面计算摆杆的动能:

$T_m = T_m' + T_m''$, 其中 T_m', T_m'' 分别为摆杆的平动动能和转动动能。

设以下变量:

xpend —— 摆杆质心横坐标;

ypend —— 摆杆质心纵坐标

有:

$$\begin{aligned} xpend &= x + l \sin \phi \\ ypend &= -l \cos \phi \end{aligned}$$

摆杆的动能为:

$$T_m' = \frac{1}{2} m \left(\left(\frac{d(xpend)}{dt} \right)^2 + \left(\frac{d(ypend)}{dt} \right)^2 \right)$$

$$T_m'' = \frac{1}{2} J_p \dot{\phi}^2 = \frac{1}{6} m l^2 \dot{\phi}^2$$

于是有系统的总动能:

$$T_m = T_m' + T_m'' = \frac{1}{2} m \left(\left(\frac{d(xpend)}{dt} \right)^2 + \left(\frac{d(ypend)}{dt} \right)^2 \right) + \frac{1}{6} m l^2 \dot{\phi}^2$$

系统的势能为:

$$V = V_m = m \times g \times ypend = -mgl \cos \phi$$

由于系统在 ϕ 广义坐标下只有摩擦力作用, 所以有:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\phi}} - \frac{\partial L}{\partial \phi} = b \dot{x}$$

对于直线一级倒立摆系统，系统状态变量为：

$$\{x, \phi, \dot{x}, \dot{\phi}\}$$

为求解状态方程：

$$\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases}$$

需要求解 $\ddot{\phi}$ ，

因此设

$$\ddot{\phi} = f(x, \phi, \dot{x}, \dot{\phi}, \ddot{x})$$

将在平衡位置附近进行泰勒级数展开，并线性化，可以得到：

$$\ddot{\phi} = k_{11}x + k_{12}\phi + k_{13}\dot{x} + k_{14}\dot{\phi} + k_{15}\ddot{x}$$

$$\text{其中 } k_{11} = \frac{\partial f}{\partial x} \Big|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0} \quad k_{12} = \frac{\partial f}{\partial \phi} \Big|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0}$$

$$k_{13} = \frac{\partial f}{\partial \dot{x}} \Big|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0} \quad k_{14} = \frac{\partial f}{\partial \dot{\phi}} \Big|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0}$$

$$k_{15} = \frac{\partial f}{\partial \ddot{x}} \Big|_{x=0, \phi=0, \dot{x}=0, \dot{\phi}=0, \ddot{x}=0}$$

在 Mathematica 中计算以上各式。

（计算文件请参见“\GLIP2001\建模文件\MathematicaFile”中的“L1DP.nb”文件。）

PRO 4-1 直线一级顺摆建模程序

```

xpend = x[t] + l*Sin[phi[t]];
ypend = -l*Cos[phi[t]];

tpend = 1/2*m*((D_t xpend)^2 + (D_t ypend)^2) + 1/6*m*l^2*(phi'[t])^2;
Simplify[tpend];
v = m*g*ypend;

lang = tpend - v;
Simplify[lang];

ldad = D_phi'[t] lang;
Simplify[ldad];
fa = D_t ldad - D_phi[t] lang;
Simplify[fa];

Solve[{fa == 0}, {phi''[t]}];
add = phi''[t] /. %;

k11 = D_x[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0
k12 = D_phi[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0
k13 = D_x'[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0
k14 = D_phi'[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0
k15 = D_x''[t] add /. x[t] -> 0 /. phi[t] -> 0 /. x'[t] -> 0 /. phi'[t] -> 0 /. x''[t] -> 0

g = 9.8;
l = 0.25;

Simplify[k12]
Simplify[k15]

```

运行程序得到:

$$k_{11} = 0$$

$$k_{12} = -\frac{3g}{4l}$$

$$k_{13} = 0$$

$$k_{14} = 0$$

$$k_{15} = -\frac{3}{4l}$$

设 $X = \{x, \dot{x}, \phi, \dot{\phi}\}$, 系统状态空间方程为:

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

$$y = CX + Du$$

则有：

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{3g}{4l} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{3}{4l} \end{bmatrix} u \quad (4-3)$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

通过转化可以得到：

$$\ddot{\phi} = -\frac{3g}{4l}\phi - \frac{3}{4l}\ddot{x}$$

摆杆角度和小车位置的传递函数为：

$$\frac{\Phi(s)}{X(s)} = \frac{-\frac{3}{4l}s^2}{s^2 + \frac{3g}{4l}}$$

摆杆角度和小车加速度的传递函数为：

$$\frac{\Phi(s)}{V(s)} = \frac{-\frac{3}{4l}}{s^2 + \frac{3g}{4l}}$$

4.1.2 实际系统模型

实际系统的物理参数见系统物理参数章节相关内容，把参数代入，可以得到系统的实际模型。

摆杆角度和小车位移的传递函数：

$$\frac{\Phi(s)}{X(s)} = \frac{-3s^2}{s^2 + 29.4} \quad (4-4)$$

摆杆角度和小车加速度之间的传递函数为：

$$\frac{\Phi(s)}{V(s)} = \frac{-3}{s^2 + 29.4} \quad (4-5)$$

因此以小车加速度作为输入的系统状态方程：

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -29.4 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ -3 \end{bmatrix} u$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u, \quad (4-6)$$

4.1.3 系统可控性分析

同 3.1.1.4 系统可控性分析，对系统进行可控性分析原理，有：

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -29.4 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -3 \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

代入上式，并在 MATLAB 中计算：

```
clear;
A=[ 0  1  0  0;
    0  0  0  0;
    0  0  0  1;
    0  0 -29.4 0];
B=[ 0  1  0  -3]';
C=[ 1  0  0  0;
    0  1  0  0];
D=[ 0 0 ]';
cona=[B A*B A^2*B A^3*B];
cona2=[C*B C*A*B C*A^2*B C*A^3*B D];
rank(cona)
rank(cona2)
```

或直接利用计算可控性矩阵的 *ctrb* 命令和计算可观性的矩阵 *obsv* 命令来计算。

```

Uc=ctrb(A,B);
Vo=obsv(A,C);

rank(Uc)
rank(Vo)

```

可以得到：

```

ans =
     4
ans =
     2

```

可以看出，和一级倒立摆相同，系统的状态完全可控性矩阵的秩等于系统的状态维数，系统的输出完全可控性矩阵的秩等于系统输出向量 y 的维数，所以系统可控，因此可以对系统进行控制器的设计，使系统稳定。

4.2 直线一级顺摆根轨迹分析

在得到系统传递函数后，我们对其进行根轨迹分析，和 3.1.3 直线一级倒立摆根轨迹控制实验相似，在 MATLAB 下新建一个文件，键入如下命令：

```

clear;
num=[-3 ];
den=[1 0 29.4];
rlocus(num,den)
z=roots(num);
p=roots(den);

```

得到结果如下，系统的零极点为：

```

z =
     0
     0
p =
    0 + 5.4222i
    0 - 5.4222i

```

系统的根轨迹：

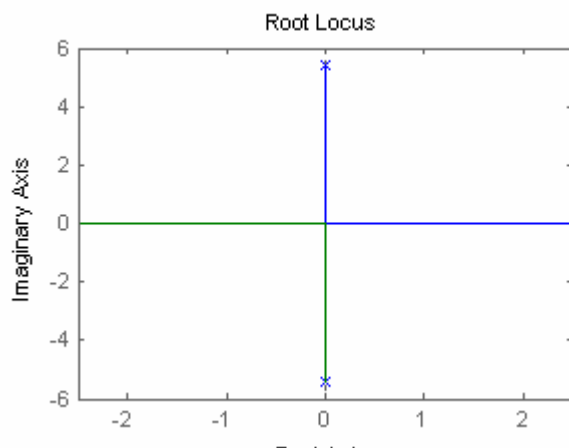


图 4-2 直线一级顺摆的根轨迹图

可以看出，系统的零点位于无穷远的地方，在虚轴上有两个极点。

4.3 直线一级顺摆频率响应分析

参考 3.1.4 直线一级倒立摆频率响应控制实验相关实验内容的原理，对直线一级顺摆进行频率响应分析，在 MATLAB 中键入以下命令：

```
clear;
num=[-3];
den=[1 0 29.4];

subplot(2,1,1)
bode(num,den)
subplot(2,1,2)
nyquist(num,den)
```

得到如下图所示的结果：

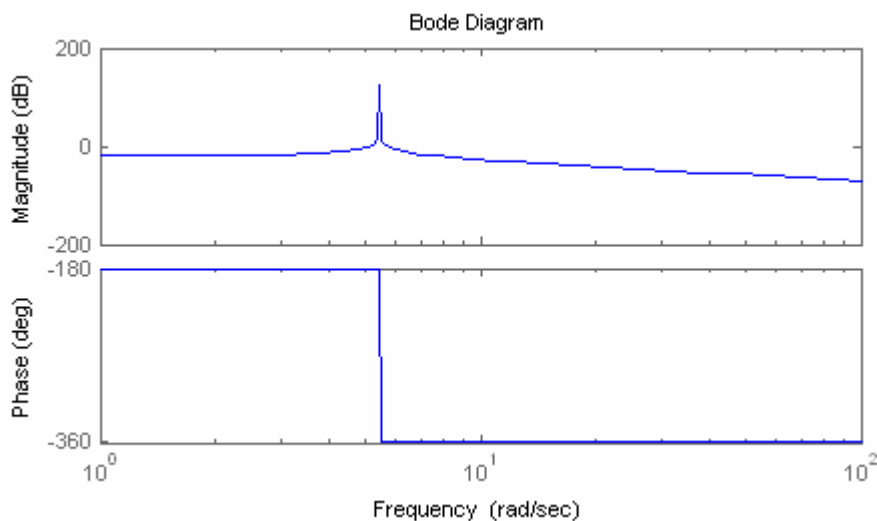


图 4-3 直线一级顺摆的 Bode 图

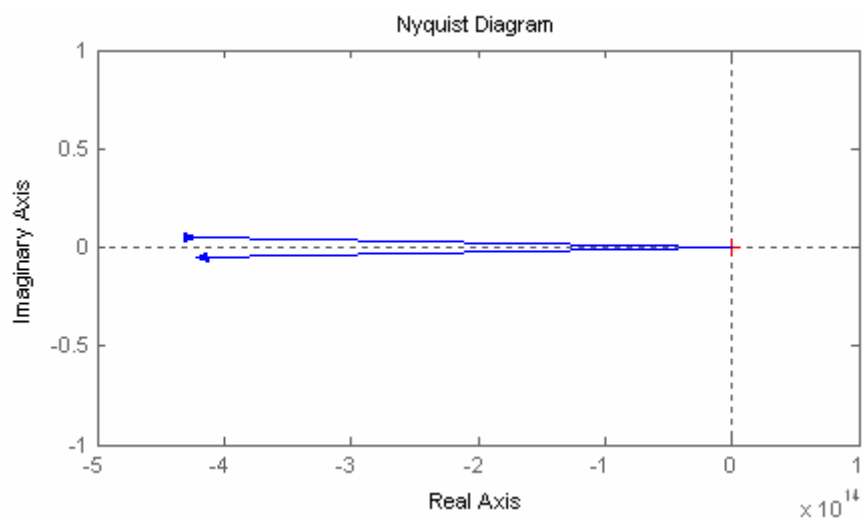


图 4-4 直线一级顺摆的 Nyquist 图

可以看出，系统的奈魁斯特曲线顺时针绕-1 点一圈，系统不稳定。

4.4 直线一级顺摆阶跃响应分析

同 3.1.2 系统阶跃响应分析原理相似，对直线一级顺摆进行阶跃响应分析。

```
clear;
num=[-3];
den=[1 0 29.4];

step(num,den)
```

运行得到以下结果：

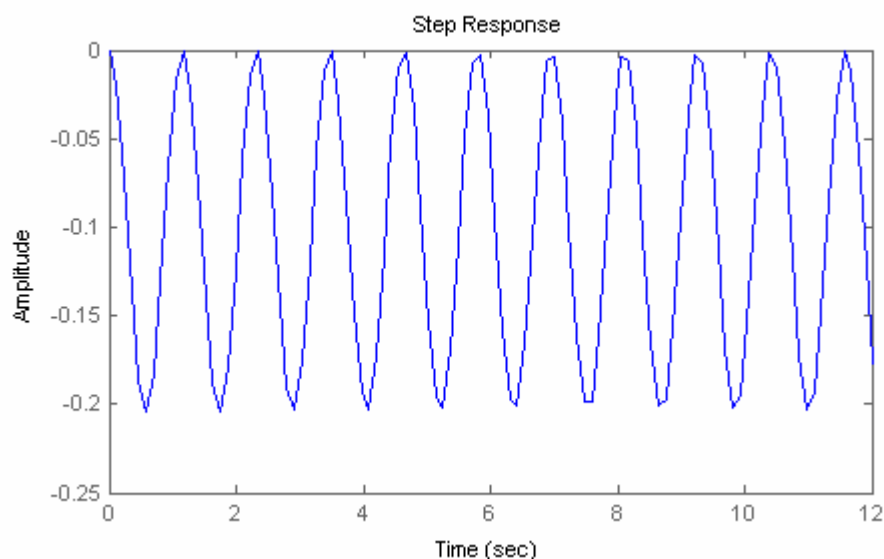


图 4-5 直线一级顺摆的阶跃响应结果

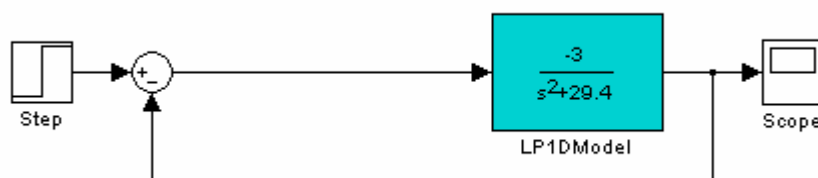
可以看出，系统在阶跃响应下会出现持续振荡，是一个典型的无阻尼的二阶系统。

4.5 直线一级顺摆的 PID 控制仿真与实验

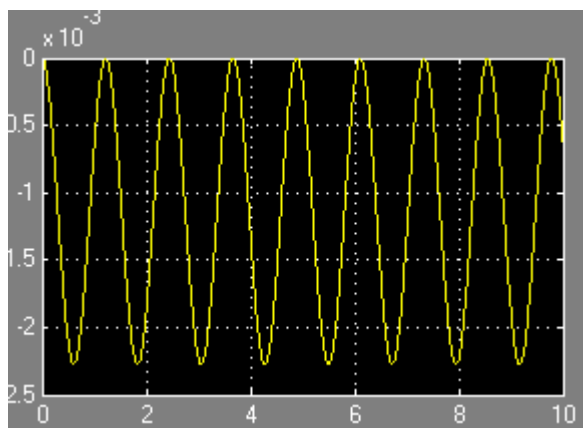
和直线一级倒立摆相似，直线一级顺摆也可以进行根轨迹校正实验、频率响应校正实验和状态空间极点配置实验，以上实验请参照直线一级倒立摆的相关实验和直线一级顺摆的 PID 控制实验进行，对于直线一级顺摆，我们只进行 PID 控制实验和 LQR 控制实验。

4.5.1 直线一级顺摆的 PID 控制分析与仿真

在 MTALAB Simulink 下建立直线一级顺摆的模型如下：



运行得到如下结果：



可以看出，和上面的仿真结果一样，下面添加 PID 控制器：
(进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage Pendulum Experiment\PID Experiments” 中的 “PID Control Simulink”)

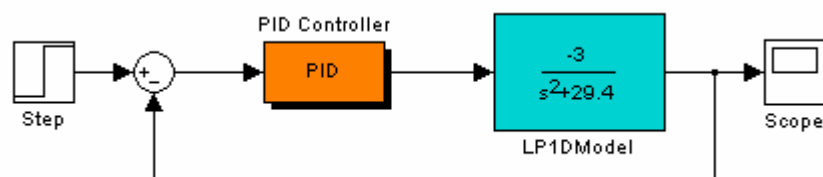


图 4-6 直线一级顺摆的 PID 控制仿真

双击 PID 模块设置 PID 参数：

Parameters	
Kp	-40
Ki	0
Kd	0

注意，由于系统模型中带有“-”号，PID 参数需要设置为负值。
运行仿真得到以下结果：

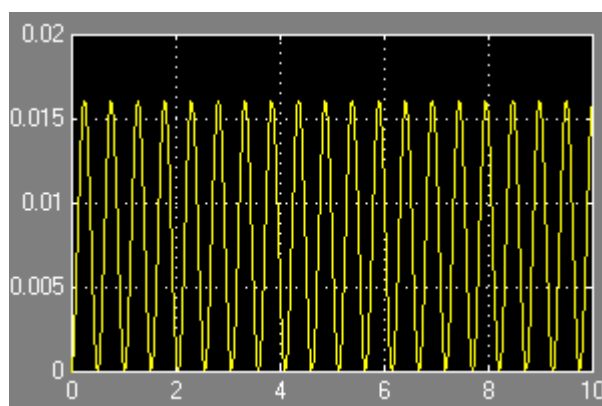
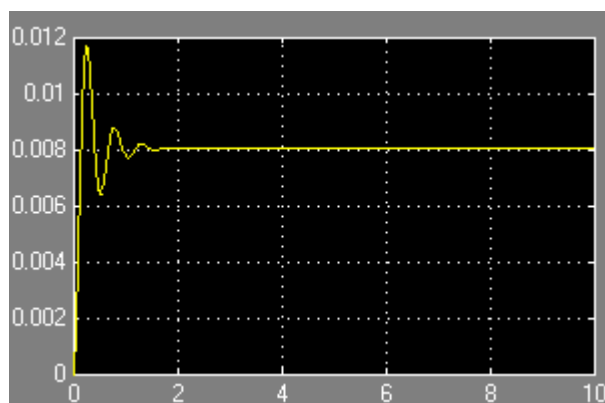


图 4-7 直线一级顺摆 P 控制结果

可以看出，在 P 控制器作用下，系统呈现等幅振荡，需要给系统增加微分控制，设置 PID 参数为：

Parameters	
Kp	-40
Ki	0
Kd	-2

运行仿真得到如下结果：



可以看出，系统振荡次数过多，稳定时间较长，因此增加微分参数：

Parameters	
Kp	-40
Ki	0
Kd	-4

给系统施加幅值为 0.01 的阶跃信号，得到下面的仿真结果：

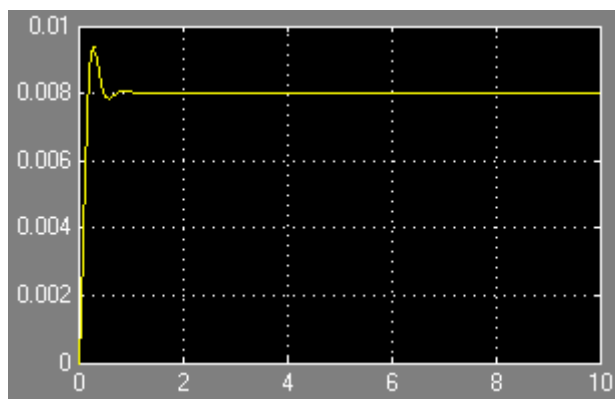
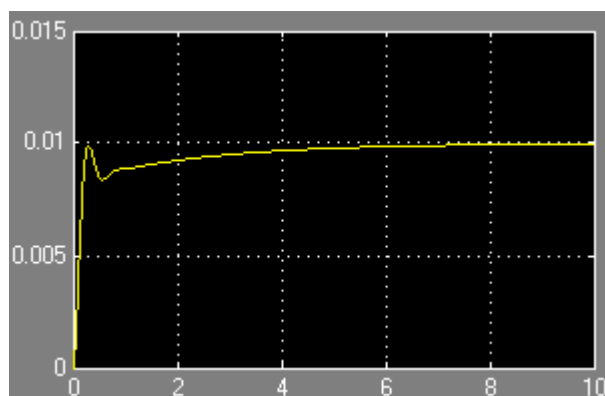


图 4-8 直线一级顺摆 PD 控制结果

可以看出，系统可以很快的稳定，但是存在一定的稳态误差，为消除稳态误差，增加积分控制：

Parameters	
Kp	-40
Ki	-20
Kd	-4



如果要求系统在 1 秒内达到平衡状态，超调不超过 20%。修改控制参数，观察仿真结果的变化，设：

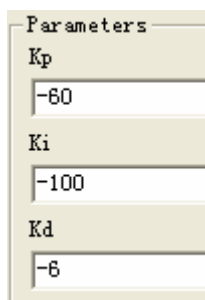


图 4-9 直线一级顺摆 PID 控制器

得到仿真结果如下：

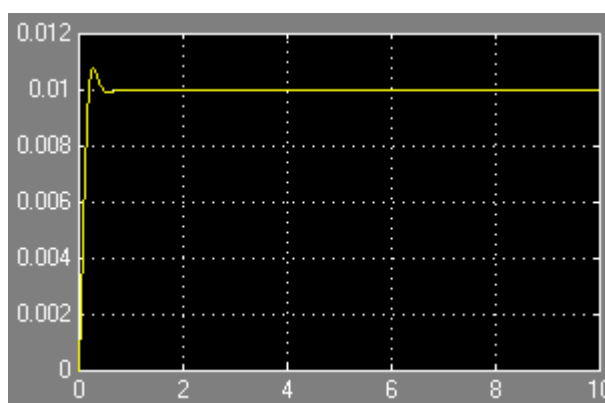


图 4-10 直线一级顺摆 PID 控制结果

从仿真结果来看，仿真结果基本上可以满足要求。

4.5.2 直线一级顺摆的 PID 实时控制实验

实时控制实验在 MATLAB Simulink 环境下进行，用户在实验前请仔细阅读使用手册。



注意

- 在进行 MATLAB 实时控制实验时，请用户检查倒立摆系统机械结构和电气接线有无危险因素存在，在保障实验安全的情况下进行实验。

实验步骤：

- 1) 在 MATLAB Simulink 中打开直线一级顺摆实时控制程序。

（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage Pendulum Experiment\PID Experiments” 中的 “PID Control Demo”）

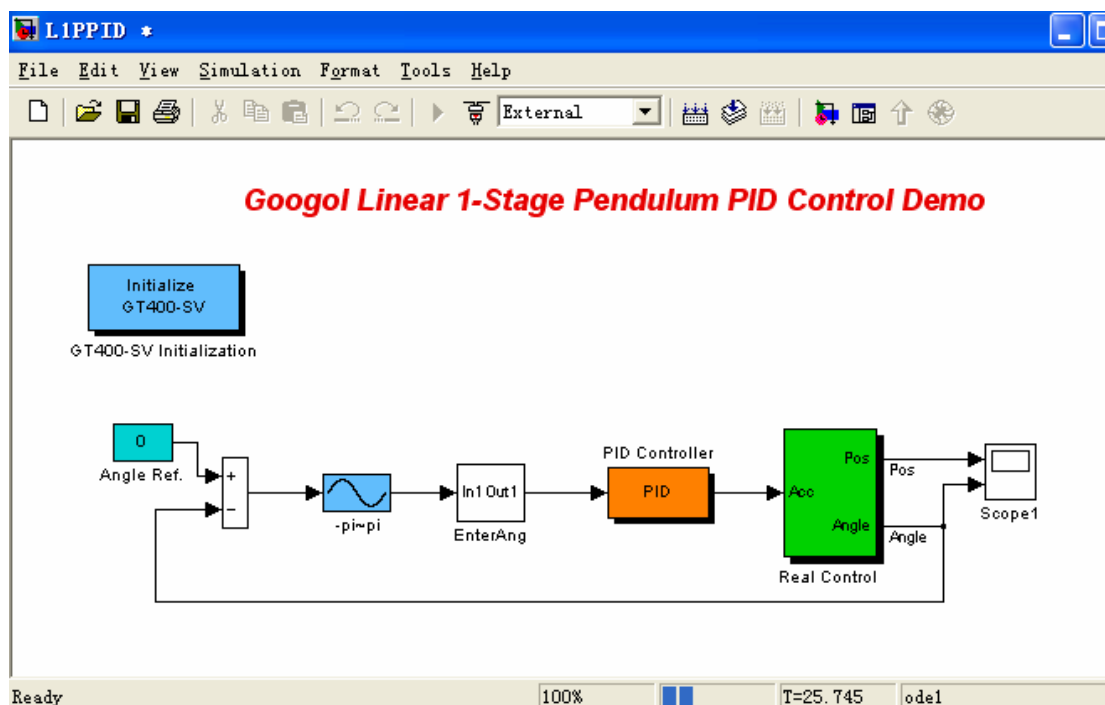
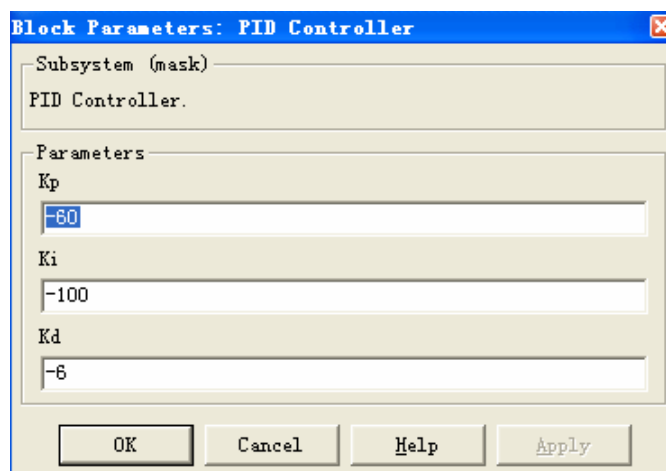


图 4-11 直线一级顺摆 PID 实时控制程序



2) 双击“PID Controller”模块打开 PID 参数设置界面：



把前面仿真得到的 PID 参数输入其中。

3) 点击“”编译程序，在 MATLAB 命令窗口中有编译提示信息，在编译成功后进行以下实验。

4) 打开电控箱电源，确认运行安全后进行下面的操作。

5) 点击“”连接程序，在连接成功后点击“”运行程序，在系统保持稳定的情况下给系统施加干扰。

得到以下实控结果：

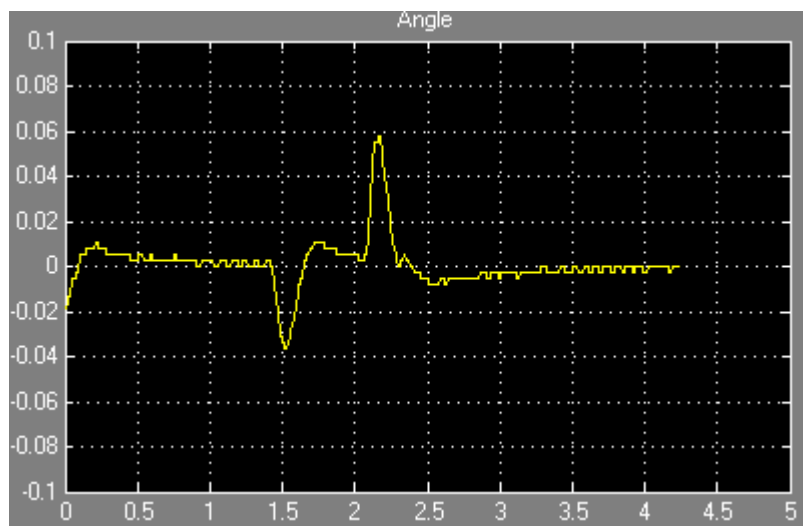


图 4-12 直线一级顺摆 PID 实时控制结果

可以看出，在施加干扰后，系统在 1 秒内基本上可以回复到新的平衡状态，超调也较小。

注意，由于 PID 控制是单输出控制，因此我们并没有控制小车的位置，一般情况下，小车都会往一个方向做慢速的运动，当运动到一边时，需要手动挡一下摆杆，避免小车运动到限位，程序停止运行。

- 6) 修改 PID 参数，再次进行实验，观察实验结果变化。
- 7) 记录实验结果，完成实验报告。



注意

- 在进行 MATLAB 实时控制实验时，请用户不要过大的改变控制器参数，过大的参数会使系统有失控的危险，请在保障实验安全的情况下进行实验。

4.6 直线一级顺摆的 LQR 控制仿真与实验

前面所述的 PID 控制只能控制摆杆的角度，并不能控制小车的位置，为了在控制摆杆角度的同时，同时也控制小车的位置，这需要一个单输入（小车加速度），多输出（小车位置和摆杆角度）的控制器，因为我们已经得到系统的状态方程，可以采用状态空间极点配置和线性二次最优控制等控制方法对其进行控制。下面我们以 LQR 控制方法对其进行控制，用户可以对其进行其它算法的设计和实验。

LQR 控制算法的原理请参照 3.1.7.1 线性二次最优控制 LQR 基本原理及分析相关内容或相关现代控制理论教材。

4.6.1 直线一级顺摆的 LQR 控制分析与仿真

前面我们已经得到以小车加速度作为输入的系统状态方程：

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -29.4 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ -3 \end{bmatrix} u$$

$$y = \begin{bmatrix} x \\ \phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

四个状态量 $x, \dot{x}, \phi, \dot{\phi}$ 分别代表小车位移、小车速度、摆杆角度和摆杆角速度，输出 $y = [x, \phi]'$ 包括小车位置和摆杆角度。

下面我们应用 LQR 控制算法对其进行计算：

设最佳最佳控制向量的矩阵 K：

$$u(t) = -K^* x(t)$$

$$K = \text{lqr}(A, B, Q, R)$$

在 MATLAB 中建立如下的 m 文件：

其中选取小车位置权重为 1000，摆杆角度权重为 500，一般选取 $R=1$ ，仿真时间为 5 秒，仿真步长为 0.005 秒。

直线一级顺摆的 LQR 计算仿真文件如下：

（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage Pendulum Experiment\LQR Control Experiments” 中的 “LQR Control M Files”）

PRO 4-2 直线一级顺摆 LQR 控制 MATLAB 计算程序

```

clear;
A=[ 0  1  0  0;
    0  0  0  0;
    0  0  0  1;
    0  0 -29.4 0];
B=[ 0  1  0  -3]';
C=[ 1  0  0  0;
    0  0  1  0];
D=[ 0 0 ]';

Q11=1000; Q33=500;
Q=[Q11 0 0 0;
    0 0 0 0;
    0 0 Q33 0;
    0 0 0 0];
R = 1;
K = lqr(A,B,Q,R)

Ac = [(A-B*K)]; Bc = [B]; Cc = [C]; Dc = [D];

T=0:0.005:5;
U=0.2*ones(size(T));
Cn=[1 0 0 0];
Nbar=rscale(A,B,Cn,0,K);
Bcn=[Nbar*B];
[Y,X]=lsim(Ac,Bcn,Cc,Dc,U,T);

plot(T,X(:,1),'-');hold on;
plot(T,X(:,2),'-');hold on;
plot(T,X(:,3),'-');hold on;
plot(T,X(:,4),'-')
legend('CartPos','CartSpd','PendAng','PendSpd')

```

运行程序，得到如下所示的结果：

得到反馈控制矩阵：

K =

31.6228 14.4046 -22.3514 0.1188

可以看出，在给定小车位置干扰后，系统在 3 秒内可以达到新的平衡，小车平衡到指定的位置，摆杆保持静止下垂的状态。

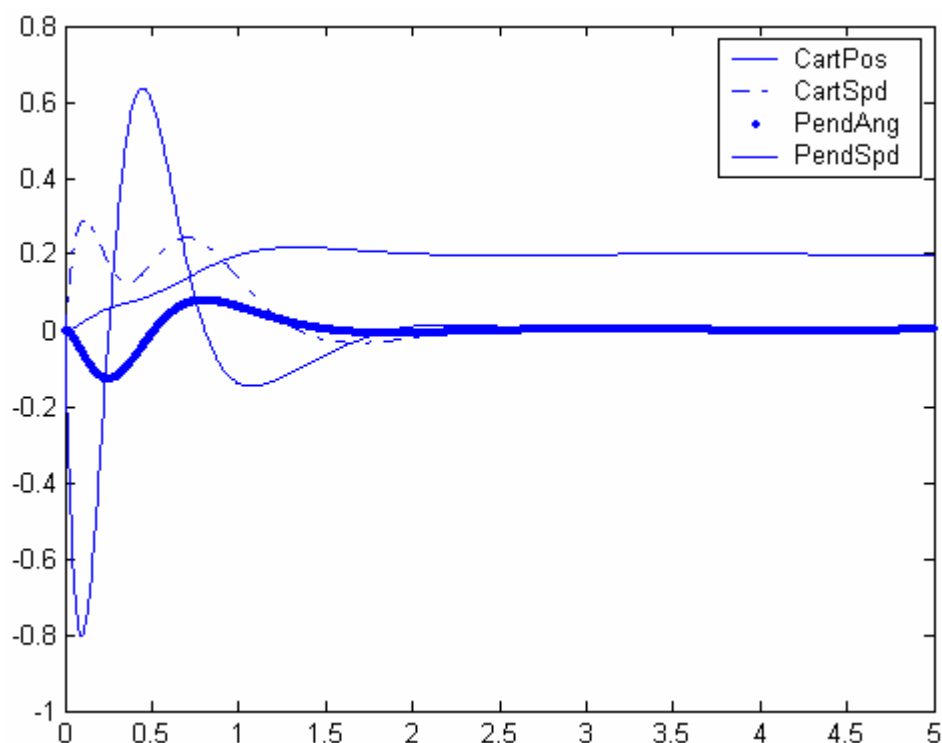


图 4-13 直线一级顺摆仿真结果图（给小车施加干扰）

同直线一级倒立摆相似，在 MATLAB Simulink 中建立直线一级顺摆的模型：
（进入 MATLAB Simulink 实时控制工具箱“Googol Education Products”打开“*Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage Pendulum Experiment\LQR Control Experiments*”中的“*LQR Control Simulink*”）

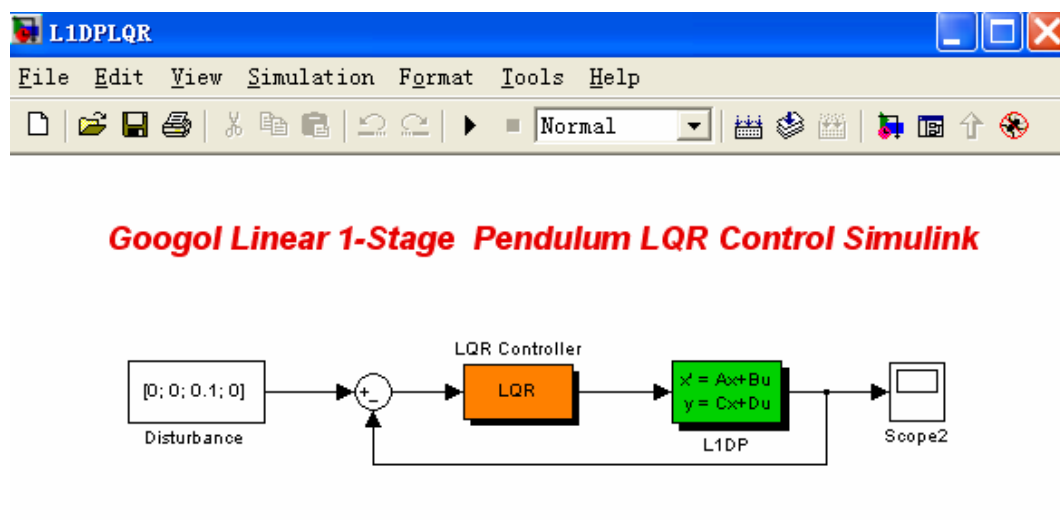
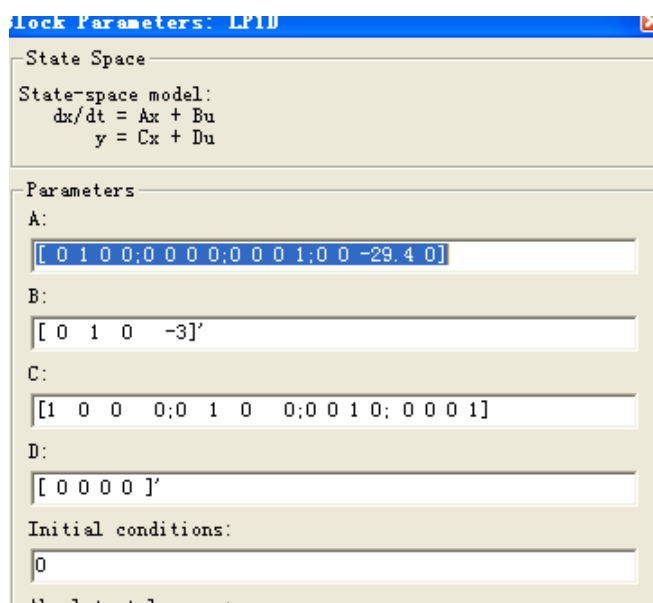


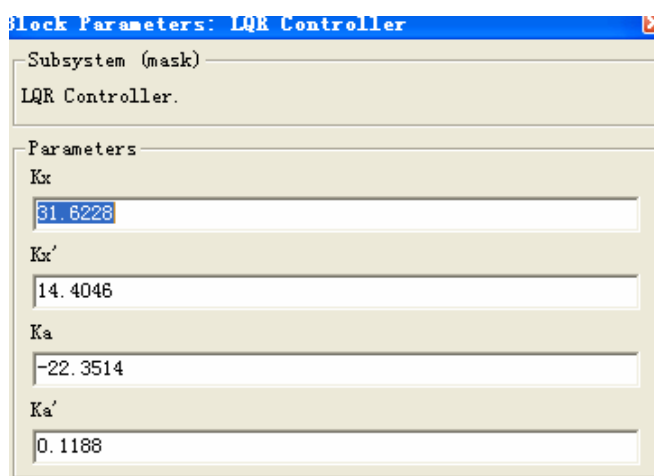
图 4-14 直线一级顺摆 Simulink 模型

双击“LP1D”模块打开直线一级顺摆的模型设置窗口如下：



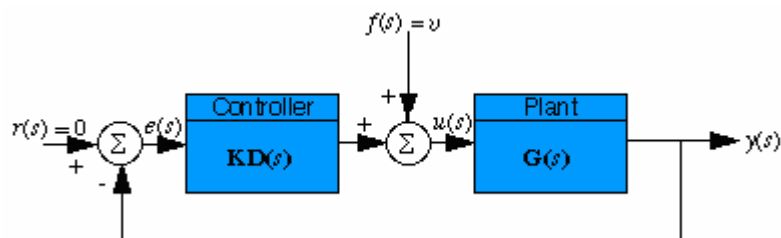
把矩阵 A,B,C,D 的值设置为实际系统模型的值。

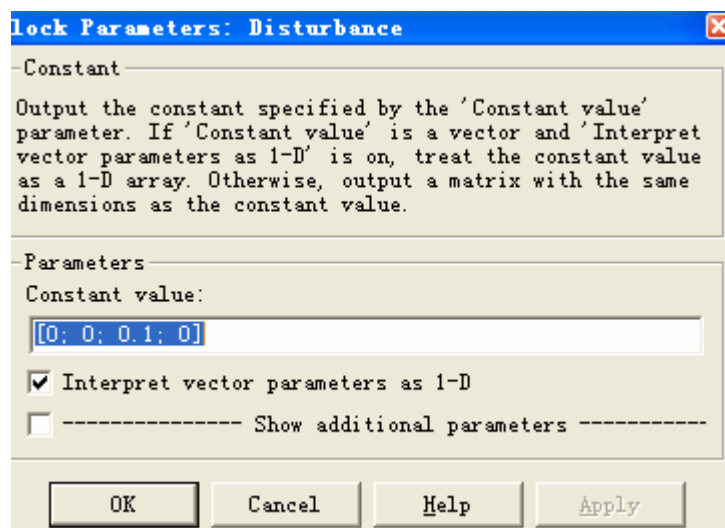
双击“LQR Controller”打开 LQR 控制参数的设置窗口：



把上面计算得到的反馈控制参数 K 依次输入到窗口中。

双击“Disturbance”模块设置干扰信号，设干扰信号为给摆杆一个幅值为 0.1 弧度的干扰信号，其原理如下图所示，给 $f(s)$ 施加一个干扰信号，控制系统在存在干扰的情况下保持新的平衡。





设置好各项参数后，点击“▶”运行仿真，得到如下的仿真结果：

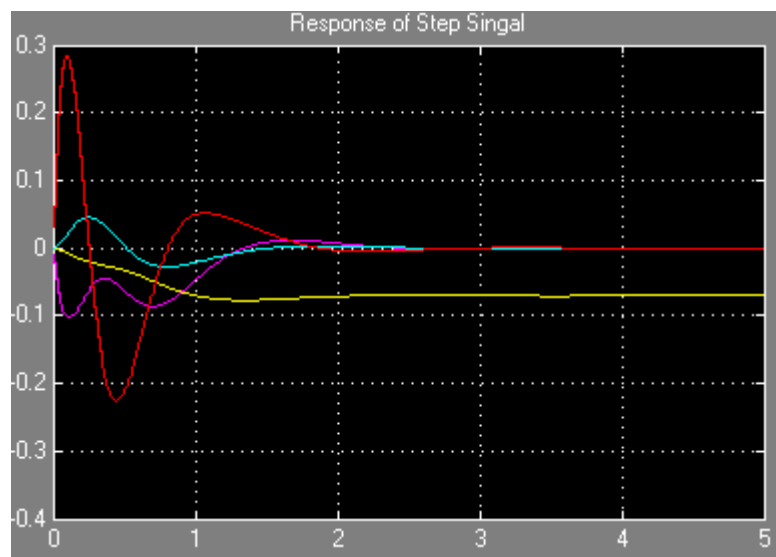


图 4-15 直线一级顺摆 LQR 控制仿真结果

用户可以修改权重矩阵 Q 的值，得到不同的反馈控制参数 K ，再输入到仿真模型中进行仿真，观察仿真结果的变化。

4.6.2 直线一级顺摆的 LQR 控制控制实验

实时控制实验在 MATLAB Simulink 环境下进行，用户在实验前请仔细阅读使用手册。



注意

- 在进行 MATLAB 实时控制实验时，请用户检查倒立摆系统机械结构和电气接线有无危险因素存在，在保障实验安全的情况下进行实验。

操作步骤：

1) 打开直线一级顺摆 LQR 实时控制模块:

(进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage Pendulum Experiment\LQR Control Experiments” 中的 “LQR Control Demo”)

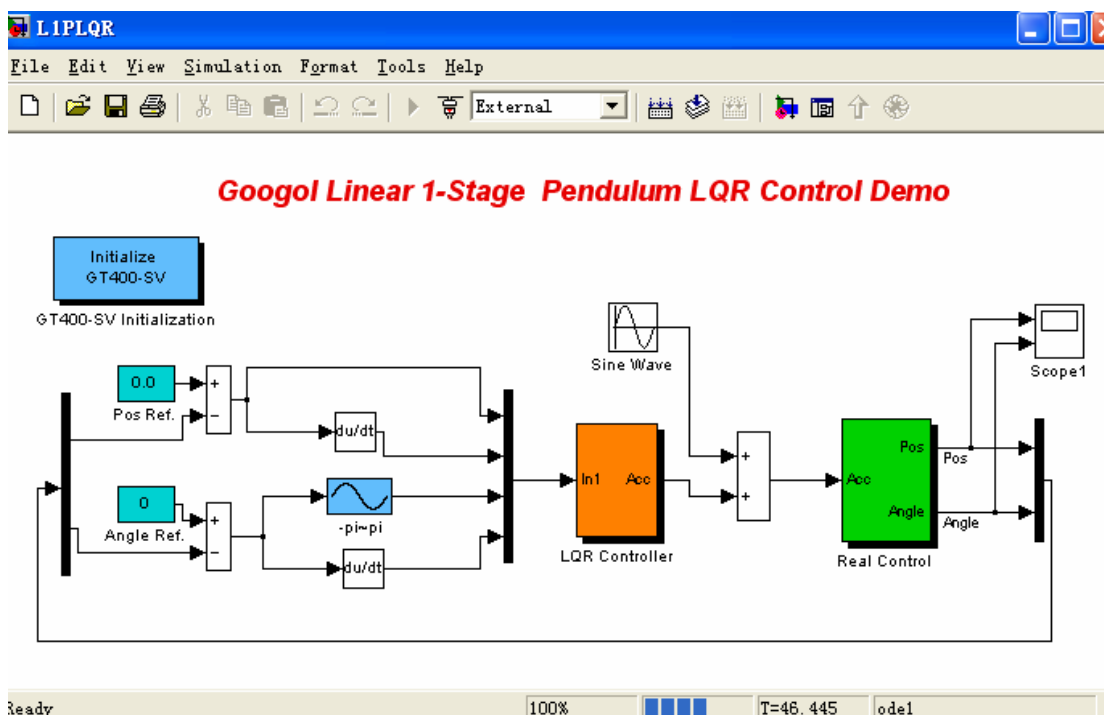
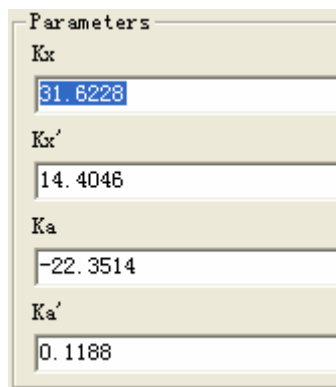
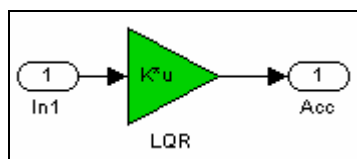


图 4-16 直线一级顺摆 LQR 实时控制程序

其中 “LQR Controller” 为 LQR 控制器模块, “Real Control” 为实时控制模块, 双击 “LQR Controller” 模块打开 LQR 控制器参数设置窗口如下:

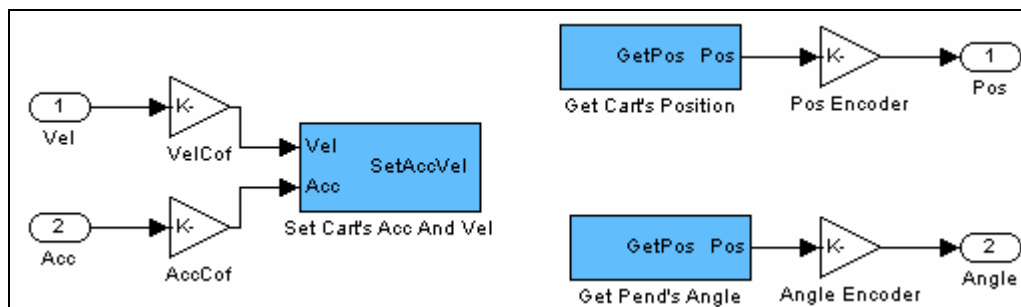





在 “LQR Controller” 模块上点击鼠标右键, 选择 “Look under mask”, 打开模型如下:



双击 “Real Control” 模块, 打开实时控制模块如下图:

双击“Pendulum”模块打开其内部结构:



- 2) 点击编译程序，编译成功后点击“”连接，再点击“”运行程序，程序运行后，小车应该停在中间位置，摆杆保持静止下垂。
- 3) 实验运行结果如下图所示：

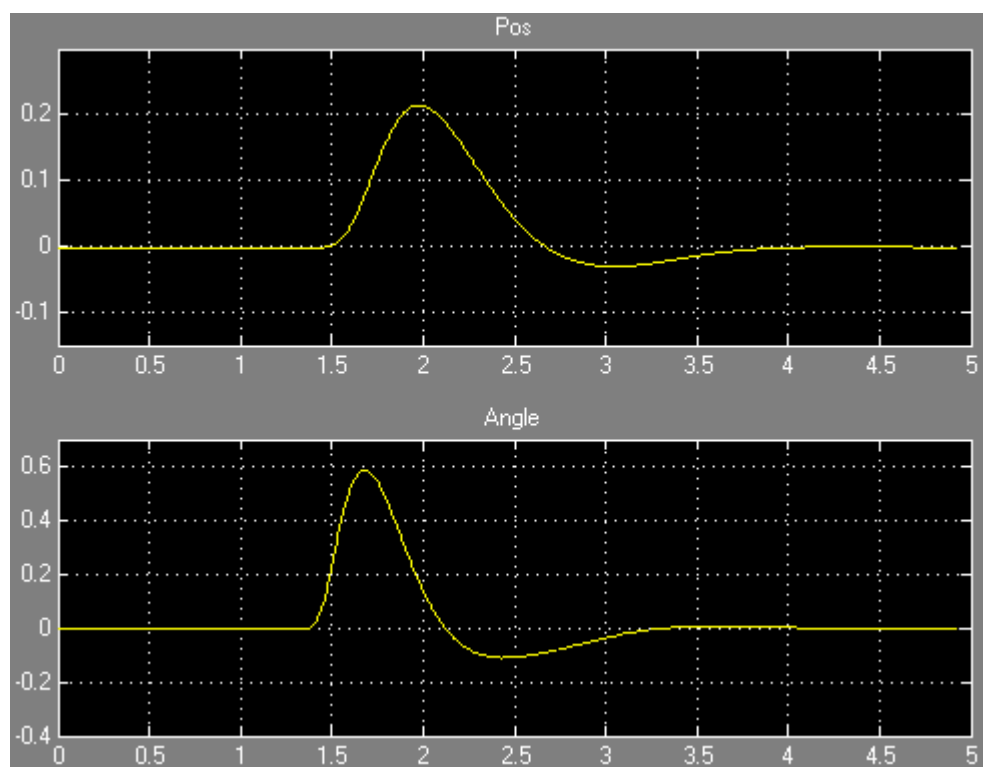


图 4-17 直线一级顺摆 LQR 控制结果

其中图片上半部分为小车的位置曲线，下半部分为摆杆角度的变化曲线，从图中可以看出，在给定外界干扰后，小车可以在 2 秒内回到平衡位置。达到了较好的控制效果。

- 4) 改变 Q 矩阵的值 Q_{11} 和 Q_{33} ，再把仿真得到的 LQR 控制参数输入实时控制程序，运行实时控制程序，观察控制效果的变化。
- 5) 记录实验结果，分析实验数据，完成实验报告。



注意

- 在进行 MATLAB 实时控制实验时，请用户不要过大的改变控制器参数，过大的参数会使系统有失控的危险，请在保障实验安全的情况下进行实验。

第5章 直线一级倒立摆自动摆起控制实验

对于直线一级倒立摆，其初始状态为静止下垂状态，为使其转化到竖直向上的状态，需要给摆杆施加力的作用。上面的实验，我们都是采用手动的方法将摆杆提起，下面我们采用自动摆起的方法对其进行控制。

5.1 摆起的能量控制策略

单个不受约束的倒立摆系统的能量为：

$$E = \frac{1}{2} J \dot{\phi}^2 + mgl(\cos \phi - 1)$$

有：

$$\frac{dE}{dt} = J \dot{\phi} \ddot{\phi} - mgl \dot{\phi} \sin \phi = -mul \dot{\phi} \cos \phi$$

其中 u ——为水平向右的控制量。

应用李亚普诺夫方法，令：

$$V = \frac{1}{2} (E - E_{ref})^2$$

则：

$$\frac{dV}{dt} = -(E - E_{ref}) mul \dot{\phi} \cos \phi$$

因此，令： $u = k(E - E_{ref}) \dot{\phi} \cos \phi$

注意当 $\dot{\phi}=0$ 或 $\cos \phi=0$ 时， $u=0$ 。

另外，由于实际物理系统的限制，控制量不能太大，因此采用：

$$v = \begin{cases} \text{sign}[(E - E_{ref}) \dot{\phi} \cos \phi] \cdot ng & |\theta| \leq \frac{\pi}{2} \\ 0 & \text{其他} \end{cases}$$

其中， $\text{sign}()$ 为取符号函数， $n = v_{\max} / g$ 为常数。

5.2 直线一级倒立摆摆起控制实验

实时控制实验在 MATLAB Simulink 环境下进行，用户在实验前请仔细阅读使用手册。



注意

- 在进行 MATLAB 实时控制实验时，请用户检查倒立摆系统机械结构和电气接线有无危险因素存在，在保障实验安全的情况下进行实验。

实验步骤:

- 1) 在 MATLAB Simulink 中打开直线一级倒立摆起摆控制程序:

(进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 1-Stage IP Swing-Up Control” 中的 “Swing-Up Control Demo”)

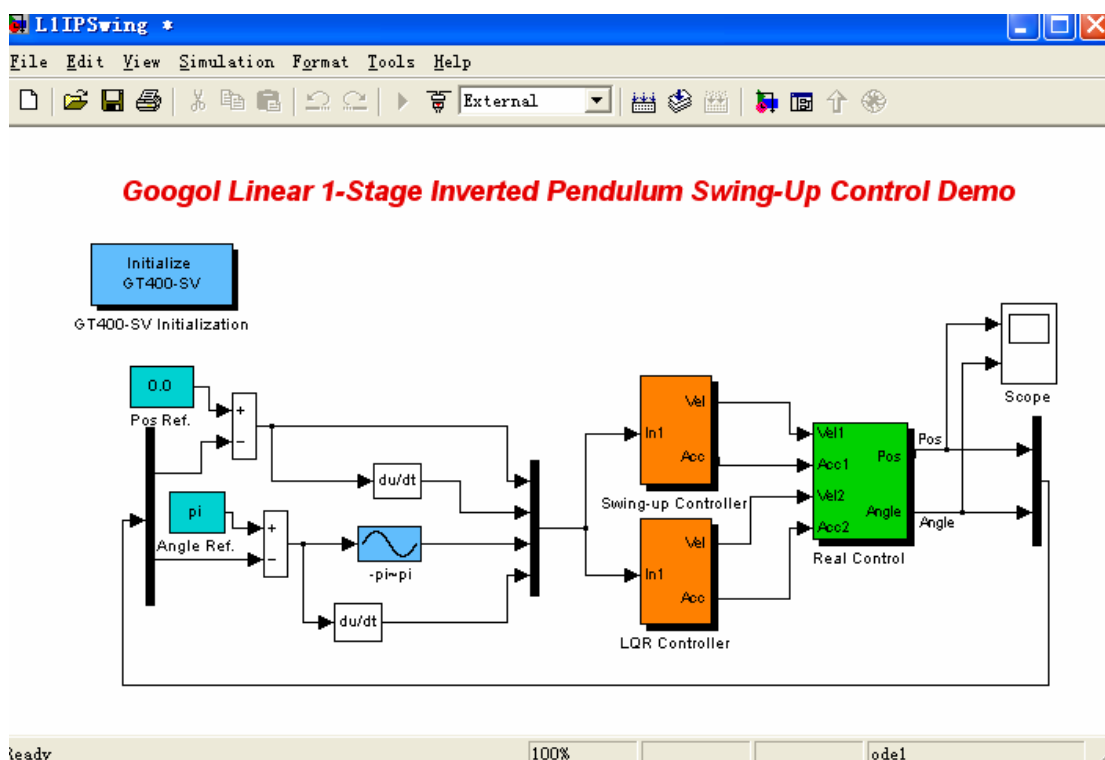
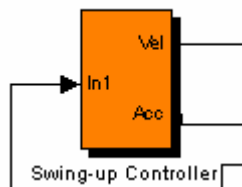
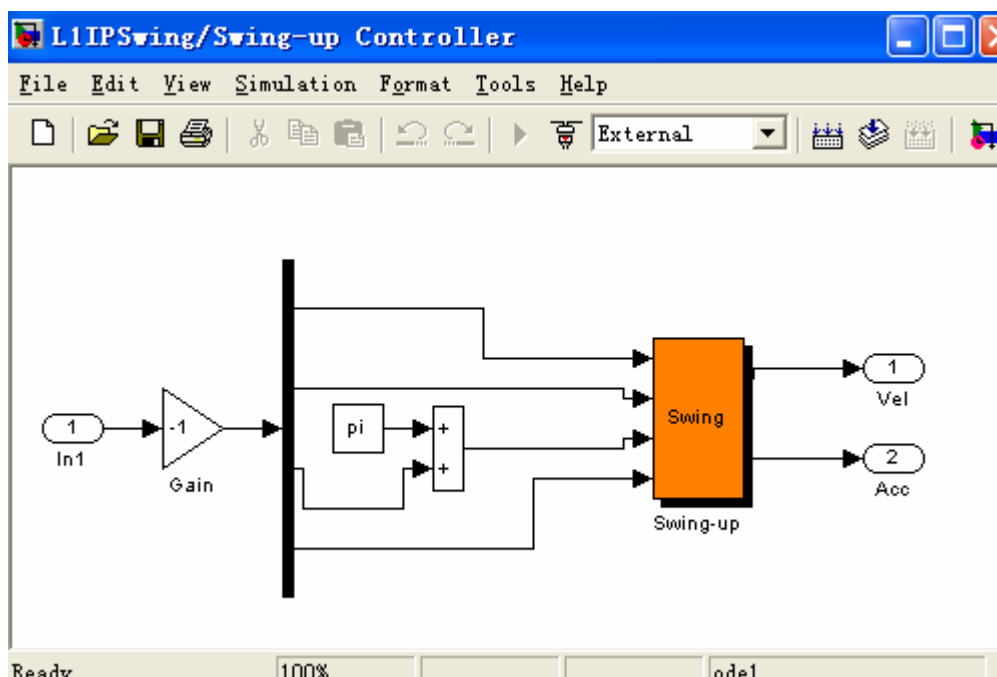


图 5-1 直线一级倒立摆摆起实时控制程序

- 2) 其中 “Swing-up Controller”为起摆控制模块。

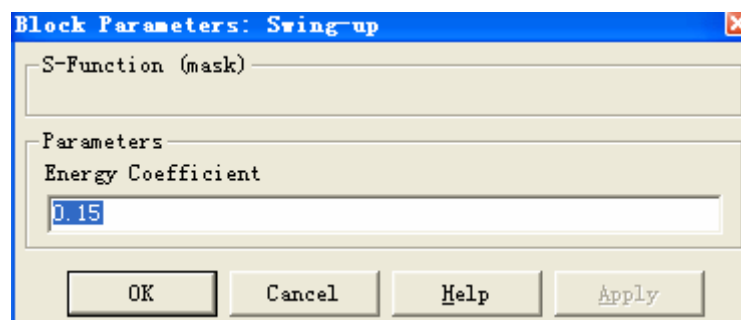





双击 “Swing-up Controller”模块打开模块如下图所示:



其中“Swing-up”模块为编写好的起摆函数。

双击“Swing-up”模块打开能量系数设置窗口：



- 3) 在确认系数正确后，点击“”编译程序。
- 4) 编译成功后，点击“”连接程序。
- 5) 点击“”运行程序，得到如下的实验结果：

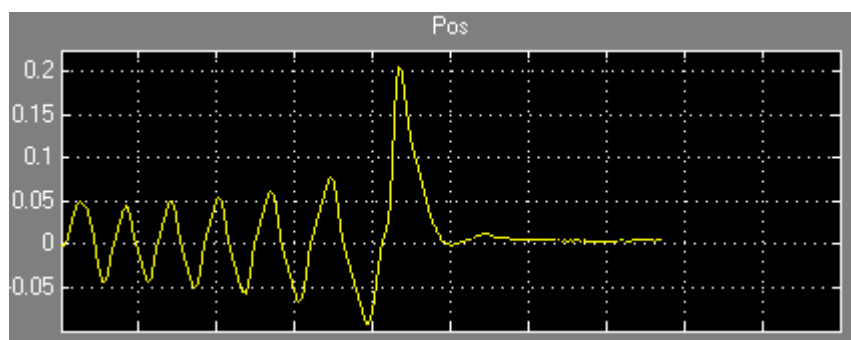


图 5-2 直线一级倒立摆起摆控制结果（小车位置）

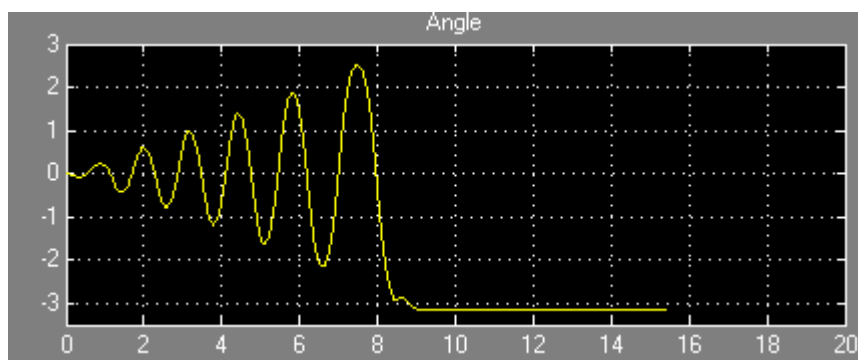


图 5-3 直线一级倒立摆起摆控制结果（摆杆角度）

可以看出，摆杆的角度的最大值慢慢加大，当进入一定的范围之内，开始进入稳摆控制，此时，小车平衡在初始位置（0）附近，摆杆平衡在竖直向上的位置（相对于初始位置，即静止下垂的角度为 π ）。

6) 记录实验结果，分析实验数据，完成实验报告。

5.3 其它算法实验实现方法

在 MATLAB 实时控制软件中，用户可以很方便的添加自己开发的各种算法，对于经典控制理论的单输入单输出算法，可以在 3.1.5 直线一级倒立摆 *PID* 控制实验中所描述的模型中，修改 *PID* 控制器为用户开发的控制器：

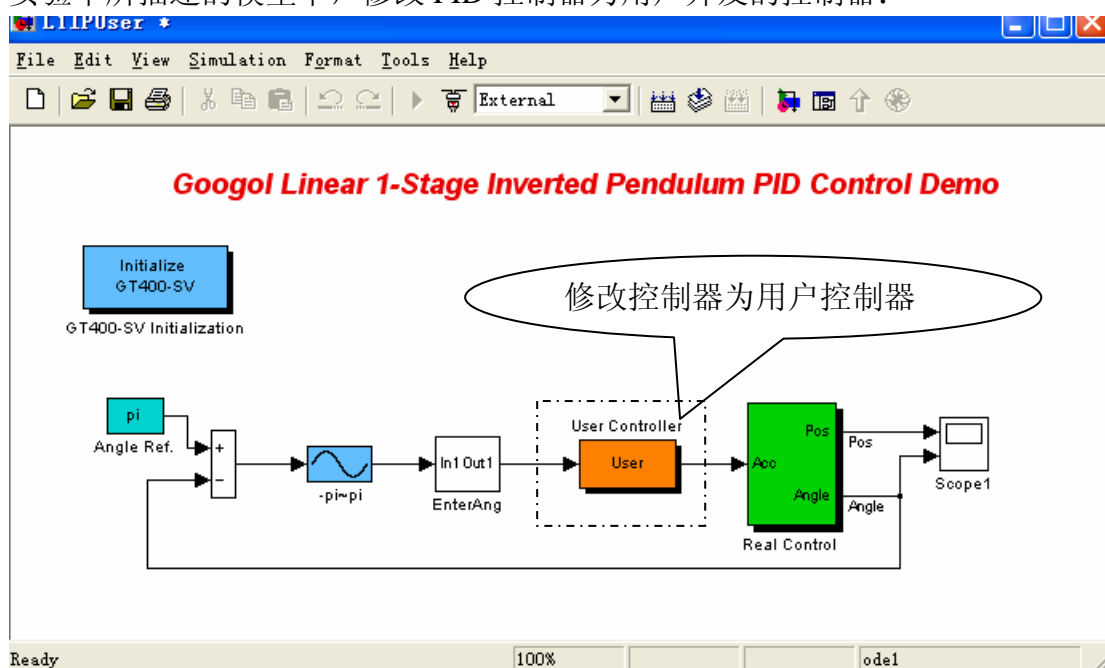


图 5-4 用户控制器添加方法示意图（单输入单输出）

摆杆角度和小车位置等信号采集模块和其它当量转换模块等，在没有特殊的情况下，并不需要改动。

应用现代控制理论等单输入（小车加速度）多输出（摆杆角度和小车位置）的控制算法时，用户可以在如 3.1.7 线性二次最优控制 *LQR* 控制实验中修改 *LQR* 控制器为用户开发的控制器：

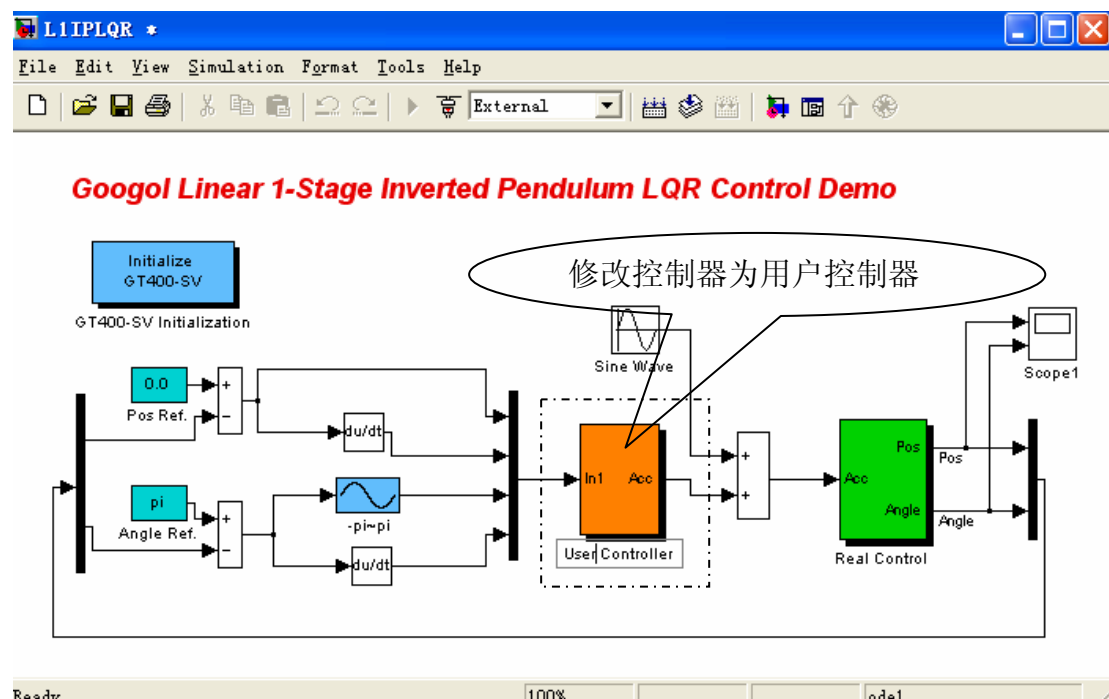


图 5-5 用户控制器添加方法示意图（单输入多输出）

在修改控制器后，重新编译程序，编译成功后连接程序，在保证实验安全的情况下运行程序。

用户也可以添加自己开发的起摆控制算法，如下图所示：

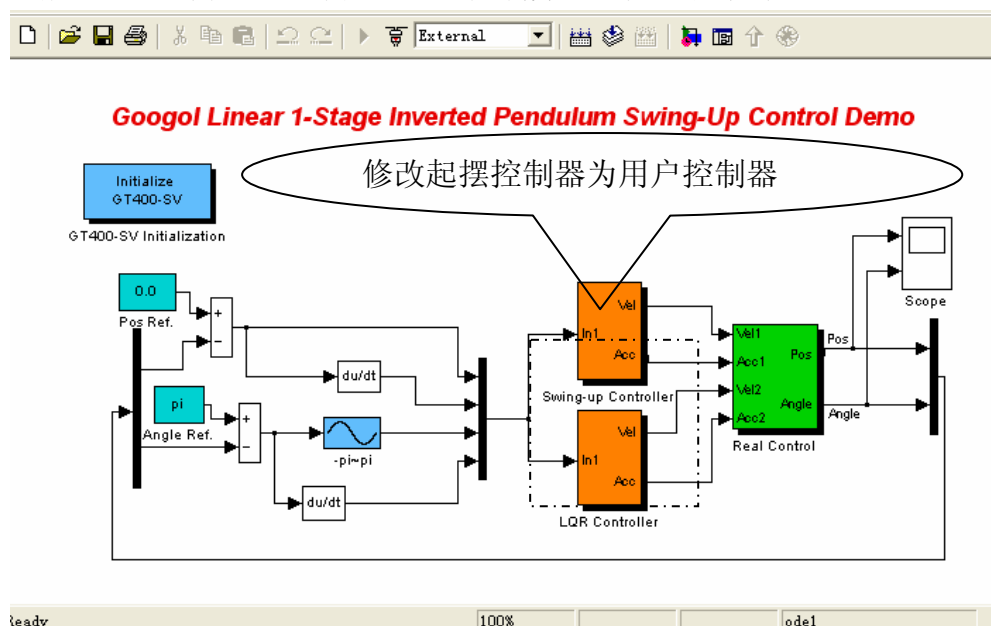


图 5-6 起摆控制算法修改示意图



注意

- 在进行用户自己设计的控制器进行控制时，需要特别注意实验安全，用户必须在得到较好的仿真结果后，才能进行实际控制实验，如果控制不成功，请立即关闭电控箱电源。

第6章 直线两级倒立摆

直线两级倒立摆由直线运动模块和两级倒立摆组件组成。

6.1 系统物理模型

为简化系统,我们在建模时忽略了空气阻力和各种摩擦,并认为摆杆为刚体。二级倒立摆的组成如图 6-1 所示:

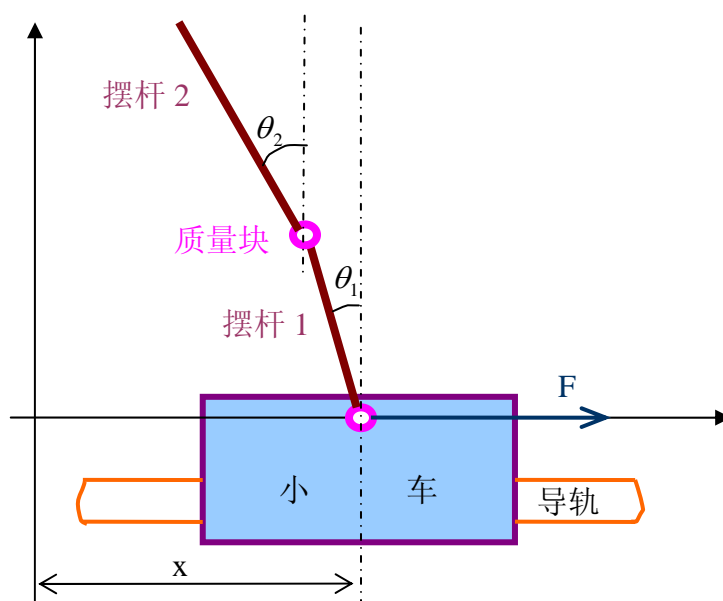


图 6-1 直线两级倒立摆物理模型

倒立摆参数定义如下:

M 小车质量

m_1 摆杆 1 的质量

m_2 摆杆 2 的质量

m_3 质量块的质量

l_1 摆杆 1 中心到转动中心的距离

l_2 摆杆 2 中心到转动中心的距离

θ_1 摆杆 1 与竖直方向的夹角

θ_2 摆杆 2 与竖直方向的夹角

F 作用在系统上的外力

利用拉格朗日方程推导运动学方程：

拉格朗日方程为：

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q, \dot{q})$$

其中 L 为拉格朗日算子， q 为系统的广义坐标， T 为系统的动能， V 为系统的势能。

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = f_i$$

其中 $i=1,2,3,\dots,n$ ， f_i 为系统在第 i 个广义坐标上的外力，在二级倒立摆系统

中，系统的广义坐标有三个广义坐标，分别为 x, θ_1, θ_2 。

首先计算系统的动能：

$$T = T_M + T_{m1} + T_{m2} + T_{m3}$$

其中 $T_M, T_{m1}, T_{m2}, T_{m3}$ 分别为小车的动能，摆杆 1 的动能，摆杆 2 的动能和质量块的动能。

小车的动能：

$$T_M = \frac{1}{2} M \dot{x}^2$$

$T_{m1} = T_{m1}' + T_{m1}''$ 其中 T_{m1}', T_{m1}'' 分别为摆杆 1 的平动动能和转动动能。

$T_{m2} = T_{m2}' + T_{m2}''$ 其中 T_{m2}', T_{m2}'' 分别为摆杆 2 的平动动能和转动动能。

对于系统，设以下变量：

xpend1 摆杆 1 质心横坐标；

yangle1 摆杆 1 质心纵坐标；

xpend2 摆杆 2 质心横坐标；

yangle2 摆杆 2 质心纵坐标；

xmass 质量块质心横坐标；

ymass 质量块质心纵坐标；

又有：

$$\begin{cases} x_{pend1} = x - l_1 \sin \theta_1 \\ y_{pend1} = l_1 \cos \theta_1 \\ x_{pend2} = x - 2l_1 \sin \theta_1 - l_2 \sin(\theta_2) \\ y_{pend2} = 2l_1 \cos \theta_1 + l_2 \cos \theta_2 \\ x_{mass} = x - 2l_1 \sin \theta_1 \\ y_{mass} = 2l_1 \cos \theta_1 \end{cases}$$

则有:

$$\begin{aligned} T_{m1}' &= \frac{1}{2} m_1 \left(\left(\frac{d(x_{pend1})}{dt} \right)^2 + \left(\frac{d(y_{pend1})}{dt} \right)^2 \right) \\ T_{m1}'' &= \frac{1}{2} J_{p1} \dot{\theta}_1^2 = \frac{1}{6} m_1 l_1^2 \dot{\theta}_1^2 \end{aligned}$$

同理:

$$\begin{aligned} T_{m2}' &= \frac{1}{2} m_2 \left(\left(\frac{d(x_{pend2})}{dt} \right)^2 + \left(\frac{d(y_{pend2})}{dt} \right)^2 \right) \\ T_{m2}'' &= \frac{1}{2} J_{p2} \dot{\theta}_2^2 = \frac{1}{6} m_2 l_2^2 \dot{\theta}_2^2 \\ T_{m3} &= \frac{1}{2} m_3 \left(\left(\frac{d(x_{mass})}{dt} \right)^2 + \left(\frac{d(y_{mass})}{dt} \right)^2 \right) \end{aligned}$$

于是有系统的总动能:

$$\begin{aligned} T &= T_M + T_{m1} + T_{m2} + T_{m3} \\ &= \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m_1 \left(\left(\frac{d(x_{pend1})}{dt} \right)^2 + \left(\frac{d(y_{pend1})}{dt} \right)^2 \right) + \frac{1}{6} m_1 l_1^2 \dot{\theta}_1^2 \\ &\quad + \frac{1}{2} m_2 \left(\left(\frac{d(x_{pend2})}{dt} \right)^2 + \left(\frac{d(y_{pend2})}{dt} \right)^2 \right) + \frac{1}{6} m_2 l_2^2 \dot{\theta}_2^2 \\ &\quad + \frac{1}{2} m_3 \left(\left(\frac{d(x_{mass})}{dt} \right)^2 + \left(\frac{d(y_{mass})}{dt} \right)^2 \right) \end{aligned}$$

系统的势能为:

$$\begin{aligned} V &= V_{m1} + V_{m2} + V_{m3} = m_1 y_{pend1} + m_2 y_{pend2} + m_3 y_{mass} \\ &= m_1 l_1 \cos \theta_1 + m_2 (2l_1 \cos \theta_1 + l_2 \cos \theta_2) + 2m_3 l_1 \cos \theta_1 \end{aligned}$$

由于系统在 θ_1, θ_2 广义坐标下没有外力作用, 所以有:

$$\begin{cases} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} = 0 \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} = 0 \end{cases}$$

对于二级倒立摆系统，系统状态变量为：

$$\{x, \theta_1, \theta_2, \dot{x}, \dot{\theta}_1, \dot{\theta}_2\}$$

为求解状态方程：

$$\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases}$$

需要求解 $\ddot{\theta}_1$ 和 $\ddot{\theta}_2$

因此设：

$$\begin{cases} \ddot{\theta}_1 = f_1(x, \theta_1, \theta_2, \dot{x}, \dot{\theta}_1, \dot{\theta}_2, \ddot{x}) \\ \ddot{\theta}_2 = f_2(x, \theta_1, \theta_2, \dot{x}, \dot{\theta}_1, \dot{\theta}_2, \ddot{x}) \end{cases}$$

将在平衡位置附近进行泰勒级数展开，并线性化，可以得到：

$$\begin{cases} \ddot{\theta}_1 = k_{11}x + k_{12}\theta_1 + k_{13}\theta_2 + k_{14}\dot{x} + k_{15}\dot{\theta}_1 + k_{16}\dot{\theta}_2 + k_{17}\ddot{x} \\ \ddot{\theta}_2 = k_{21}x + k_{22}\theta_1 + k_{23}\theta_2 + k_{24}\dot{x} + k_{25}\dot{\theta}_1 + k_{26}\dot{\theta}_2 + k_{27}\ddot{x} \end{cases}$$

其中

$$\begin{aligned} k_{11} &= \frac{\partial f_1}{\partial x} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} & k_{12} &= \frac{\partial f_1}{\partial \theta_1} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} \\ k_{13} &= \frac{\partial f_1}{\partial \theta_2} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} & k_{14} &= \frac{\partial f_1}{\partial \dot{x}} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} \\ k_{15} &= \frac{\partial f_1}{\partial \dot{\theta}_1} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} & k_{16} &= \frac{\partial f_1}{\partial \dot{\theta}_2} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} \\ k_{17} &= \frac{\partial f_1}{\partial \ddot{x}} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} & k_{21} &= \frac{\partial f_2}{\partial x} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} \\ k_{22} &= \frac{\partial f_2}{\partial \theta_1} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} & k_{23} &= \frac{\partial f_2}{\partial \theta_2} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} \end{aligned}$$

$$k_{24} = \frac{\partial f_2}{\partial \dot{x}} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} \quad k_{25} = \frac{\partial f_2}{\partial \dot{\theta}_1} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0}$$

$$k_{26} = \frac{\partial f_2}{\partial \dot{\theta}_2} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0} \quad k_{27} = \frac{\partial f_2}{\partial \ddot{x}} \Big|_{x=0, \theta_1=0, \theta_2=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \ddot{x}=0}$$

在 Mathematica 中计算以上各式:

(计算文件请参见“\GLIP2002\建模文件\MathematicaFile”中的“L2DIP.nb”文件。)

PRO 6-1 直线两级倒立摆建模 Mathematica 程序

```

TM = 1/2 Mcar * x'[t]^2 ;

xpend1 = x[t] - l1 * Sin[theta1[t]] ;
ypend1 = l1 * Cos[theta1[t]] ;
tpend1 = 1/2 * m1 * ((D_t xpend1)^2 + (D_t ypend1)^2) + 1/2 * (1/3 * m1 * l1^2) * theta1'[t]^2 ;

xpend2 = x[t] - 2 * l1 * Sin[theta1[t]] - l2 * Sin[theta2[t]] ;
ypend2 = 2 * l1 * Cos[theta1[t]] + l2 * Cos[theta2[t]] ;
tpend2 = 1/2 * m2 * ((D_t xpend2)^2 + (D_t ypend2)^2) + 1/2 * (1/3 * m2 * l2^2) * (theta2'[t])^2 ;

xmass = x[t] - 2 * l1 * Sin[theta1[t]] ;
ymass = 2 * l1 * Cos[theta1[t]] ;
tmass = 1/2 * m3 * ((D_t xmass)^2 + (D_t ymass)^2) ;

v = m1 * g * ypend1 + m2 * g * ypend2 + m3 * g * ymass ;
Simplify[v] ;

lang = TM + tpend1 + tpend2 + tmass - v ;
Simplify[lang] ;

ldad = D_t lang ;
Simplify[ldad] ;
f1 = D_t ldad - D_t lang ;
Simplify[f1] ;

ldbd = D_t lang ;
Simplify[ldbd] ;
f2 = D_t ldbd - D_t lang ;
Simplify[f2] ;

Solve[{f1 == 0, f2 == 0}, {theta1''[t], theta2''[t]}] ;
add = theta1''[t] /. % ;
bdd = theta2''[t] /. % ;

```

```

k11 =  $\partial_{x[t]}$  add /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k12 =  $\partial_{\theta_1[t]}$  add /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k13 =  $\partial_{\theta_2[t]}$  add /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k14 =  $\partial_{x'[t]}$  add /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /. a'[t]  $\rightarrow$  0 /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k15 =  $\partial_{\theta_1'[t]}$  add /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k16 =  $\partial_{\theta_2'[t]}$  add /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k17 =  $\partial_{x''[t]}$  add /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;

k21 =  $\partial_{x[t]}$  bdd /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k22 =  $\partial_{\theta_1[t]}$  bdd /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k23 =  $\partial_{\theta_2[t]}$  bdd /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k24 =  $\partial_{x'[t]}$  bdd /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k25 =  $\partial_{\theta_1'[t]}$  bdd /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k26 =  $\partial_{\theta_2'[t]}$  bdd /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;
k27 =  $\partial_{x''[t]}$  bdd /. x[t]  $\rightarrow$  0 /.  $\theta_1[t] \rightarrow 0$  /.  $\theta_2[t] \rightarrow 0$  /. x'[t]  $\rightarrow$  0 /.  $\theta_1'[t] \rightarrow 0$  /.
 $\theta_2'[t] \rightarrow 0$  /. x''[t]  $\rightarrow$  0;

Simplify[k12]
Simplify[k13]
Simplify[k17]
Simplify[k22]
Simplify[k23]
Simplify[k27]

m1 = 0.05 ; m2 = 0.13; m3 = 0.236; l1 = 0.0775; l2 = 0.25; g = 9.8 ;

k12
k13
k17
k22
k23
k27

```

因其余各项为 0，所以这里仅列举了 k_{12} 、 k_{13} 、 k_{17} 、 k_{22} 、 k_{23} 、 k_{27} 等 7 项，得到结果如下：

$$k_{12} = \frac{3g(m_1 + 2(m_2 + m_3))}{(4m_1 + 3(m_2 + 4m_3))l_1} = 86.69$$

$$k_{13} = \frac{-9gm_2}{2(4m_1 + 3(m_2 + 4m_3))l_1} = -21.62$$

$$k_{17} = \frac{3(2m_1 + m_2 - 4m_3)}{2(4m_1 + 3(m_2 + 4m_3))l_1} = 6.64$$

$$k_{22} = \frac{-9g(m_1 + 2(m_2 + m_3))}{2(4m_1 + 3(m_2 + 4m_3))l_2} = -40.31$$

$$k_{23} = \frac{3g(m_1 + 3(m_2 + m_3))}{(4m_1 + 3(m_2 + 4m_3))l_2} = 39.45$$

$$k_{27} = \frac{3(m_1 - l_2 m_3)}{2(4m_1 + 3(m_2 + 4m_3))l_2} = -0.088$$

于是由系统的状态方程为：

$$\begin{bmatrix} \dot{x} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{x} \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 86.69 & -21.62 & 0 & 0 & 0 \\ 0 & -40.31 & 39.45 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta_1 \\ \theta_2 \\ \dot{x} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 6.64 \\ -0.088 \end{bmatrix} u$$

$$y = \begin{bmatrix} x \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta_1 \\ \theta_2 \\ \dot{x} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} u$$

6.2 系统可控性分析

系统状态矩阵 A, B, C, D 如下：

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 86.69 & -21.62 & 0 & 0 & 0 \\ 0 & -40.31 & 39.45 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 6.64 \\ -0.088 \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

利用 MATLAB 计算系统状态可控性矩阵和输出可控性矩阵的秩：

```
clear;
A=[ 0  0      0  1  0  0;
    0  0      0  0  1  0;
    0  0      0  0  0  1;
    0  0      0  0  0  0;
    0  86.69 -21.62 0  0  0;
    0 -40.31 39.45  0  0  0];
B=[ 0  0  0  1  6.64 -0.088]';
C=[ 1  0  0  0  0  0;
    0  1  0  0  0  0;
    0  0  1  0  0  0];
D=[ 0 0 0 ]';
cona=[B A*B A^2*B A^3*B A^4*B A^5*B];
cona2=[C*B C*A*B C*A^2*B C*A^3*B C*A^4*B C*A^5*B D];
rank(cona)
rank(cona2)
```

得到结果如下：


```
ans =
     6
ans =
     3
```

或是通过 MATLAB 命令 `ctrb` 和 `obsv` 直接得到系统的可控性和可观测性。

```
Uc=ctrb(A,B);
Vo=obsv(A,C);

rank(Uc)
rank(Vo)
```

运行得到：

```
ans =
     6
ans =
     6
```

可以得到，系统状态和输出都可控，且系统具有可观测性。

6.3 直线两级倒立摆 MATLAB 仿真

在 MATLAB Simulink 中建立直线两级倒立摆的模型：

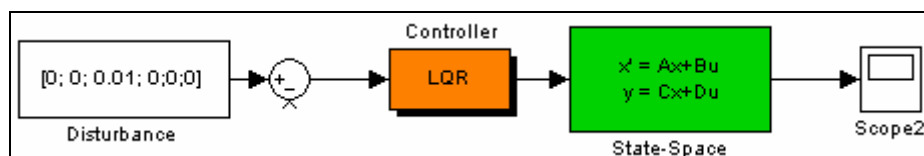
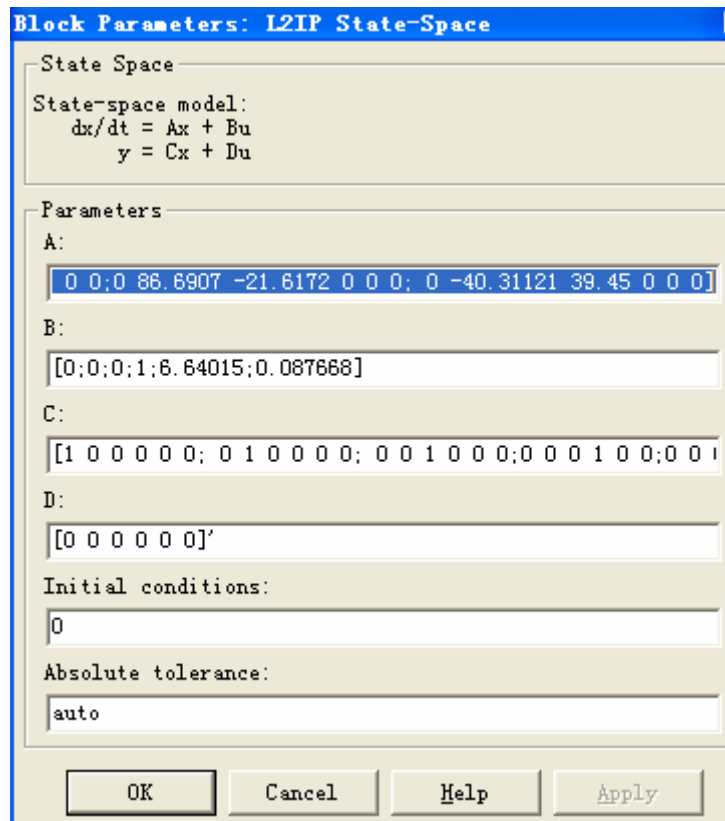
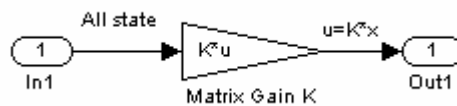


图 6-2 直线两级倒立摆开环仿真

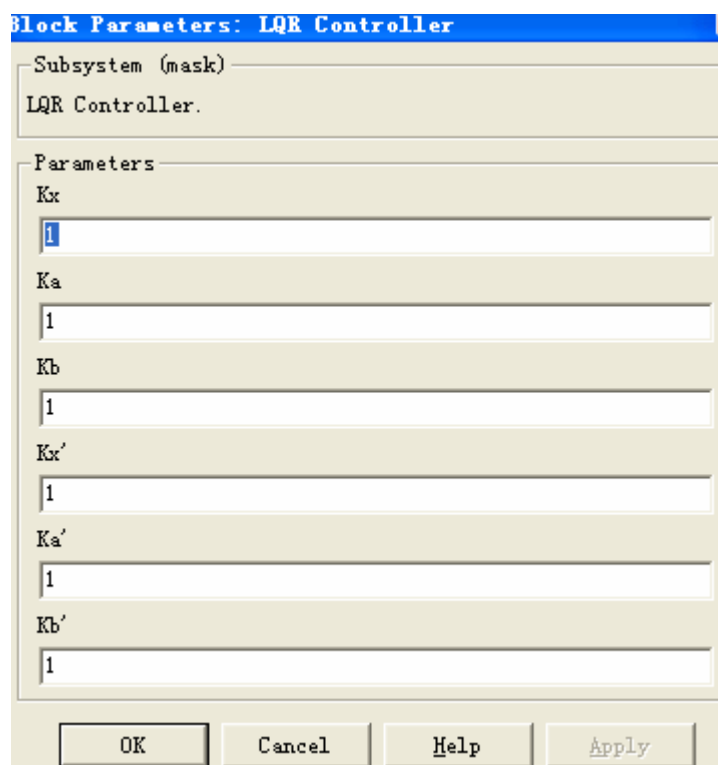
其中“State-Space”模块为直线两级倒立摆的状态方程，双击模块打开模型：



“Controller” 模块为控制器模块，在 “Controller” 模块上单击鼠标右键，选择 “Look under mask” 打开模型内部结构：



其中 “Matrix Gain K” 为反馈矩阵。
双击 “Controller” 模块打开其参数设置窗口：



先设置参数为“1”。

“Disturbance”模块为外界干扰模块，其作用是给系统施加一个阶跃信号，点击“▶”运行模型进行开环系统仿真。

得到运行结果如下：

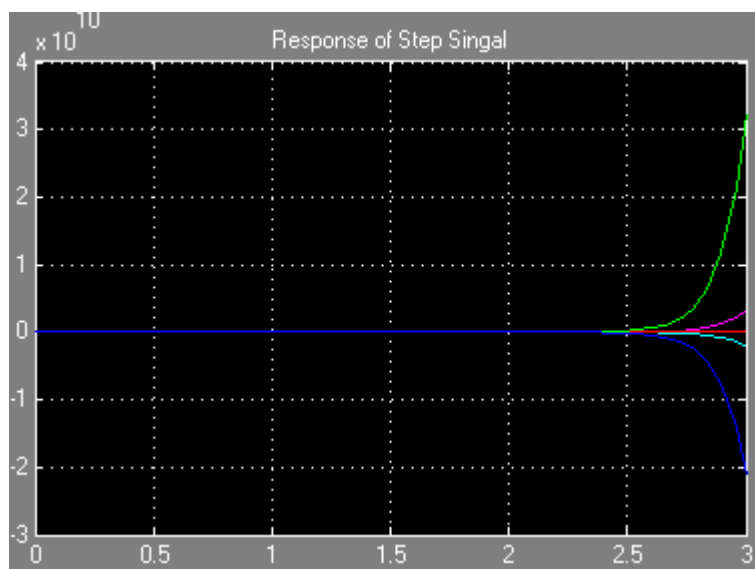


图 6-3 直线两级倒立摆开环仿真结果

从仿真结果可以看出，系统发散，为使系统稳定，需要对其添加控制器。

6.4 LQR 控制器设计及仿真

给系统添加 LQR 控制器，添加控制器后的系统闭环图如下图所示：

（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开

“Inverted Pendulum\Linear Inverted Pendulum\Linear 2-Stage IP Experiments”中的“LQR Control Simulink”)

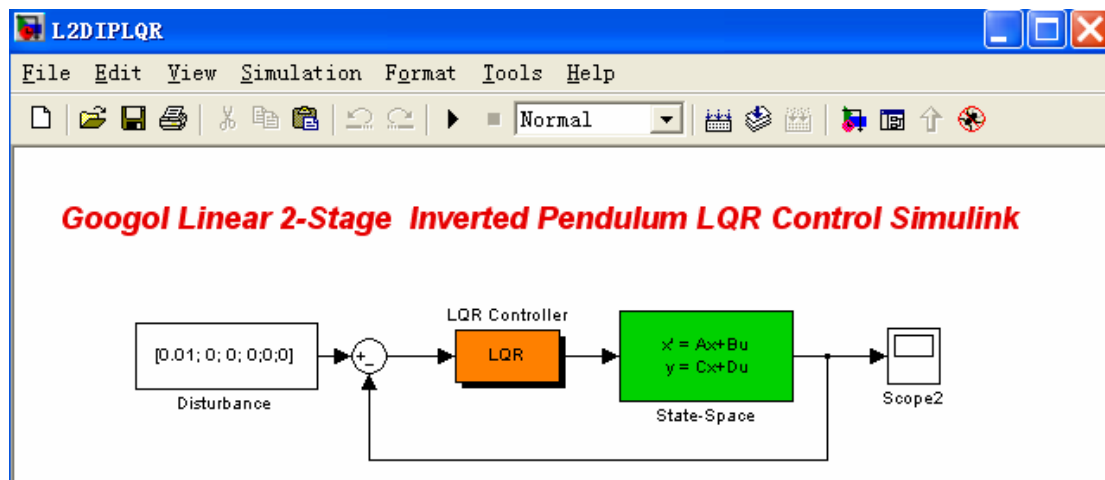


图 6-4 直线两级倒立摆 LQR 控制仿真

下面利用线性二次最优控制 LQR 方法对系统进行控制器的设计，线性二次最优控制 LQR 方法的原理请参见直线一级倒立摆“线性二次最优控制 LQR 控制实验”实验内容以及相关书籍：

（进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 2-Stage IP Experiments”中的 “LQR Control M Files”）

PRO 6-2 直线两级倒立摆 LQR 控制 MATLAB 计算程序

```
clear all;
format long;
k12=86.69;k13=-21.62;k17=6.64;
k22=-40.31;k23=39.45;k27=-0.088;

A=[0 0 0 1 0 0;
  0 0 0 0 1 0;
  0 0 0 0 0 1;
  0 0 0 0 0 0;
  0 k12 k13 0 0 0;
  0 k22 k23 0 0 0]

B=[ 0 0 0 1 k17 k27]';

C=[1 0 0 0 0 0;
  0 1 0 0 0 0;
  0 0 1 0 0 0];
D=[0; 0; 0];

Q11=1;Q22=1;Q33=1;
```

```

Q=[Q11  0  0  0  0  0;
   0  Q22  0  0  0  0;
   0  0  Q33  0  0  0;
   0  0  0  0  0  0;
   0  0  0  0  0  0;
   0  0  0  0  0  0];
R=1;
K=lqr(A,B,Q,R)

Ac=[(A-B*K)];
Bc=[B];
Cc=[C];
Dc=[D];
Cn=[1 0 0 0 0 0];

s=size(A,1);
Z=[zeros([1,s]) 1];

N=inv([A,B;Cn,0])*Z';
Nx=N(1:s);
Nu=N(1+s);
Nbar=Nu+K*Nx;

Bcn=[Nbar*B];
x0=[0 0 0.05 0 0 0];
[Y,X]=lsim(Ac,Bcn,Cc,Dc,U,T,x0);
xpos=Y(:,1);
xangle=Y(:,2);
xangle2=Y(:,3);

plot(T,xpos,'.')
hold on;
plot(T,xangle,'--')
hold on;
plot(T,xangle2,'-')

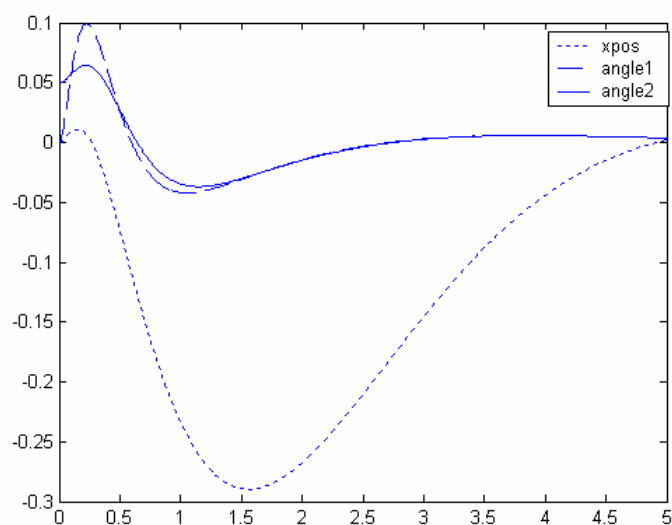
```

运行得到以下结果：

LQR 控制参数为：

$K = [1 \quad 73.818 \quad -83.941 \quad 2.0162 \quad 4.2791 \quad -13.036]$

得到仿真结果如下：



可以看出，系统稳定时间过长，因此增加权重 Q 的值。

设 $Q_{11}=300; Q_{22}=500; Q_{33}=500;$

运行得到仿真结果：

LQR 控制参数为：

$K=[17.321 \ 110.87 \ -197.57 \ 18.468 \ 2.7061 \ -32.142]$

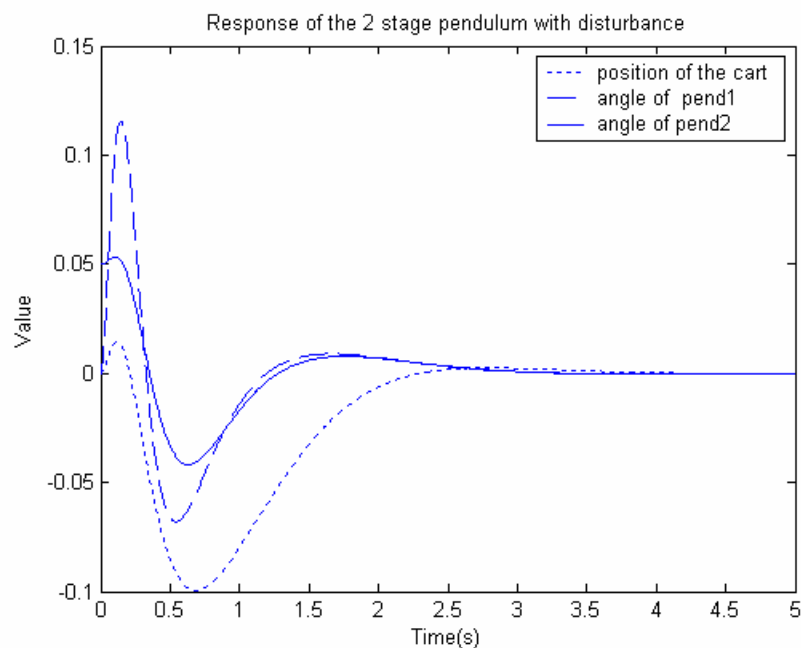


图 6-5 直线两级倒立摆 LQR 控制仿真结果 1

从图中可以看出，系统可以很好的稳定，在给定倒立摆干扰后，系统在 2.5 秒内可以恢复到平衡点附近。

把以上仿真参数输入 Simulink 模型中，



得到运行结果:

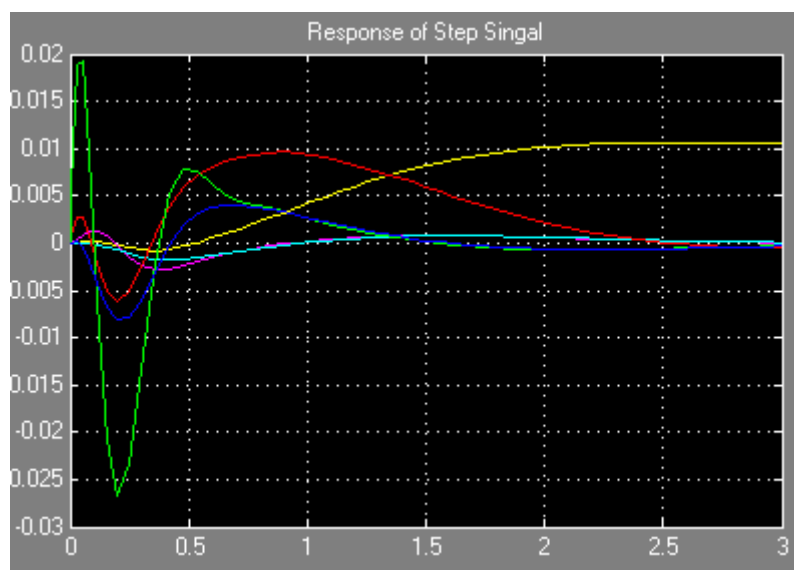


图 6-6 直线两级倒立摆 LQR 控制仿真结果 2

改变仿真中 LQR 参数，观察仿真结果变化，选取最合适的控制参数。

6.5 直线两级倒立摆 LQR 控制实验

实时控制实验在 MATLAB Simulink 环境下进行，用户在实验前请仔细阅读使用手册。



注 意

- 在进行 MATLAB 实时控制实验时，请用户检查倒立摆系统机械

结构和电气接线有无危险因素存在，在保障实验安全的情况下进行实验。

实验步骤：

7) 打开直线两级倒立摆控制程序如下图所示：

(进入 MATLAB Simulink 实时控制工具箱 “Googol Education Products” 打开 “Inverted Pendulum\Linear Inverted Pendulum\Linear 2-Stage IP Experiments” 中的 “LQR Control Demo”)

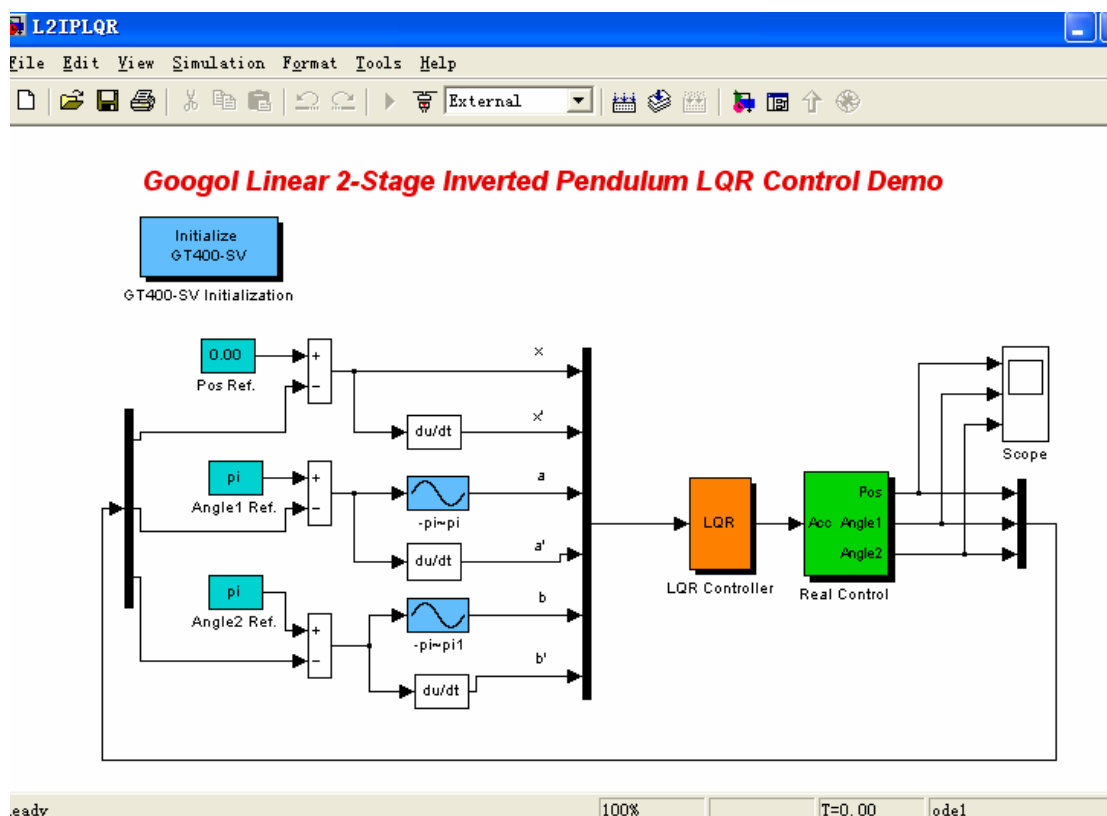
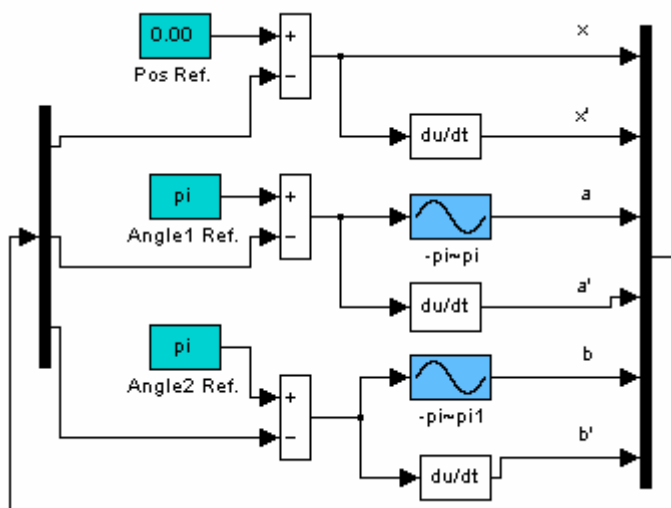


图 6-7 直线两级倒立摆 LQR 实时控制程序

左边部分为小车位置，摆杆 1 的角度和摆杆 2 的角度处理模块：



由上至下依次为小车位置、小车速度、摆杆 1 的角度、摆杆 1 的角速度、摆杆 2 的角度和摆杆 2 的角速度。

8) “LQR Controller” 为 LQR 控制器，双击模块设置参数：

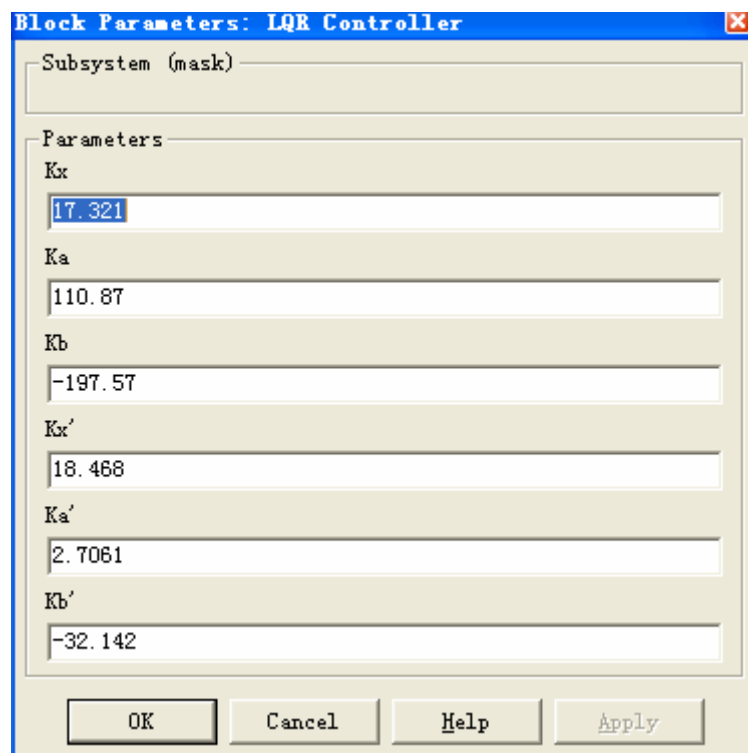





图 6-8 直线两级倒立摆 LQR 控制参数

把以上仿真得到的控制参数输入到 LQR 控制器中，请注意参数的顺序。

9) “Real Control” 为实时控制模块，输入为控制量——小车的加速度，输出为小车的位置，摆杆 1 的角度和摆杆 2 的角度。

10) 在确认系数正确后，点击“”编译程序。

11) 编译成功后，点击“”连接程序。

12) 点击“”运行程序，在电机上伺服后，缓慢提起摆杆到平衡位置，在程序进入自动控制后松开手，得到如下的实验结果：

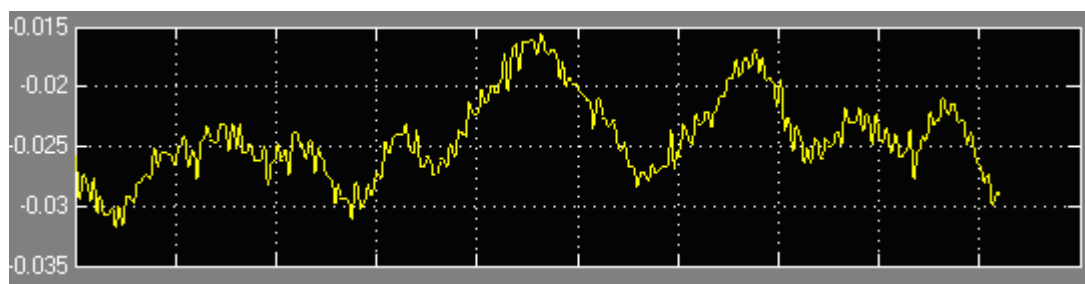


图 6-9 直线两级倒立摆 LQR 控制结果（小车位置）

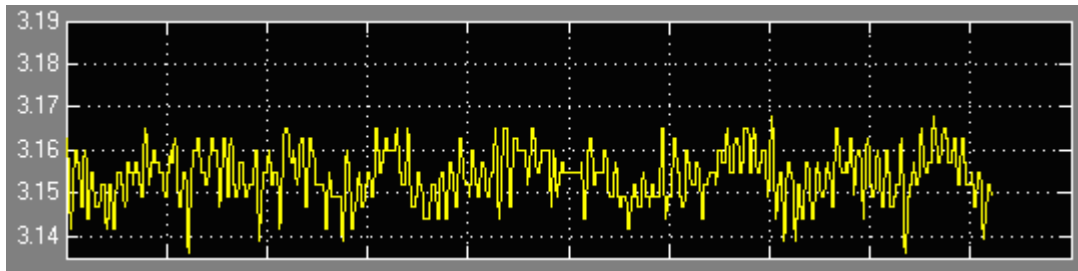


图 6-10 直线两级倒立摆 LQR 控制结果（摆杆 1 角度）

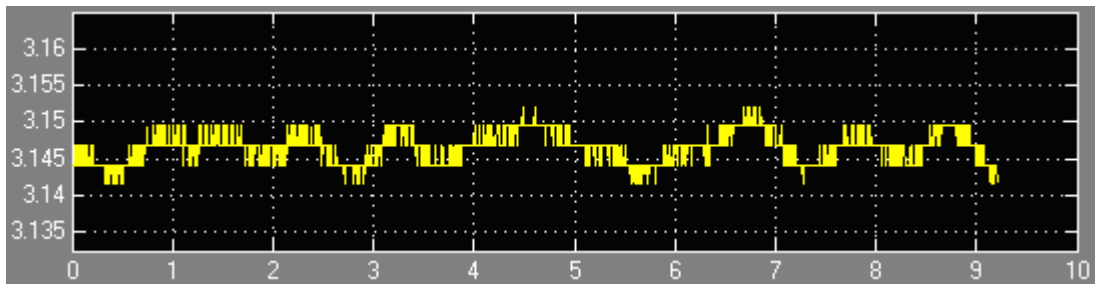


图 6-11 直线两级倒立摆 LQR 控制结果（摆杆 2 角度）

13) 给摆杆干扰后，系统的响应如下图所示：

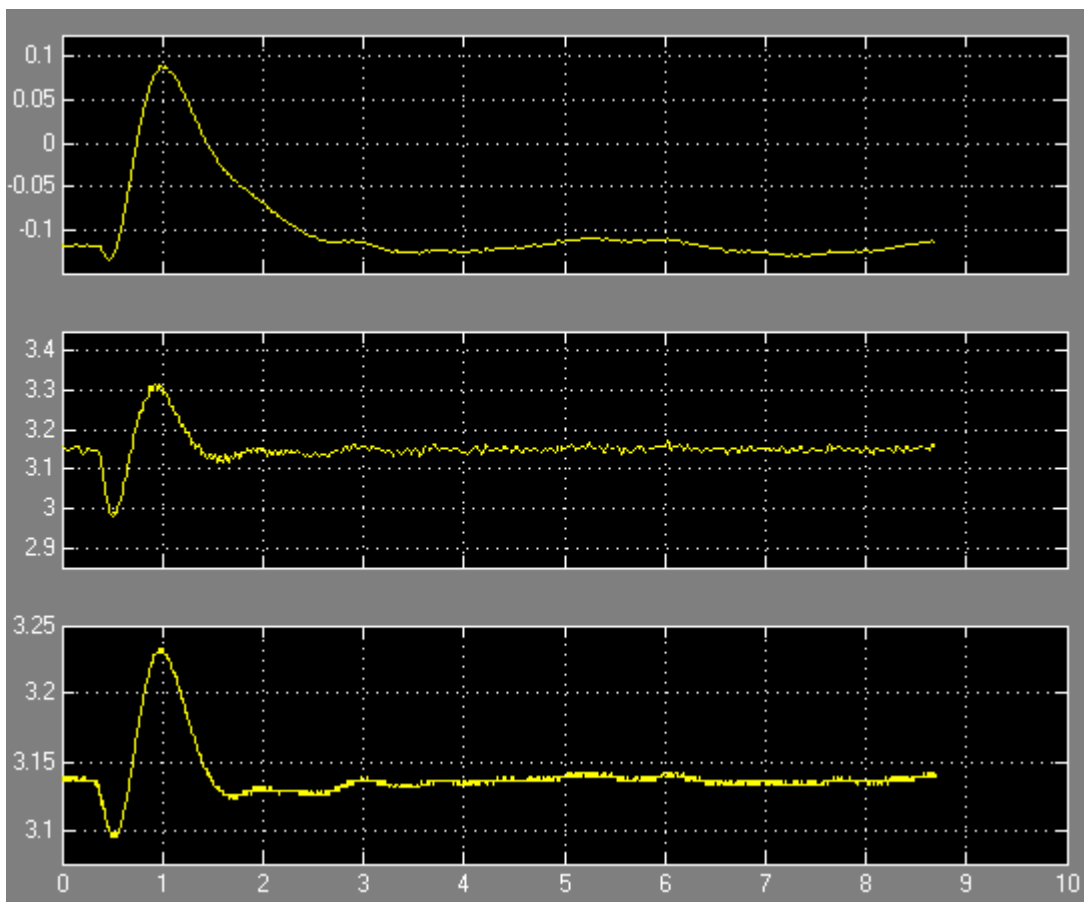


图 6-12 直线两级倒立摆 LQR 控制结果（给定干扰）

14) 可以看出，在不给定干扰的时候，倒立摆可以很好的保持稳定的状态，

虽然系统存在轻微的振动，小车位置振动幅值为 15mm，摆杆 1 的振动幅值约为 0.02 弧度，摆杆 2 的振动幅值约为 0.01 弧度。

15) 在给定摆杆干扰后，系统在 1 秒内可以达到新的平衡状态。超调量也较小。

16) 改变控制参数，在对参数进行仿真后，进行多次实验，观察和分析控制效果的变化。

6.6 实验结果及实验报告

包含实验结果，结果分析在实验报告中完成。

6.7 其它算法实验实现方法

在直线两级上可以施加很多种控制算法，目前使用较多的是模糊控制、鲁棒控制、自适应控制等控制算法。用户只需在直线两级倒立摆的实时控制程序中，将“LQR Controller”模块更换成用户自定义的算法即可。

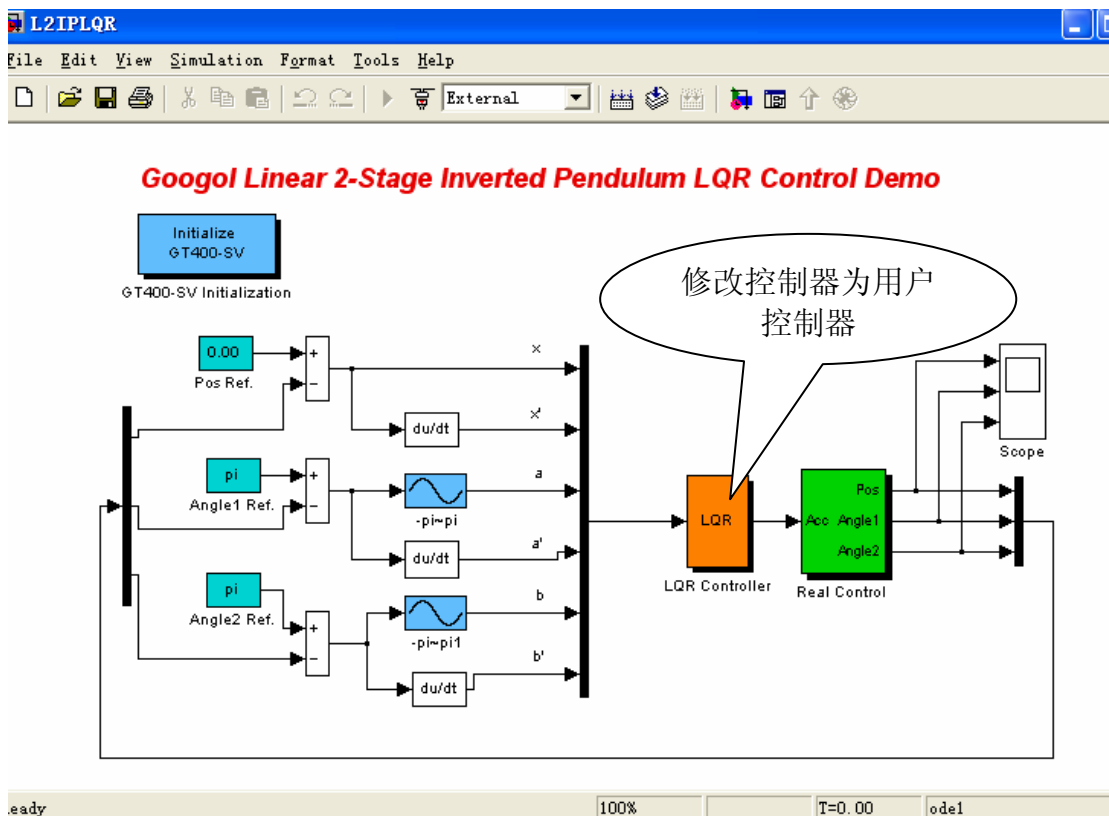


图 6-13 直线两级倒立摆控制器修改实验

第7章 直线三级倒立摆

7.1 系统物理模型

为简化系统，我们在建模时忽略了空气阻力和各种摩擦力的作用，并认为摆杆为刚体。直线三级倒立摆的组成如图 6-1 所示：

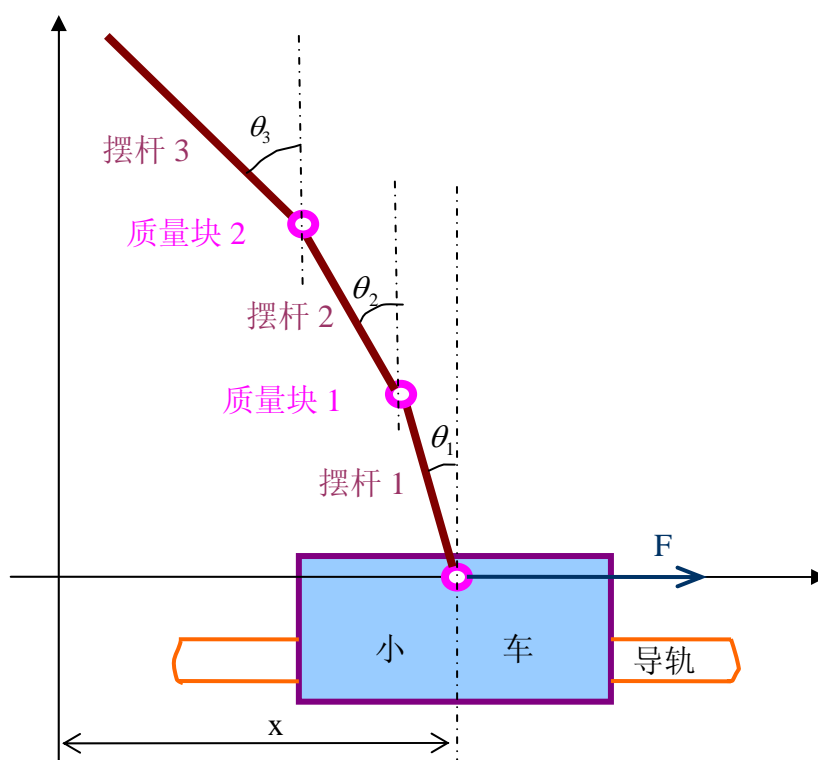


图 7-1 直线三级倒立摆物理模型

倒立摆参数定义如下：

M 小车质量

m_1 摆杆 1 的质量

m_2 摆杆 2 的质量

m_3 摆杆 3 的质量

m_4 质量块 1 的质量

m_5 质量块 2 的质量

l_1 摆杆 1 中心到转动中心的距离

l_2 摆杆 2 中心到转动中心的距离

l_3 摆杆 3 中心到转动中心的距离

θ_1 摆杆 1 与竖直方向的夹角

θ_2 摆杆 2 与竖直方向的夹角

θ_3 摆杆 2 与竖直方向的夹角

F 作用在系统上的外力

利用拉格朗日方程推导运动学方程：

拉格朗日方程为：

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q, \dot{q})$$

其中 L 为拉格朗日算子， q 为系统的广义坐标， T 为系统的动能， V 为系统的势能。

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = f_i$$

其中 $i=1,2,3,\dots,n$ ， f_i 为系统在第 i 个广义坐标上的外力，在二级倒立摆系统中，系统的广义坐标有三个广义坐标，分别为 $x, \theta_1, \theta_2, \theta_3$ 。

首先计算系统的动能：

$$T = T_M + T_{m1} + T_{m2} + T_{m3} + T_{m4} + T_{m5}$$

其中 $T_M, T_{m1}, T_{m2}, T_{m3}, T_{m4}, T_{m5}$ 分别为小车的动能，摆杆 1 的动能，摆杆 2 的动能，摆杆 3 的动能，质量块 1 的动能和质量块 2 的动能。
小车的动能：

$$T_M = \frac{1}{2} M \dot{x}^2$$

$T_{m1} = T_{m1}' + T_{m1}''$ 其中 T_{m1}', T_{m1}'' 分别为摆杆 1 的平动动能和转动动能。

$T_{m2} = T_{m2}' + T_{m2}''$ 其中 T_{m2}', T_{m2}'' 分别为摆杆 2 的平动动能和转动动能。

$T_{m3} = T_{m3}' + T_{m3}''$ 其中 T_{m3}', T_{m3}'' 分别为摆杆 3 的平动动能和转动动能。

对于系统，设以下变量：

xpend1 摆杆 1 质心横坐标；

ypend1 摆杆 1 质心纵坐标;
 xpend2 摆杆 2 质心横坐标;
 ypend2 摆杆 2 质心纵坐标;
 xpend3 摆杆 3 质心横坐标;
 ypend3 摆杆 3 质心纵坐标;
 xmass1 质量块 1 质心横坐标;
 ymass1 质量块 1 质心纵坐标;
 xmass2 质量块 2 质心横坐标;
 ymass2 质量块 2 质心纵坐标;
 又有:

$$\begin{cases} xpend1 = x - l_1 \sin \theta_1 \\ ypend1 = l_1 \cos \theta_1 \end{cases}$$

$$\begin{cases} xmass1 = x - 2l_1 \sin \theta_1 \\ ymass1 = 2l_1 \cos \theta_1 \end{cases}$$

$$\begin{cases} xpend2 = xmass1 - l_2 \sin \theta_2 \\ ypend2 = ymass1 + l_2 \cos \theta_2 \end{cases}$$

$$\begin{cases} xmass2 = xmass1 - 2l_2 \sin \theta_2 \\ ymass2 = ymass1 + 2l_2 \cos \theta_2 \end{cases}$$

$$\begin{cases} xpend3 = xmass2 - l_3 \sin \theta_3 \\ ypend3 = ymass2 + l_3 \cos \theta_3 \end{cases}$$

则有:

$$T_{m1}' = \frac{1}{2} m_1 \left(\left(\frac{d(xpend1)}{dt} \right)^2 + \left(\frac{d(ypend1)}{dt} \right)^2 \right)$$

$$T_{m1}'' = \frac{1}{2} J_{p1} \dot{\theta}_1^2 = \frac{1}{6} m_1 l_1^2 \dot{\theta}_1^2$$

同理:

$$T_{m2}' = \frac{1}{2} m_2 \left(\left(\frac{d(xpend2)}{dt} \right)^2 + \left(\frac{d(ypend2)}{dt} \right)^2 \right)$$

$$T_{m2}'' = \frac{1}{2} J_{p2} \dot{\theta}_2^2 = \frac{1}{6} m_2 l_2^2 \dot{\theta}_2^2$$

$$T_{m3}' = \frac{1}{2} m_3 \left(\left(\frac{d(xpend3)}{dt} \right)^2 + \left(\frac{d(ypend3)}{dt} \right)^2 \right)$$

$$T_{m3}'' = \frac{1}{2} J_{p3} \dot{\theta}_3^2 = \frac{1}{6} m_3 l_3^2 \dot{\theta}_3^2$$

$$T_{m4} = \frac{1}{2} m_4 \left(\left(\frac{d(xmass1)}{dt} \right)^2 + \left(\frac{d(ymass1)}{dt} \right)^2 \right)$$

$$T_{m5} = \frac{1}{2} m_5 \left(\left(\frac{d(xmass2)}{dt} \right)^2 + \left(\frac{d(ymass2)}{dt} \right)^2 \right)$$

于是有系统的总动能:

$$\begin{aligned} T &= T_M + T_{m1} + T_{m2} + T_{m3} + T_{m4} + T_{m5} \\ &= \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m_1 \left(\left(\frac{d(xpend1)}{dt} \right)^2 + \left(\frac{d(ypend1)}{dt} \right)^2 \right) + \frac{1}{6} m_1 l_1^2 \dot{\theta}_1^2 \\ &\quad + \frac{1}{2} m_2 \left(\left(\frac{d(xpend2)}{dt} \right)^2 + \left(\frac{d(ypend2)}{dt} \right)^2 \right) + \frac{1}{6} m_2 l_2^2 \dot{\theta}_2^2 \\ &\quad + \frac{1}{2} m_3 \left(\left(\frac{d(xpend3)}{dt} \right)^2 + \left(\frac{d(ypend3)}{dt} \right)^2 \right) + \frac{1}{6} m_3 l_3^2 \dot{\theta}_3^2 \\ &\quad + \frac{1}{2} m_4 \left(\left(\frac{d(xmass1)}{dt} \right)^2 + \left(\frac{d(ymass1)}{dt} \right)^2 \right) \\ &\quad + \frac{1}{2} m_5 \left(\left(\frac{d(xmass2)}{dt} \right)^2 + \left(\frac{d(ymass2)}{dt} \right)^2 \right) \end{aligned}$$

系统的势能为:

$$V = V_{m1} + V_{m2} + V_{m3} + V_{m4} + V_{m5} = m_1 ypend1 + m_2 ypend2 + m_3 ypend3 + m_4 ymass1 + m_5 ymass2$$

由于系统在 $\theta_1, \theta_2, \theta_3$ 广义坐标下没有外力作用, 所以有:

$$\begin{cases} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} = 0 \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} = 0 \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_3} - \frac{\partial L}{\partial \theta_3} = 0 \end{cases}$$

对于三级倒立摆系统, 选取系统状态变量为:

$$\left\{ x, \theta_1, \theta_2, \theta_3, \dot{x}, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3 \right\}$$

为求解状态方程:

$$\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases}$$

需要求解 $\ddot{\theta}_1, \ddot{\theta}_2$ 和 $\ddot{\theta}_3$ ，因此设：

$$\begin{cases} \ddot{\theta}_1 = f_1(x, \theta_1, \theta_2, \theta_3, \dot{x}, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \ddot{x}) \\ \ddot{\theta}_2 = f_2(x, \theta_1, \theta_2, \theta_3, \dot{x}, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \ddot{x}) \\ \ddot{\theta}_3 = f_3(x, \theta_1, \theta_2, \theta_3, \dot{x}, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \ddot{x}) \end{cases}$$

将在平衡位置附近进行泰勒级数展开，并线性化，可以得到：

$$\begin{cases} \ddot{\theta}_1 = k_{11}x + k_{12}\theta_1 + k_{13}\theta_2 + k_{14}\theta_3 + k_{15}\dot{x} + k_{16}\dot{\theta}_1 + k_{17}\dot{\theta}_2 + k_{18}\dot{\theta}_3 + k_{19}\ddot{x} \\ \ddot{\theta}_2 = k_{21}x + k_{22}\theta_1 + k_{23}\theta_2 + k_{24}\theta_3 + k_{25}\dot{x} + k_{26}\dot{\theta}_1 + k_{27}\dot{\theta}_2 + k_{28}\dot{\theta}_3 + k_{29}\ddot{x} \\ \ddot{\theta}_3 = k_{31}x + k_{32}\theta_1 + k_{33}\theta_2 + k_{34}\theta_3 + k_{35}\dot{x} + k_{36}\dot{\theta}_1 + k_{37}\dot{\theta}_2 + k_{38}\dot{\theta}_3 + k_{39}\ddot{x} \end{cases}$$

其中其中 k_{ij} 为 f_i 对其第 j 个变量的偏微分系数，例如：

$$k_{11} = \left. \frac{\partial f_1}{\partial x} \right|_{x=0, \theta_1=0, \theta_2=0, \theta_3=0, \dot{x}=0, \dot{\theta}_1=0, \dot{\theta}_2=0, \dot{\theta}_3=0, \ddot{x}=0}$$

在 Mathematics 中计算以上各式：

PRO 7-1 直线三级倒立摆建模 Mathematics 程序

```

tcar = 1/2 mcar * x'[t]^2 ;

xpend1 = x[t] - l1 * Sin[a[t]] ;
ypend1 = l1 * Cos[a[t]] ;
xdotpend1 = D[xpend1, t] ;
ydotpend1 = D[ypend1, t] ;
tpend1 = 1/2 * m1 * (xdotpend1^2 + ydotpend1^2) +
1/2 * (1/3 * m1 * l1^2) * a'[t]^2 ;
Simplify[tpend1] ;

xmass1 = x[t] - 2 * l1 * Sin[a[t]] ;
ymass1 = 2 * l1 * Cos[a[t]] ;
xdotmass1 = D[xmass1, t] ;
ydotmass1 = D[ymass1, t] ;
tmass1 = 1/2 * m4 * (xdotmass1^2 + ydotmass1^2) ;
Simplify[tmass1] ;

xpend2 = xmass1 - l2 * Sin[b[t]] ;
ypend2 = ymass1 + l2 * Cos[b[t]] ;
xdotpend2 = D[xpend2, t] ;
ydotpend2 = D[ypend2, t] ;
tpend2 = 1/2 * m2 * (xdotpend2^2 + ydotpend2^2) +
1/2 * (1/3 * m2 * l2^2) * (b'[t])^2 ;
Simplify[tpend2] ;

```



```

xmass2 = xmass1 - 2 * l2 * Sin[b[t]];
ymass2 = ymass1 + 2 * l2 * Cos[b[t]];
xdotmass2 = D[xmass2, t];
ydotmass2 = D[ymass2, t];
tmass2 = 1/2 * m5 * (xdotmass2^2 + ydotmass2^2);
Simplify[tmass2];

xpend3 = xmass2 - l3 * Sin[c[t]];
ypend3 = ymass2 + l3 * Cos[c[t]];
xdotpend3 = D[xpend3, t];
ydotpend3 = D[ypend3, t];
tpend3 = 1/2 * m3 * (xdotpend3^2 + ydotpend3^2) +
    1/2 * (1/3 * m3 * l3^2) * (c'[t])^2;
Simplify[tpend3];

vpend1 = m1 * g * ypend1; vpend2 = m2 * g * ypend2; vpend3 = m3 * g * ypend3;
vmass1 = m4 * g * ymass1; vmass2 = m5 * g * ymass1;
v = vpend1 + vpend2 + vpend3 + vmass1 + vmass2;
Simplify[v];

lang = tcar + tpend1 + tpend2 + tpend3 + tmass1 + tmass2 - v;
Simplify[lang];

ldad =  $\partial_{a'[t]}$  lang;
fa =  $\partial_t$  ldad -  $\partial_{a[t]}$  lang;
Simplify[fa];

ldbd =  $\partial_{b'[t]}$  lang;
fb =  $\partial_t$  ldbd -  $\partial_{b[t]}$  lang;
Simplify[fb];

ldcd =  $\partial_{c'[t]}$  lang;
fc =  $\partial_t$  ldcd -  $\partial_{c[t]}$  lang;
Simplify[fc];

Solve[{fa == 0, fb == 0, fc == 0}, {a''[t], b''[t], c''[t]};
add = a''[t] /. %;
bdd = b''[t] /. %%;
cdd = c''[t] /. %%%;

```

[illegible]

[illegible]

```

m1 = 0.05;
m2 = 0.06;
m3 = 0.13;
m4 = 0.236;
m5 = 0.236;
l1 = 0.0775;
l2 = 0.1;
l3 = 0.25;
g = 9.8;

Print["k11->k19"]
Simplify[k11]
Simplify[k12]
Simplify[k13]
Simplify[k14]
Simplify[k15]
Simplify[k16]
Simplify[k17]
Simplify[k18]
Simplify[k19]

Print["k21->k29"]
Simplify[k21]
Simplify[k22]
Simplify[k23]
Simplify[k24]
Simplify[k25]
Simplify[k26]
Simplify[k27]
Simplify[k28]
Simplify[k29]

Print["k31->k39"]
Simplify[k31]
Simplify[k32]
Simplify[k33]
Simplify[k34]
Simplify[k35]
Simplify[k36]
Simplify[k37]
Simplify[k38]
Simplify[k39]

```

(计算文件请参见“\GLIP2003\建模文件\MathematicaFile”中的“L3DIP.nb”文件。)

运行计算程序，得到如下结果：

$$k_{12} = 159.504$$

$$k_{13} = -38.4356$$

$$k_{14} = 0.784646$$

$$k_{19} = 6.64904$$

$$k_{22} = -127.9$$

$$k_{23} = 57.9951$$

$$k_{24} = -17.189$$

$$k_{29} = -0.15831$$

$$k_{32} = 2.57086$$

$$k_{33} = -16.9245$$

$$k_{34} = 39.3485$$

$$k_{39} = 0.00318211$$

于是可以得到系统的状态方程：

$$\begin{bmatrix} \dot{x} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \ddot{x} \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 159.504 & -38.4356 & 0.784646 & 0 & 0 & 0 & 0 \\ 0 & -127.9 & 57.9951 & -17.189 & 0 & 0 & 0 & 0 \\ 0 & 2.57086 & -16.9245 & 39.3485 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \dot{x} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 6.64904 \\ -0.15831 \\ 0.00318211 \end{bmatrix} u$$

$$y = \begin{bmatrix} x \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \dot{x} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

说明：实际系统模型和此模型可能存在差异，并且此模型没有经过实验验证，用户在使用时请自行检验。

第8章 附录

8.1 图表目录

图 1-1 直线倒立摆系列	2
图 1-2 环形倒立摆系列	2
图 1-3 平面倒立摆系列	3
图 1-4 复合倒立摆	3
图 2-1 光电编码器原理示意图	5
图 2-2 MATLAB Simulink 环境	7
图 3-1 直线一级倒立摆模型	25
图 3-2 小车及摆杆受力分析	25
图 3-3 直线一级倒立摆单位阶跃响应仿真	36
图 3-4 直线一级倒立摆闭环系统图（脉动干扰）	37
图 3-5 直线一级倒立摆闭环系统简化图（脉动干扰）	37
图 3-6 直线一级倒立摆开环根轨迹图	39
图 3-7 性能指标与根轨迹关系图	40
图 3-8 直线一级倒立摆根轨迹计算图	41
图 3-9 校正后的根轨迹图	43
图 3-10 校正后的阶跃响应	44
图 3-11 手动校正后的根轨迹图（增加零点）	44
图 3-12 手动校正后的根轨迹图（增加零点）	45
图 3-13 手动校正后的阶跃响应曲线	45
图 3-14 直线一级倒立摆的根轨迹校正仿真结果（一阶控制器）	51
图 3-15 直线一级倒立摆的根轨迹仿真模型	52
图 3-16 直线一级倒立摆的根轨迹校正仿真结果（二阶控制器）	53
图 3-17 直线一级倒立摆的根轨迹实时控制模块	54
图 3-18 直线一级倒立摆的根轨迹控制结果（平衡）	55
图 3-19 直线一级倒立摆的根轨迹控制结果（抵抗干扰）	55
图 3-20 直线一级倒立摆的 Bode 图	59
图 3-21 直线一级倒立摆的奈奎斯特图	59
图 3-22 添加增益后的直线一级倒立摆的 Bode 图和 Nyquist 图	61
图 3-23 添加控制器后的直线一级倒立摆 Bode 图和 Nyquist 图（一阶控制器）	64
图 3-24 利用频率响应方法校正后系统的单位阶跃响应（一阶控制器） ..	64
图 3-25 直线一级倒立摆的频率响应校正仿真程序	65
图 3-26 利用频率响应方法校正后的 Bode 图和 Nyquist 图（二阶控制器）	67
图 3-27 频率响应校正后阶跃响应仿真结果（二阶控制器）	68
图 3-28 直线一级倒立摆的频率响应校正实时控制程序	69
图 3-29 频率响应校正实时控制结果（一阶控制器）	70

图 3-30 频率响应校正实时控制结果（二阶控制器）	71
图 3-31 直线一级倒立摆闭环系统图	72
图 3-32 直线一级倒立摆闭环系统简化图	72
图 3-33 直线一级倒立摆 PID 控制 MATLAB 仿真模型	74
图 3-34 PID 参数设置窗口	74
图 3-35 直线一级倒立摆 P 控制仿真结果图 ($K_p=9$)	74
图 3-36 直线一级倒立摆 P 控制仿真结果图 ($K_p=40$)	75
图 3-37 直线一级倒立摆 PD 控制仿真结果图 ($K_p=40, K_d=4$)	75
图 3-38 直线一级倒立摆 PD 控制仿真结果图 ($K_p=40, K_d=10$)	75
图 3-39 直线一级倒立摆 PID 控制仿真结果图 ($K_p=40, K_i=20, K_d=4$)	76
图 3-40 直线一级倒立摆 PD 控制仿真结果图（小车位置曲线）	76
图 3-41 直线一级倒立摆 PID 控制 MATLAB 仿真结果（脉冲干扰）	77
图 3-42 直线一级倒立摆 MATLAB 实时控制界面	78
图 3-43 直线一级倒立摆 PID 控制实验结果 1	79
图 3-44 直线一级倒立摆 PID 控制实验结果 2（施加干扰）	80
图 3-45 直线一级倒立摆 PID 控制实验结果 3（改变 PID 控制参数）	81
图 3-46 状态反馈闭环控制原理图	82
图 3-47 倒立摆极点配置原理图	84
图 3-48 极点配置仿真结果	86
图 3-49 直线一级倒立摆状态空间极点配置 MATLAB Simulink 仿真	92
图 3-50 直线一级倒立摆状态空间极点配置 MATLAB Simulink 仿真结果	94
图 3-51 直线一级倒立摆状态空间极点配置实时控制程序	95
图 3-52 直线一级倒立摆状态空间极点配置实时控制结果（平衡）	96
图 3-53 直线一级倒立摆状态空间极点配置实时控制结果（施加干扰）	96
图 3-54 最优控制 LQR 控制原理图	98
图 3-55 直线一级倒立摆 LQR 控制仿真模型	101
图 3-56 直线一级倒立摆 LQR 控制仿真结果 1	102
图 3-57 直线一级倒立摆 LQR 控制仿真结果 2	102
图 3-58 直线一级倒立摆 LQR 控制实时控制程序	103
图 3-59 直线一级倒立摆 LQR 控制实时控制结果	105
图 4-1 直线一级顺摆物理模型图	107
图 4-2 直线一级顺摆的根轨迹图	114
图 4-3 直线一级顺摆的 Bode 图	114
图 4-4 直线一级顺摆的 Nyquist 图	115
图 4-5 直线一级顺摆的阶跃响应结果	115
图 4-6 直线一级顺摆的 PID 控制仿真	116
图 4-7 直线一级顺摆 P 控制结果	117
图 4-8 直线一级顺摆 PD 控制结果	118
图 4-9 直线一级顺摆 PID 控制器	119
图 4-10 直线一级顺摆 PID 控制结果	119
图 4-11 直线一级顺摆 PID 实时控制程序	120
图 4-12 直线一级顺摆 PID 实时控制结果	121
图 4-13 直线一级顺摆仿真结果图（给小车施加干扰）	124

图 4-14 直线一级顺摆 Simulink 模型.....	124
图 4-15 直线一级顺摆 LQR 控制仿真结果	126
图 4-16 直线一级顺摆 LQR 实时控制程序	127
图 4-17 直线一级顺摆 LQR 控制结果	129
图 5-1 直线一级倒立摆摆起实时控制程序	131
图 5-2 直线一级倒立摆摆起摆控制结果（小车位置）	132
图 5-3 直线一级倒立摆摆起摆控制结果（摆杆角度）	133
图 5-4 用户控制器添加方法示意图（单输入单输出）	133
图 5-5 用户控制器添加方法示意图（单输入多输出）	134
图 5-6 起摆控制算法修改示意图	134
图 6-1 直线两级倒立摆物理模型	135
图 6-2 直线两级倒立摆开环仿真	143
图 6-3 直线两级倒立摆开环仿真结果	145
图 6-4 直线两级倒立摆 LQR 控制仿真	146
图 6-5 直线两级倒立摆 LQR 控制仿真结果 1	148
图 6-6 直线两级倒立摆 LQR 控制仿真结果 2	149
图 6-7 直线两级倒立摆 LQR 实时控制程序	150
图 6-8 直线两级倒立摆 LQR 控制参数	151
图 6-9 直线两级倒立摆 LQR 控制结果（小车位置）	151
图 6-10 直线两级倒立摆 LQR 控制结果（摆杆 1 角度）	152
图 6-11 直线两级倒立摆 LQR 控制结果（摆杆 2 角度）	152
图 6-12 直线两级倒立摆 LQR 控制结果（给定干扰）	152
图 6-13 直线两级倒立摆控制器修改实验	153
图 7-1 直线三级倒立摆物理模型	154
表 1-1 警告标志	II
表 1-1 倒立摆实验	4
表 3-1 直线一级倒立摆根轨迹实时控制实验报告	57

8.2 程序目录

PRO 3-1 直线一级倒立摆建模程序	31
PRO 3-2 直线一级倒立摆根轨迹计算 Mathematics 程序	42
PRO 3-3 直线一级倒立摆根轨迹计算 MATLAB 程序	43
PRO 3-4 直线一级倒立摆根轨迹校正 MATLAB 程序 2	46
PRO 3-5 直线一级倒立摆的频率响应校正 MATLAB 程序	63
PRO 3-6 直线一级倒立摆 PID 控制 MATLAB 仿真程序	77
PRO 3-7 直线一级倒立摆状态空间极点配置 MATLAB 程序 1	87
PRO 3-8 直线一级倒立摆状态空间极点配置 MATLAB 程序 2	89
PRO 3-9 直线一级倒立摆状态空间极点配置 MATLAB 程序 3（爱克曼公式）	91
PRO 3-10 直线一级倒立摆状态空间极点配置 MATLAB 程序 4（爱克曼公式）	92

PRO 3-11 直线一级倒立摆 LQR 控制 MATLAB 计算程序	100
PRO 4-1 直线一级顺摆建模程序	110
PRO 4-2 直线一级顺摆 LQR 控制 MATLAB 计算程序	123
PRO 6-1 直线两级倒立摆建模 Mathematics 程序	139
PRO 6-2 直线两级倒立摆 LQR 控制 MATLAB 计算程序	146
PRO 7-1 直线三级倒立摆建模 Mathematics 程序	158