

上海理工大学光电信息与计算机工程学院

## 《智能仿真实验》报告



专    业    智能科学与技术

姓    名    高浩琦

学    号    2035060413

年    级    2020 级

指导教师    陈玮

成    绩:

教师签字:

## 实验一 多项式曲线拟合及最小二乘曲线拟合

- 一、实验目的
- 二、实验原理
  - 2.1 多项式拟合原理
  - 2.2 最小二乘拟合原理
  - 2.3 协方差矩阵
  - 2.4 MATLAB 中多项式拟合方法
    - 2.4.1 polyfit函数
    - 2.4.2 polyval 函数
  - 2.5 拟合优度分析原理
- 三、实验内容
- 四、分析过程及部分代码
  - 1. 相关性分析
  - 2. 多项式拟合
  - 3. 拟合优度分析
  - 4. 数学模型
- 五、实验小结

## 实验二 基于GUI的高尔夫手推车生产销售程序设计

- 一、实验目的
- 二、部分实验控件介绍
- 三、实验设备
- 四、实验内容
- 五、实验设计过程
  - 5.1 界面设计
  - 5.2 回调函数设计
- 六、实验小结

## 实验三 微分方程的Simulink求解及子系统封装

- 一、实验目的
- 二、实验设备
- 三、实验内容
  - (一) Lorenz模型的状态方程表示为：
  - (二) 微分代数方程仿真：
  - (三) 子系统封装及仿真
- 四、实验过程
  - 4.1 Lorenz模型的状态方程
  - 4.2 微分方程仿真
  - 4.3 子系统封装及仿真
- 五、实验小结

## 实验四 二阶模糊逻辑控制系统仿真

- 一、实验目的
- 二、实验内容
- 三、实验过程及分析
- 四、实验小结

# 实验一 多项式曲线拟合及最小二乘曲线拟合

## 一、实验目的

1. 掌握多项式曲线拟合和最小二乘拟合的原理和方法；
2. 掌握Matlab中多项式拟合及最小二乘曲线拟合的方法。

## 二、实验原理

### 2.1 多项式拟合原理

已知变量x,y之间的函数关系为

$$y = a_1 x^n + a_2 x^{n-1} + \cdots + a_n x + a_{n+1}$$

现希望通过实验获得一组 $\{x_i, y_i | i = 1, 2, \cdots, m\}$ 测量数据, 确定出系数 $(a_1, a_2, \cdots, a_{n+1})$ , 这类问题就是多项式拟合。

### 2.2 最小二乘拟合原理

最小二乘法多项式曲线拟合, 根据给定的m个点,并不要求这条曲线精确地经过这些点, 而是曲线 $y=f(x)$

的近似曲线 $y = \varphi(x)$ 。使偏差平方和最小, 即  $\min_{\varphi} \sum_{i=1}^m \delta_i^2 = \sum_{i=1}^m (\varphi(x_i) - y_i)^2$

按偏差平方和最小的原则选取拟合曲线, 并且采取二项式方程为拟合曲线的方法,称为最小二乘法。

### 2.3 协方差矩阵

$$\text{corrcoef}(X, Y) = \begin{bmatrix} \text{Cov}(X, X) & \text{Cov}(X, Y) \\ \text{Cov}(Y, X) & \text{Cov}(Y, Y) \end{bmatrix}$$

数字越接近于1 说明相关性越强

### 2.4 MATLAB 中多项式拟合方法

#### 2.4.1 polyfit函数

常用函数用法:

$$p = \text{polyfit}(x, y, n)$$

$$[p, s] = \text{polyfit}(x, y, n)$$

$$[p, S, mu] = \text{polyfit}(x, y, n)$$

返回次数为n的多项式p(x)的系数, 该阶数是y中数据的最佳拟合 (在最小二乘方式中)p中的系数按降幂排列, p的长度为n+1, 函数模型为:

$$p(x) = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_{n+1}$$

返回的结构体S, 可用作polyfit的输入来获取误差估计值。

返回的mu, 是一个二元素向量, 包含中心化值和缩放值。mu(1)是mean(x), mu(2)是std(x)。使用这些值时, polyfit 将x的中心置于零值处并缩放为具有单位标准差:  $\tilde{x} = (x - \bar{x}) / \sigma_x$  这种中心化和缩放变换可同时改善多项式和拟合算法的数值属性

## 2.4.2 polyval 函数

常用函数用法：

$$y = polyval(p, x)$$

$$[y, delta] = polyval(p, x, S)$$

$$y = polyval(p, x, [], mu)$$

$$[y, delta] = polyval(p, x, S, mu)$$

delta是使用p(x)预测x处的未来观测值时的标准误差估计值。

## 2.5 拟合优度分析原理

### 1. SSE(残差平方和): the Sum of Square due to Error

$$SS_{res} = \sum_{i=1}^n (y_i - \hat{y})^2 = e_i^2$$

该参数越接近于0,表示曲线拟合越成功

### 2. RMSE(均方根误差、标准差): Root mean squared error

$$RMSE = \sqrt{MSE} = \sqrt{\frac{SSE}{n}} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}$$

每个误差对 RMSE 的影响与平方误差的大小成正比; 因此较大的误差对 RMSE 的影响非常大,所以 RMSE 对异常值很敏感。

**MSE(均方差、方差):** Mean squared error

数理统计中均方误差是指参数估计值与参数值之差平方的期望值, 记为MSE。MSE是衡量“平均误差”的一种较方便的方法, MSE可以评价数据的变化程度, MSE的值越小, 说明预测模型描述实验数据具有更好的精确度。预测数据和原始数据对应点误差平方和的均值

$$MSE = \frac{SSE}{n}$$

### 3. R-square( $R^2$ )(决定系数):Coefficient of determination

$R^2$  即判定系数, 也称为拟合优度 // 区分于相关系数r和 $\rho_{xy}$

拟合优度越大, 自变量对因变量的解释程度越高, 自变量引起的变动占总变动的百分比就越高。观察点在回归直线附近越密集。取值范围: 0-1.

$$R_2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2}$$

$$SS_{reg} = \sum_{i=1}^n (\hat{y} - \bar{y})^2$$

$$SS_{res} = \sum_{i=1}^n (y_i - \hat{y})^2 = e_i^2$$

$$SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2$$

回归平方和: SSR(Sum of Squares for regression) = ESS (explained sum of squares)

残差平方和: SSE(Sum of Squares for Error) = RSS(residual sum of squares)

总离差平方和: SST(Sum of Squares for total) = TSS(total sum of squares)

SSE+SSR=SST

4.  $R^2_{adj}$  (校正决定系数): adjust R-square

$R^2$  评价拟合模型的好坏具有一定的局限性，即使向模型中增加的变量没有统计学意义， $R^2$  值仍会增大。因此需对其进行校正，从而形成了校正的决定系数(Adj  $R^2$ )。与  $R^2$  不同的是，当模型中增加的变量没有统计学意义时，Adj  $R^2$  会减小，因此 Adj  $R^2$  是衡量所建模型好坏的重要指标之一，Adj  $R^2$  越大，模型拟合得越好。

校正决定系数 (Adj  $R^2$ ) 引入了样本数量和特征数量，公式如下：

$$R^2_{adj} = 1 - \frac{(1-R^2)(n-1)}{n-p-1}$$

### 三、实验内容

1. 以中美两国1980年至2016年的GDP历史数据为基础，用多项式拟合进行曲线拟合，确定其数学模型，并给出拟合过程及分析。
2. 用最佳的多项式拟合模型进行最小二乘拟合，并观察模型误差。
3. 用拟合模型预测2017年至2020年中、美GDP，并计算模型的均方误差MSE和拟合优度 $R^2$ 。

- 数据：

(1) 中国1980年至2016年GDP，单位：万亿美元。

```

1 CH=[305350,290724,286729,307683,316666,... %1980~1984
2
3 312616,303340,330303,411923,461066,...
4
5 398623,415604,495671,623054,566471,... %1990~1994
6
7 736870,867224,965320,1032576,1097133,...
8
9 1214912,1344097,1477483,1671072,1966223,... %2000~2004
10
11 2308786,2774308,3571451,4604285,5121681,...
12
13 6066351,7522103,8570348,9635025,10534526,... %2010~2014
14
15 11226186,11232108].'*1e-6; %2015~2016

```

(2) 美国1980年至2016年GDP，单位：万亿美元。

```

1 US=[2862475,3210950,3345000,3638125,4040700,... %1980~1984
2
3 4346750,4590125,4870225,5252625,5657700,...
4
5 5979575,6174050,6539300,6878700,7308775,... %1990~1994
6
7 7664050,8100175,8608525,9089150,9660625,...
8

```

```

9      10284750,10621825,10977525,11510675,12274925,... %2000~2004
10
11      13093700,13855900,14477625,14718575,14418725,...
12
13      14964400,15517925,16155250,16691500,17393100,... %2010~2014
14
15      18120700,18624450].'*1e-6;

```

## 四、分析过程及部分代码

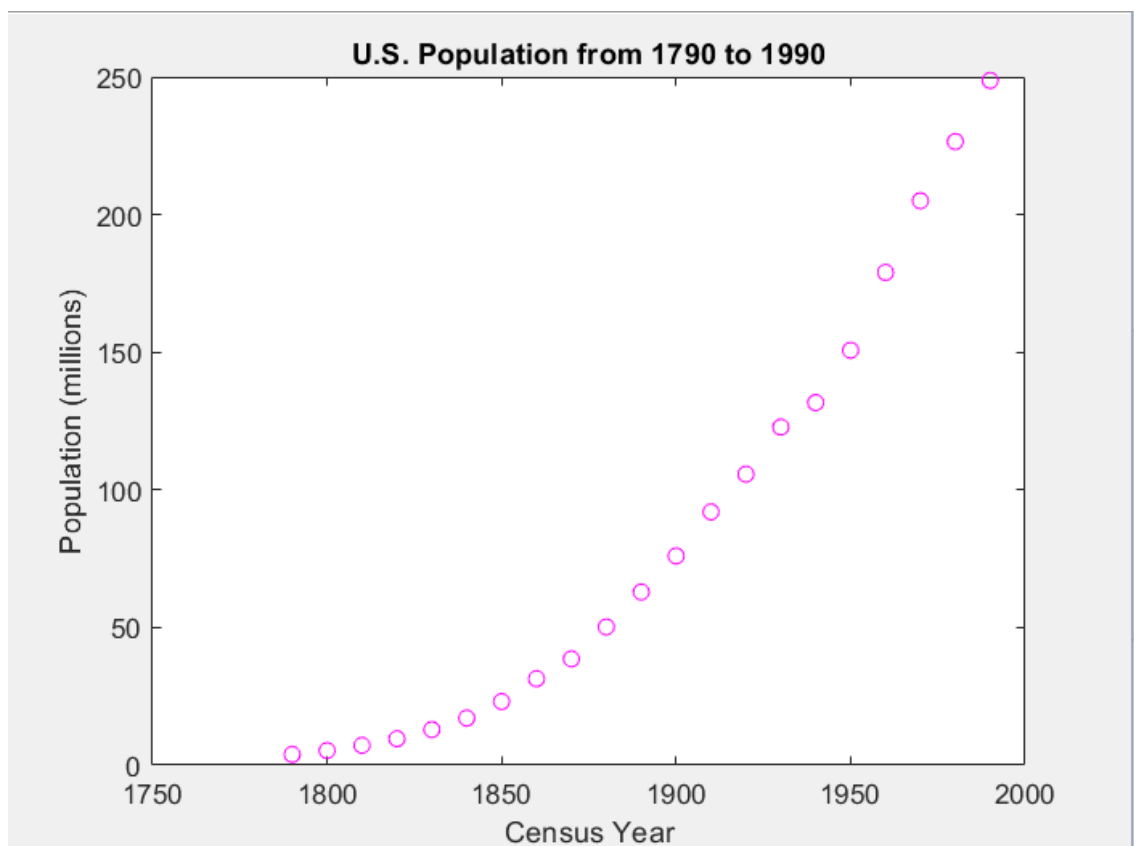
### 1. 相关性分析

- 散点图绘制

```

1 plot(cdate,pop,'om')
2 title('U.S. Population from 1790 to 1990');
3 xlabel('Census Year');
4 ylabel('Population (millions)');

```



- 计算协方差矩阵

```

1 >> corrcoef(cdate,pop)
2
3      ans =
4
5      1.0000    0.9597
6      0.9597    1.0000

```

- 结论

- 从图像上可以直观地看出原始样本数据具有较强的相关性

2. 协方差矩阵系数都接近于1, 进一步证明数据的强相关性

## 2. 多项式拟合

### • 原理

#### ◦ [Feature scaling](#)(特征缩放)--Standardization(标准化)

- 对不同特征维度的伸缩变换的目的是使得不同度量之间的特征具有可比性,对目标函数的影响体现在几何分布上, 同时不改变原始数据的分布。
- 特征标准化使数据中每个特征的值具有零均值 (当减去分子中的均值时) 和单位方差。该方法广泛用于许多机器学习算法 (例如, [支持向量机](#)、[逻辑回归](#)和[人工神经网络](#)) 中的归一化。

$$X^* = \frac{X - \bar{X}}{\sigma}$$

- ```
1 (cdate-mean(cdate))./std(cdate);  
2 //代码演示
```

- 工具箱内勾选"Center and Scale"

#### ◦ 通过Curve Fitting Tool==工具箱绘制拟合图像和残差图

### • 部分代码和图像

#### ◦ 初始化数据

```
1 %% Initialization.  
2  
3 % Initialize arrays to store fits and goodness-of-fit.  
4 fitresult = cell( 2, 1 );  
5 gof = struct( 'sse', cell( 2, 1 ), ...  
6   'rsquare', [], 'dfe', [], 'adjrsquare', [], 'rmse', [] );  
7
```

#### ◦ 拟合曲线(注:这里只贴出poly2的源码)

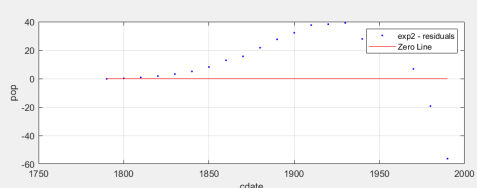
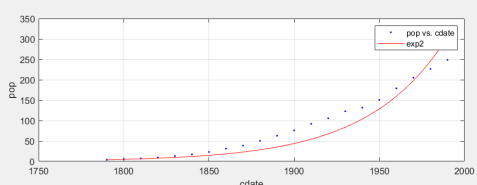
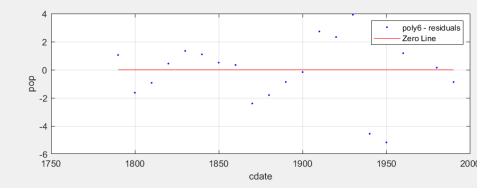
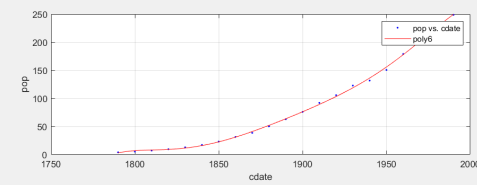
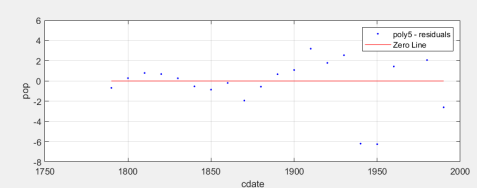
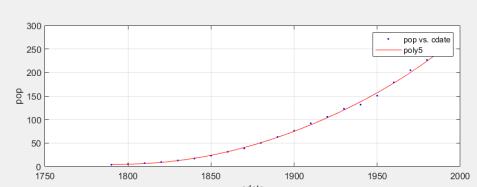
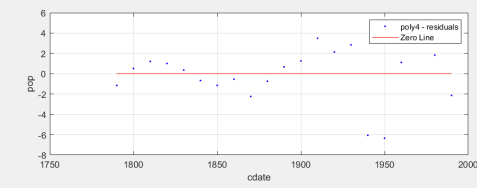
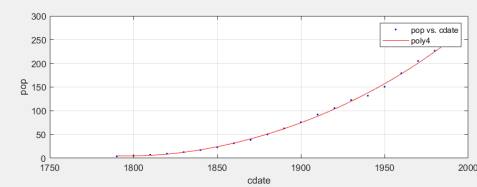
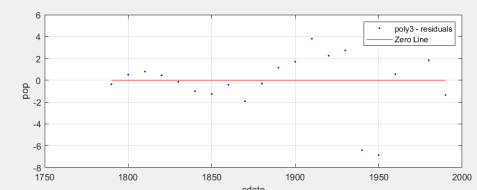
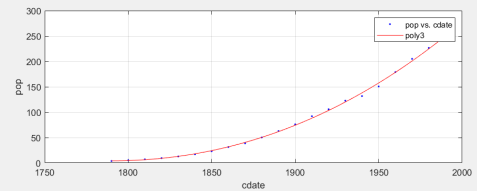
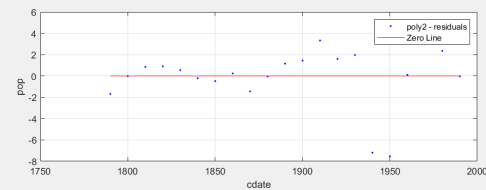
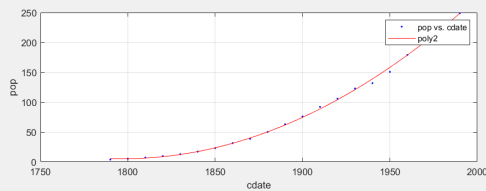
```
1 %% Fit: 'poly2'.  
2 [xData, yData] = prepareCurveData( cdate, pop );  
3  
4 % Set up fittype and options.  
5 ft = fittype( 'poly2' );  
6  
7 % Fit model to data.  
8 [fitresult{1}, gof(1)] = fit( xData, yData, ft, 'Normalize', 'on' );  
9  
10 % Create a figure for the plots.  
11 figure( 'Name', 'poly2' );  
12  
13 % Plot fit with data.  
14 subplot( 2, 1, 1 );  
15 h = plot( fitresult{1}, xData, yData );  
16 legend( h, 'pop vs. cdate', 'poly2', 'Location', 'NorthEast' );
```

```

17 % Label axes
18 xlabel cdate
19 ylabel pop
20 grid on
21
22 % Plot residuals.
23 subplot( 2, 1, 2 );
24 h = plot( fitresult{1}, xData, yData, 'residuals' );
25 legend( h, 'poly2 - residuals', 'Zero Line', 'Location',
26         'NorthEast' );
27
28 % Label axes
29 xlabel cdate
30 ylabel pop
31 grid on

```

○ poly2~poly6,exp2的拟合曲线及残差图

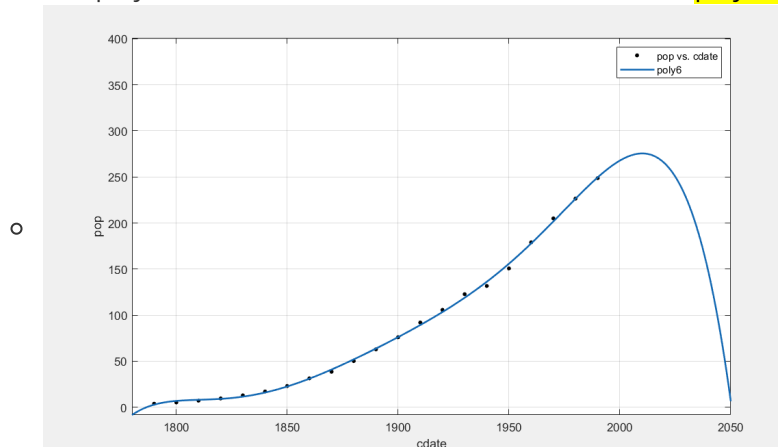




### 3. 拟合优度分析

- 拟合人口普查数据的目的是预测出未来人口数

- 对于poly6来说,数据明显超出了我们想要得到的结果,因此poly6舍去

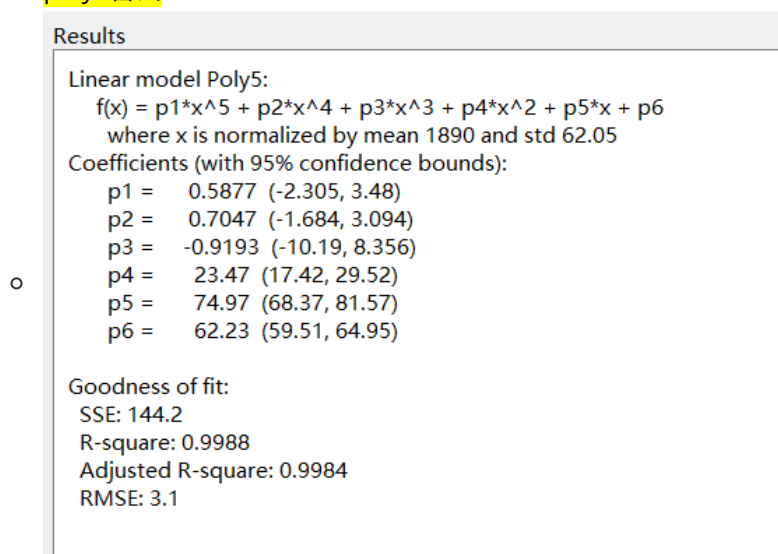


- 注: 其余拟合图像均有着合理的预测结果,在此不再赘述.

- 过拟合判断、

得到拟合结果后, 若计算出来的最高项系数过0(Zero Crossing)并且在0附近,则表明这个系数对于真实的多项式拟合没有任何的帮助, 即发生了过拟合(overfitting)。

- 观察到poly5的数据p1,p2,p3与0非常接近的且p3<0,证明poly5拟合曲线发生了过拟合,故poly5舍去



- 注: 其余拟合参数均无上述情况,在此不再赘述.

- 工具箱计算部分数据如下表:

| Fit Name | SSE | RMSE | R-square | DFE | Adj R-sq |
|----------|-----|------|----------|-----|----------|
|----------|-----|------|----------|-----|----------|

| Fit Name | SSE      | RMSE  | R-square | DFE | Adj R-sq |
|----------|----------|-------|----------|-----|----------|
| poly6    | 106.9276 | 2.764 | 0.99913  | 14  | 0.998764 |
| poly5    | 144.1661 | 3.100 | 0.99883  | 15  | 0.998444 |
| poly4    | 145.9689 | 3.020 | 0.99882  | 16  | 0.998523 |
| poly3    | 149.7687 | 2.968 | 0.99879  | 17  | 0.998574 |
| poly2    | 159.0293 | 2.972 | 0.99871  | 18  | 0.99857  |
| exp2     | 475.9491 | 5.291 | 0.99615  | 17  | 0.995468 |

- 由表可得
  - SSE最大的为exp2,拟合效果**极差**,exp2舍去
  - 对比剩余曲线,其 $R^2$  和 Adj  $R^2$  近似相等,故对比RMSE

| Fit Name | SSE      | RMSE  | R-square | DFE | Adj R-sq |
|----------|----------|-------|----------|-----|----------|
| poly3    | 149.7687 | 2.968 | 0.99879  | 17  | 0.998574 |
| poly2    | 159.0293 | 2.972 | 0.99871  | 18  | 0.99857  |
| poly4    | 145.9689 | 3.020 | 0.99882  | 16  | 0.998523 |

- 显然poly3>poly2>poly4,poly2和poly4舍去

## 4. 数学模型

由以上分析，拟合次数为3时多项式系数分别为 0.921，25.183，73.859，61.744可得到最终的三阶多项式拟合的美国人口增长数学模型

$$f(x) = 0.921x^3 + 25.183x^2 + 73.859x + 61.744$$

## 五、实验小结

不得不说，curve fit 工具箱相对于编程来说真的是方便不少。现如今，已经有大量的复杂繁琐的工作被工具箱所封装了起来，在app中都可以找到。不过，在经历了实验二的app制作后，得到一个小发现，我们实际操作中，有些app封装的无论有多好，我们总会有无数的细节需要实现，这个时候，简单的工具箱就不能满足我们了，我们不可能保证工具箱容纳所有的操作，工具箱也不可能因此就变得笨拙，所以这就需要我们用编程来实现了。

所以说，我们在工具箱和代码的过程中，应当学会做一定的取舍，根据我们所需要实现的功能来一一进行抉择。

# 实验二 基于GUI的高尔夫手推车生产销售程序设计

## 一、实验目的

- 1. 掌握在MATLAB中创建图形用户界面（GUI）的方法和应用；
- 2. 掌握回调函数的编写和应用；
- 3. 界面控件的合理设计。

## 二、部分实验控件介绍

- 常用控件

常用控件的作用如表1所示。

表1控件的功能

| 控件名   | PropertyName | 功能                                              |
|-------|--------------|-------------------------------------------------|
| 按钮    | PushButton   | 最常用的控件，用于响应用户的鼠标单击，按钮上有说明文字说明其作用。               |
| 切换按钮  | ToggleButton | 当单击时会凹凸状态切换。                                    |
| 单选按钮  | RadioButton  | 当单击时会用黑白点切换，总是成组出现，多个单选按钮互斥，一组中只有一个被选中。         |
| 复选框   | CheckBox     | 当单击时会用√切换，有选中、不选中 and 不确定等状态，总是成组出现，多个复选框可同时选用。 |
| 文本框   | EditText     | 凹形方框，可随意输入和编辑单行和多行文字，并显示出来。                     |
| 静态文本框 | StaticText   | 用于显示文字信息，但不接受输入。                                |
| 滚动条   | Slider       | 可以用图示的方式显示在一个范围内数值的大概值范围，用户可以移动滚动条改变数值。         |
| 框架    | Frame        | 将一组控件围在框架中，用于装饰界面。                              |
| 列表框   | ListBox      | 显示下拉文字列表，用户可以从列表中选择一项和多项。                       |
| 弹出式菜单 | PopupMenu    | 相当于文本框和列表框的组合，用户可以从下拉列表中选择。                     |
| 坐标轴   | Axes         | 用于绘制坐标轴。                                        |

- 控件的创建

在可视化界面环境中创建控件

在可视化界面环境中创建控件很简单，就是在图形对象面板中选中控件，然后在空白的界面编辑面板中拖放即可，如图1所示为各种控件的显示。

### 三、实验设备

装载MATLAB的计算机

### 四、实验内容

某公司生产和销售高尔夫手推车。每周周末，公司都将那一周所生产的手推车转移到仓库（库存）之中。卖出的所有手推车都是从库存中提取。这个过程的一个简单模型为： $I(k+1)=P(k)+I(k)-S(k)$

其中， $P(k)$ 为第 $K$ 周所生产的手推车数量； $I(k)$ 为第 $K$ 周库存中手推车数量； $S(k)$ 为第 $K$ 周所卖出的手推车数量。

| 周   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
|-----|----|----|----|----|----|----|----|----|----|----|
| 销售量 | 50 | 55 | 60 | 70 | 70 | 75 | 80 | 80 | 90 | 55 |

表1为10周计划中的每周销售额。假设每周的产量都基于前一周的销售量，所以有 $P(k)=S(k-1)$ 。假设第一周的产量为50辆手推车，即 $P(1)=50$ 。

编写一个MATLAB的GUI程序，要求：

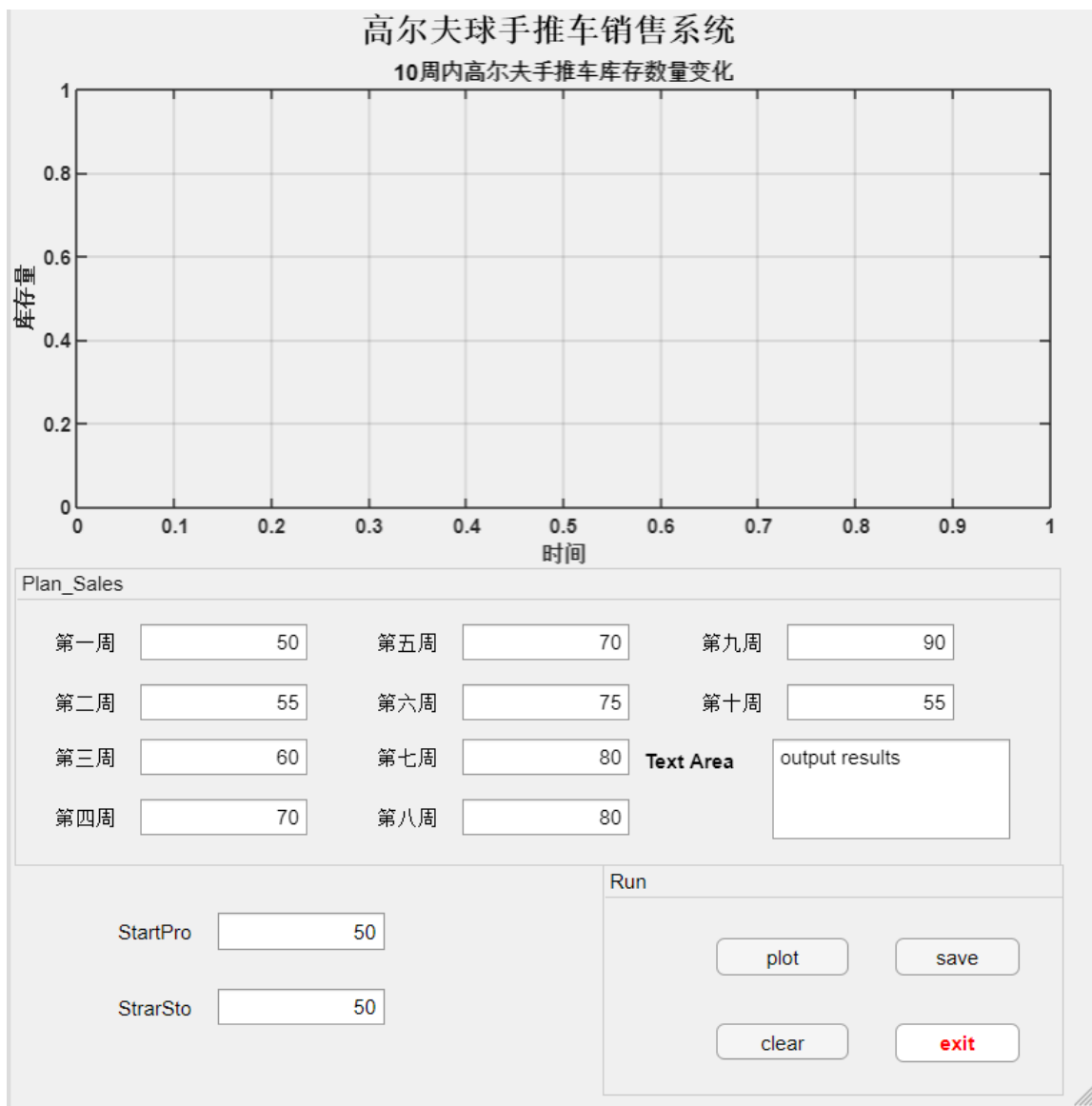
- （1）应能够输入：初始库存数量；初始生产量；10周计划销售量。
- （2）应能够输出：第 $K$ 周的库存数量；10周之内库存数量减少到0为止的时间，并同时绘制反映库存数量减少到0过程的图形。
- （3）将表1数据保存到数据文件xs.mat中。

在MATLAB的GUI下建立一个名为xiaoshou(包括xiaoshou.fig和xiaoshou.m两个文件)的图形用户界面应用程序，其标题为“高尔夫手推车生产销售系统”；界面自行设计，详细说明设计过程。

### 五、实验设计过程

#### 5.1 界面设计

利用APP Designer编辑器设计如下画面



## 5.2 回调函数设计

### 1. 考虑所需要的全局变量 properties

```

1      properties (Access = private)
2          start_producty;
3          start_storey;
4          Plan_sales = ones(10,1);
5          stores = ones(10,1);
6          produce = ones(10,1);
7          week = [1;2;3;4;5;6;7;8;9;10];
8          figure;
9          cellArrayText = '';
10         % Description
11     end

```

### 2. plot按钮设计

```

1      % Button pushed function: plotButton
2      function plotButtonPushed(app, event)
3          app.Plan_sales(1) = app.sales_1.Value;
4          app.Plan_sales(2) = app.sales_2.Value;
5          app.Plan_sales(3) = app.sales_3.Value;

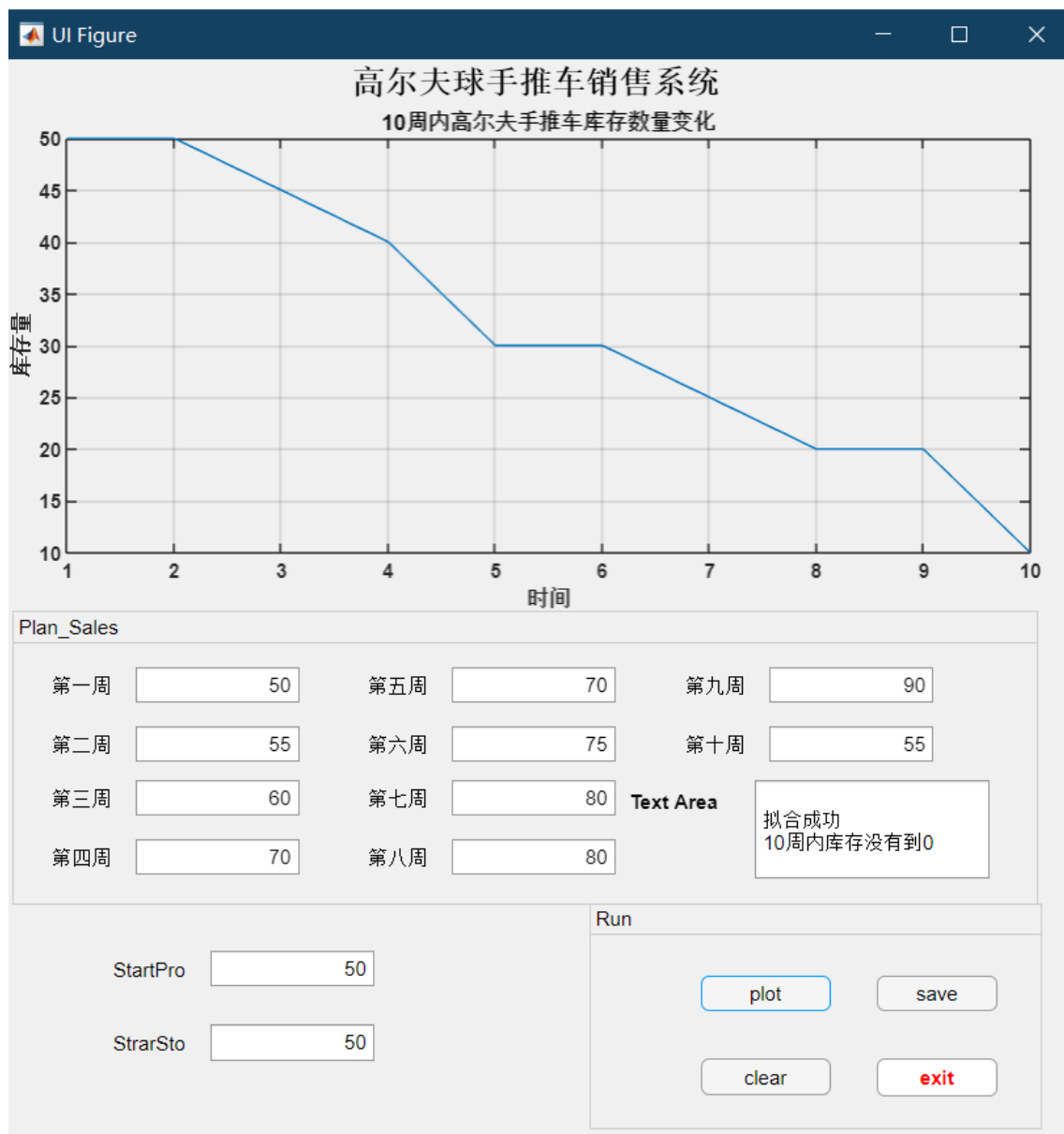
```

```

6         app.Plan_sales(4) = app.sales_4.Value;
7         app.Plan_sales(5) = app.sales_5.Value;
8         app.Plan_sales(6) = app.sales_6.Value;
9         app.Plan_sales(7) = app.sales_7.Value;
10        app.Plan_sales(8) = app.sales_8.Value;
11        app.Plan_sales(9) = app.sales_9.Value;
12        app.Plan_sales(10) = app.sales_10.Value;
13        app.produce(1) = app.StartProEditField.Value;
14        app.stores(1) = app.StrarStoEditField.Value;
15        flag= 1;
16
17        t = 100;
18        for i=1:10
19            if i>1
20                app.produce(i)=app.Plan_sales(i-1);
21                app.stores(i) = app.stores(i-1) + app.produce(i-1) -
app.Plan_sales(i-1);
22                if app.stores(i) < 0
23                    app.stores(i) = 0;
24                end
25            end
26            if (app.stores(i)==0)&&(flag== 1)
27                t = i;
28                flag = 0;
29            end
30        end
31
32        if (t~=100)
33            text = sprintf('%s%d%s', '第', t, '周库存减少到0');
34        else
35            text = sprintf('%s', '10周内库存没有到0');
36        end
37        app.cellArrayText = strcat(app.cellArrayText,10,'拟合成功');
38        app.cellArrayText = strcat(app.cellArrayText,10,text);
39        app.TextArea.Value = app.cellArrayText;
40        app.figure = plot(app.UIAxes,app.week,app.stores);
41
42    end
43

```

输入数据后，拟合出结果如下

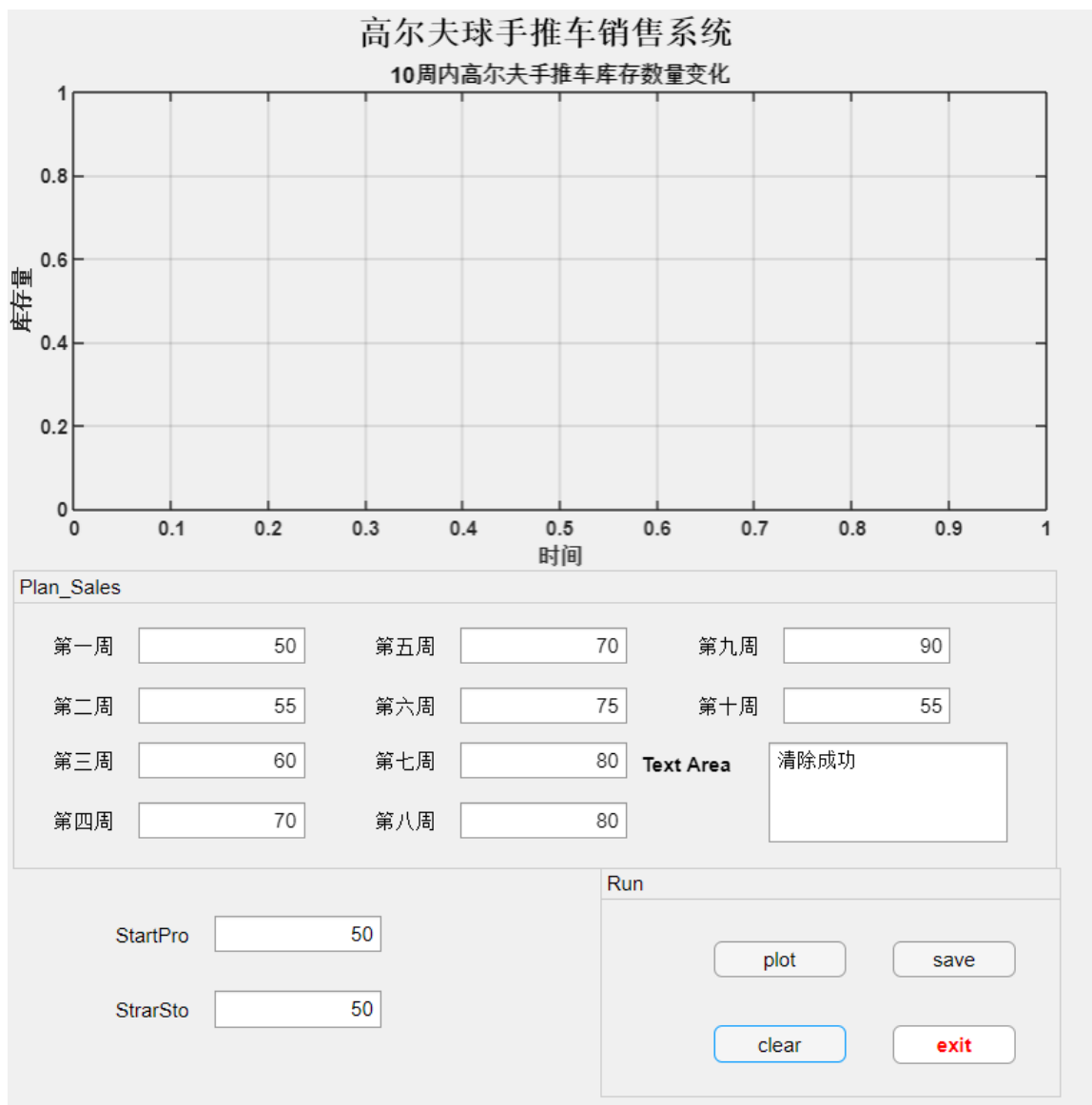


### 3. clear按钮设计

```

1 % Button pushed function: clearButton
2 function clearButtonPushed(app, event)
3     app.cellArrayText = '';
4     app.TextArea.Value = '清除成功';
5     delete(app.figure);
6 end

```



#### 4. save按钮设计

```

1      % Button pushed function: saveButton
2      function saveButtonPushed(app, event)
3          sales = app.Plan_sales;
4          save ('xs.mat','sales');
5
6          app.cellArrayText = strcat(app.cellArrayText,10,'保存成功');
7          app.TextArea.Value = app.cellArrayText;
8      end

```

#### 5. exit按钮设计



```

1      % Button pushed function: exitButton
2      function exitButtonPushed(app, event)
3          selection = uiconfirm(app.UIFigure, 'Close APP?', 'Quit',
4              'Icon', 'warning');
5          switch selection
6              case 'OK'
7                  app.delete();
8              case 'Cancel'
9                  return
10             end
11         end

```



## 六、实验小结

经历本次实验后，不得不说终于明白了原来我一直利用的Matlab工具箱竟然是这样封装起来的。

在App Designer的设计过程中，功能模块的摆放，回调函数的设计，每一步都是环环相扣的，哪里出错了，后续就会再次报出这个这个错误。实验后，真的越来越佩服那些制作app来方便我们的人了。他们耗费的大量精力，才有了现在的Matlab，我们真的要珍惜现有的前人智慧的结晶。

# 实验三 微分方程的Simulink求解及子系统封装

## 一、实验目的

1. 熟悉MATLAB中关于常微分方程和微分代数方程求解的各种方法；
2. 对状态方程的求解，分别会用脚本仿真和用Simulink仿真；
3. 对给定的微分代数方程分别会用脚本仿真和用Simulink仿真；
4. 熟悉在Simulink下进行子系统封装及仿真。

## 二、实验设备

装载MATLAB的计算机

## 三、实验内容

### (一) Lorenz模型的状态方程表示为：

$$\begin{cases} \frac{dx_1(t)}{dt} = -8x_1(t)/3 + x_2(t)x_3(t) \\ \frac{dx_2(t)}{dt} = -10x_2(t) + 10x_3(t) \\ \frac{dx_3(t)}{dt} = -x_1(t)x_2(t) + 28x_2(t) - x_3(t) \end{cases}$$

初值为  $x_1(0) = x_2(0) = 0, x_3(0) = 10^{-10}$

要求：

1. 建立脚本，采用四阶五级的(Runge-Kutta-Felhberg, RKF)方法，调用ode45()方法求解该方程；
2. 在Simulink下完成下列要求：
  - (1) 在Simulink下为上述方程组建立仿真模型，保存在st1.mdl中；
  - (2) 仿真时间tout初值为0，终值根据实际情况而定，在图形窗口中设置4个子窗口，将 和相三维图分别绘在其中：
- a) 给该图形窗口增加标题“Lorenz模型的状态方程仿真图形”；
- b) 给每个子窗口加坐标轴名称。
- (3) 将tout 中的数据分别保存到变量t, x1, x2及x3中，将这四个变量保存到st1\_data.mat数据文件中。
- (4) 说明上述要求的实现过程。

## (二) 微分代数方程仿真：

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ 0 \end{bmatrix} = \begin{bmatrix} -0.2x_1 + x_2x_3 + 0.3x_1x_2 \\ 2x_1x_2 - 5x_2x_3 - 2x_2^2 \\ x_1 + x_2 + x_3 - 1 \end{bmatrix}$$

初始条件为 $x_1(0) = 0.8, x_2(0) = x_3(0) = 0.1$

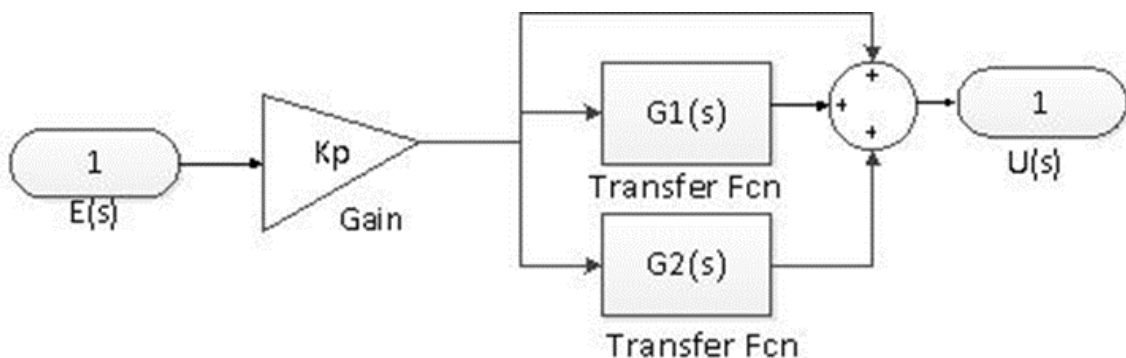
要求：

1. 建立脚本，采用刚性方程求解算法ode15()函数求解该方程；
2. 在Simulink下完成下列要求
  - 1) 在Simulink下为上述一阶微分方程组建立仿真模型，保存在st1.mdl中；
  - 2) 利用1) 中的仿真模型，在Simulink下调用ode15()方法进行仿真求解 $x_1(t)$ ,  $x_2(t)$ 和 $x_3(t)$ ，仿真时间tout初值为0，终值为20，在图形窗口中设置2×2四个子窗口，将 $x_1(t)$ ,  $x_2(t)$ ,  $x_3(t)$ 和 $x(t)$ (包含 $x_1(t)$ ,  $x_2(t)$ 和 $x_3(t)$ 三个变量)，分别绘在四个子窗口中，并包括：
    - a) 给该图形窗口增加标题“微分代数方程仿真图形”；
    - b) 给每个子窗口加坐标轴名称。
  - 3) 将tout,  $x_1(t)$ ,  $x_2(t)$ 和 $x_3(t)$ 中的数据分别保存到变量t, y1, y2和y3中，将这四个变量保存到st1\_data.mat数据文件中。
3. 详细说明上述要求的实现过程。

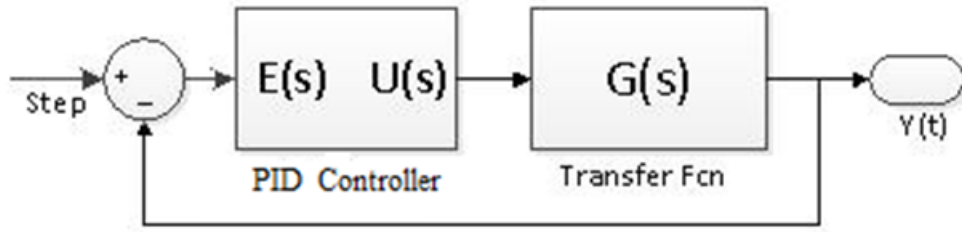
## (三) 子系统封装及仿真

(1) 将图1中的PID Controller模型图在Simulink下封装成子系统, 其中：

$$U(s) = K_p \left( 1 + \frac{1}{T_i s} + \frac{sT_d}{T_d s + 1} \right)$$



(2) 将 (1) 中的子系统用在图 2 所示的仿真模型（保存到st2.mdl）中，在Simulink下调用ode45()方法进行仿真求解单位阶跃响应 $y(t)$ （包括 $y(t)$ 响应曲线和tout在0-20秒内的 $y(t)$ 的值）。



$G(s) = \frac{1}{(s+1)(s+3)(s+5)} Kp = 11.520$ ,  $Ti = 0.658$ ,  $Td = 0.164$ , Step为单位阶跃函数。如果仿真输出超调量过大，希望调整以上参数以获取更好的动态性能。

## 四、实验过程

### 4.1 Lorenz模型的状态方程

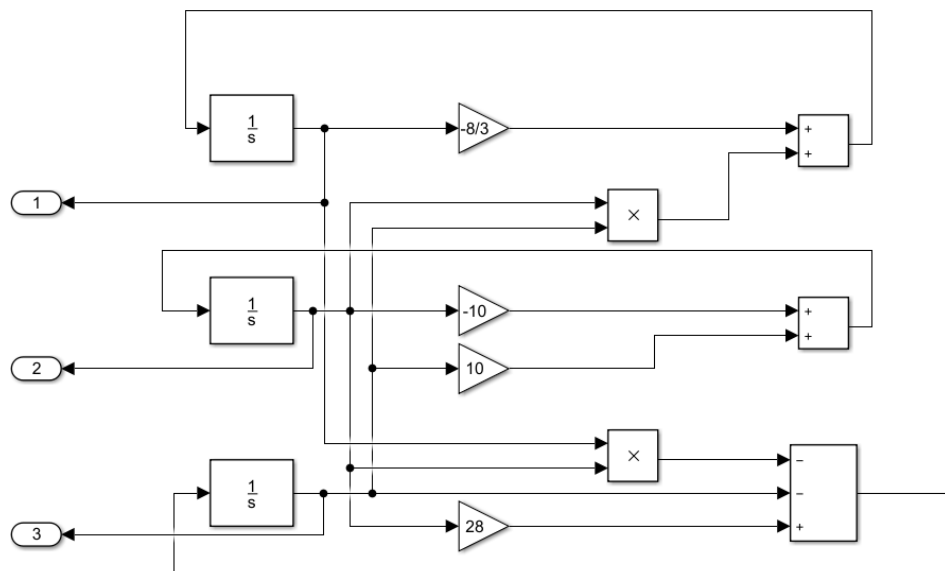
1. 建立脚本，采用四阶五级的(Runge-Kutta-Felhberg, RKF)方法，调用ode45()方法求解该方程；

```

1  f=@(t, x)[-8/3 * x(1) + x(2) * x(3); -10 * x(2) + 10 * x(3); -x(1) *
   x(2) + 28 * x(2) - x(3)];
2  t_final = 100;
3  x0 = [0; 0; 1e-10];
4
5  [t, x] = ode45(f, [0, t_final], x0);
6
7  color = ['r', 'g', 'b', 'y'];
8  indexs = ['x', 'y', 'z'];
9  for i = 1 : 3
10     subplot(2, 2, i);
11     plot(t, x(:, i), color(mod(i, 4) + 1));
12     box on
13     grid on
14     axis([t(1), t(end), min(x(:, i)), max(x(:, i))])
15     title(sprintf("Lorenz模型的状态方程仿真图形"));
16     xlabel(sprintf("Time t"));
17     ylabel(sprintf("Solution %s(t)", indexs(i)));
18 end
19 i = i + 1;
20 subplot(2, 2, i);
21 plot3(x(:, 1), x(:, 2), x(:, 3), 'color', [1 0.5 0]);
22 axis([10 40 -20 20 -20 20]);
23 title(sprintf("相三维图"));
24 sgtitle(sprintf("Lorenz模型的状态方程仿真图形"));
25

```

2. 建立Simulink仿真模型

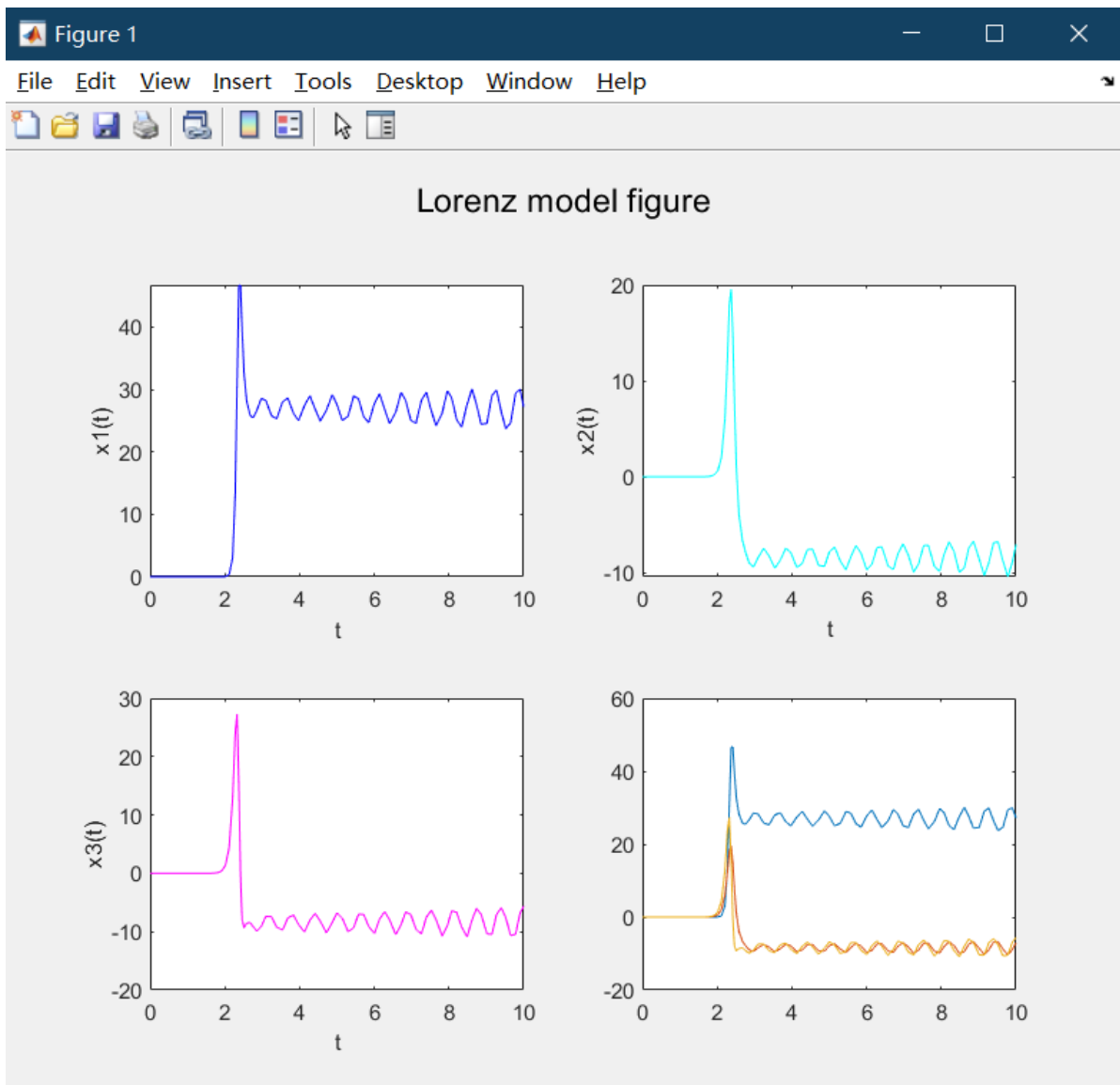


### 3. 实现图形的绘制

```

1  x1 = yout(:,1);
2  x2 = yout(:,2);
3  x3 = yout(:,3);
4  t = tout;
5
6  figure(1);
7
8  subplot(2,2,1);
9  plot(t,x1,'b');
10 xlabel("t");
11 ylabel("x1(t)")
12
13 subplot(2,2,2);
14 plot(t,x2,'c');
15 xlabel("t");
16 ylabel("x2(t)")
17
18 subplot(2,2,3);
19 plot(t,x3,'m');
20 xlabel("t");
21 ylabel("x3(t)")
22
23 subplot(2,2,4);
24 plot(t,yout)
25 subtitle("Lorenz model figure");
26
27 save stl_data.mat t x1 x2 x3
28

```



## 4.2 微分方程仿真

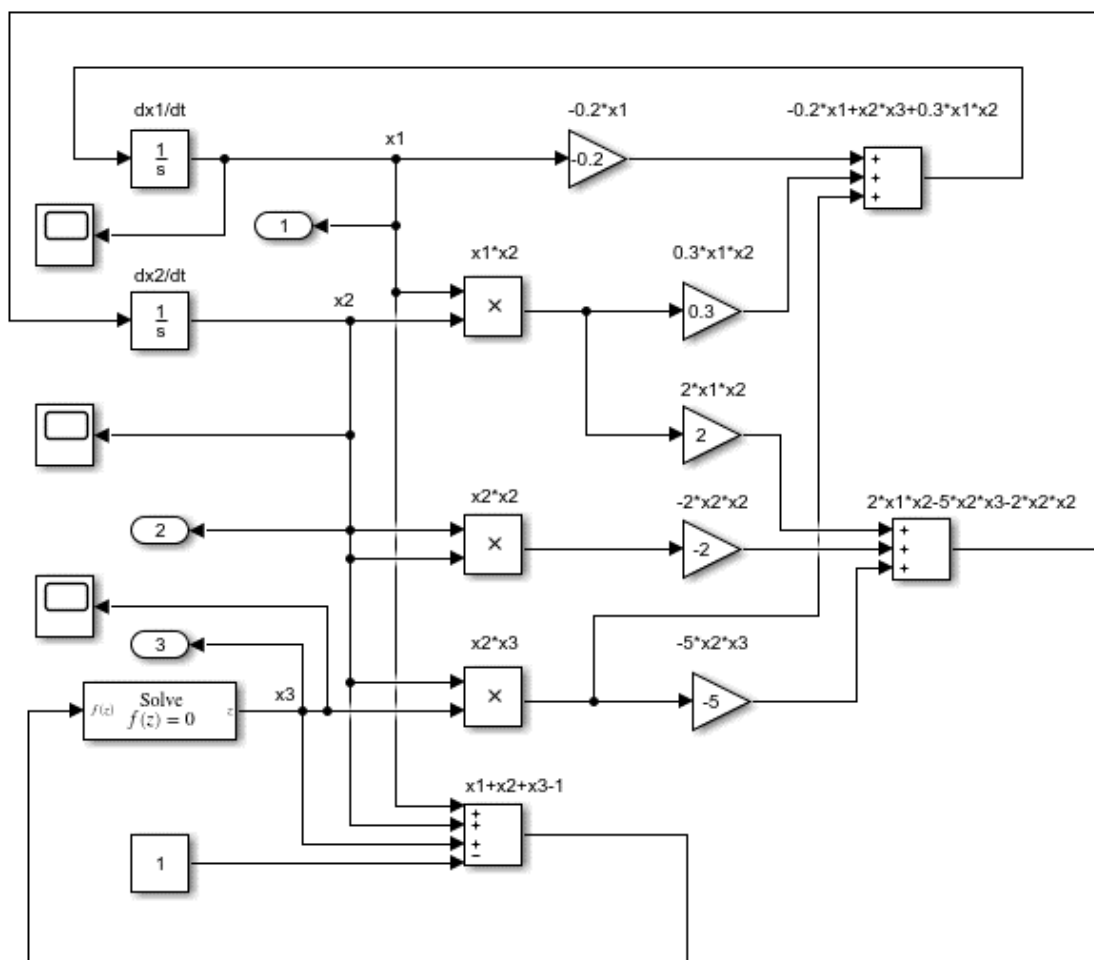
1. 建立脚本，采用刚性方程求解算法ode15s()函数求解该方程；

```

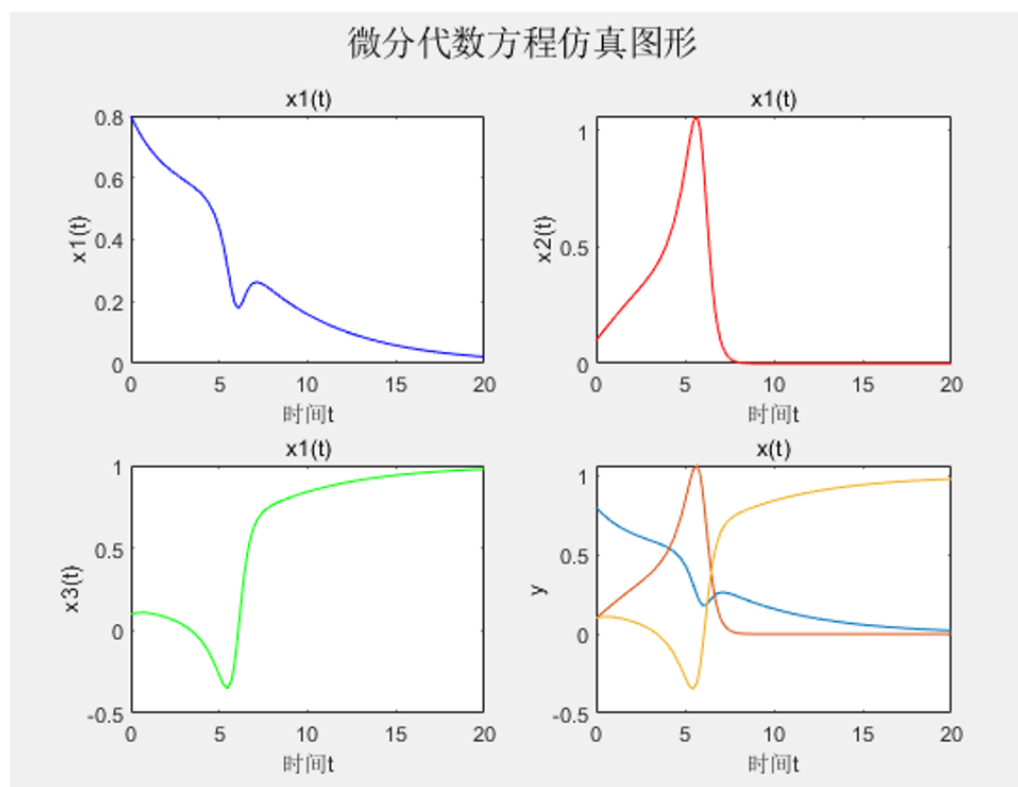
1 f=@(t, x)[-0.2 * x(1) + x(2) * x(3) + 0.3 * x(1) * x(2); 2 * x(1) * x(2) - 5
2 * x(2) * x(3) - 2 * x(2) * x(2); x(1) + x(2) + x(3) - 1];
2
3 M = [1, 0, 0; 0, 1, 0; 0, 0, 0];
4
5 options = odeset;
6 options.Mass = M;
7 x0 = [0.8, 0.1, 0.1];
8 [t, x] = ode15s(f, [0, 20], x0, options);
9

```

2. 建立Simulink仿真模型

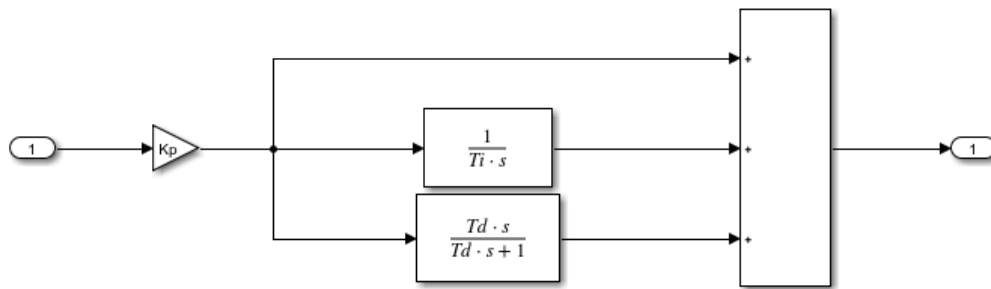


3. 代码与上题一样

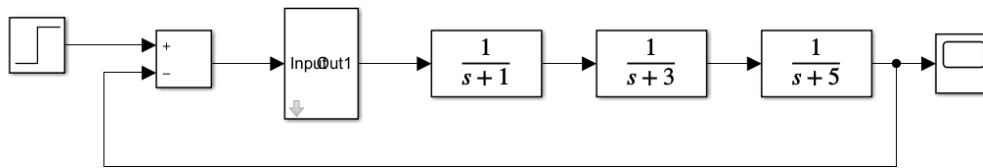


## 4.3 子系统封装及仿真

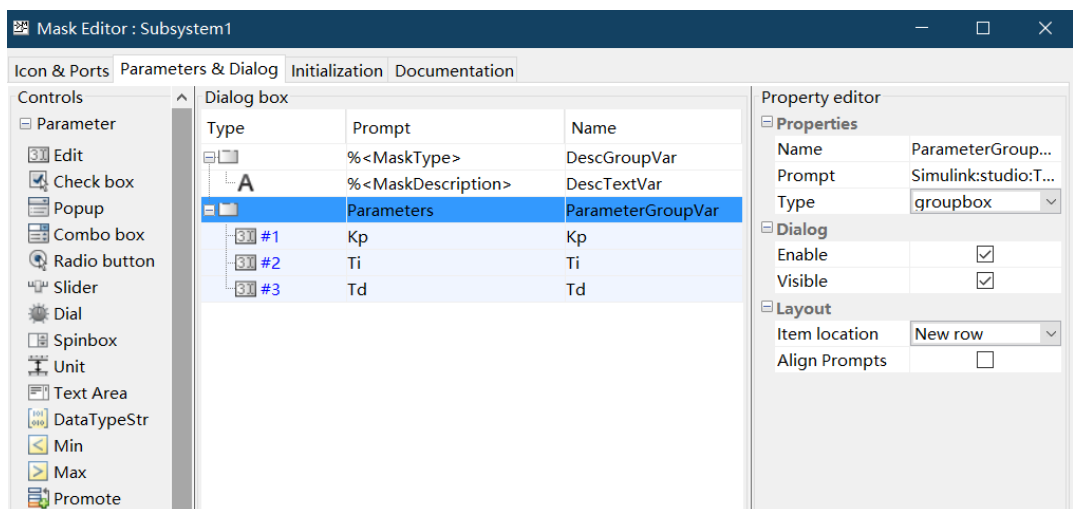
### 1. 绘制 PID Controller模型



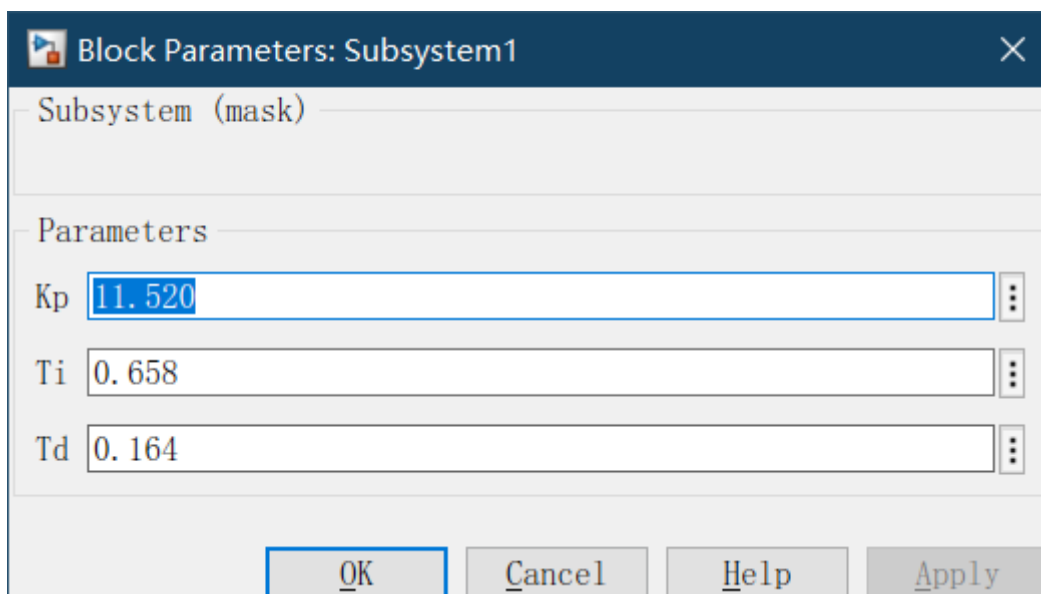
### 2. 绘制母系统



#### o mask



#### o 参数设置





### 3. 设置仿真模式ode45

Simulation time

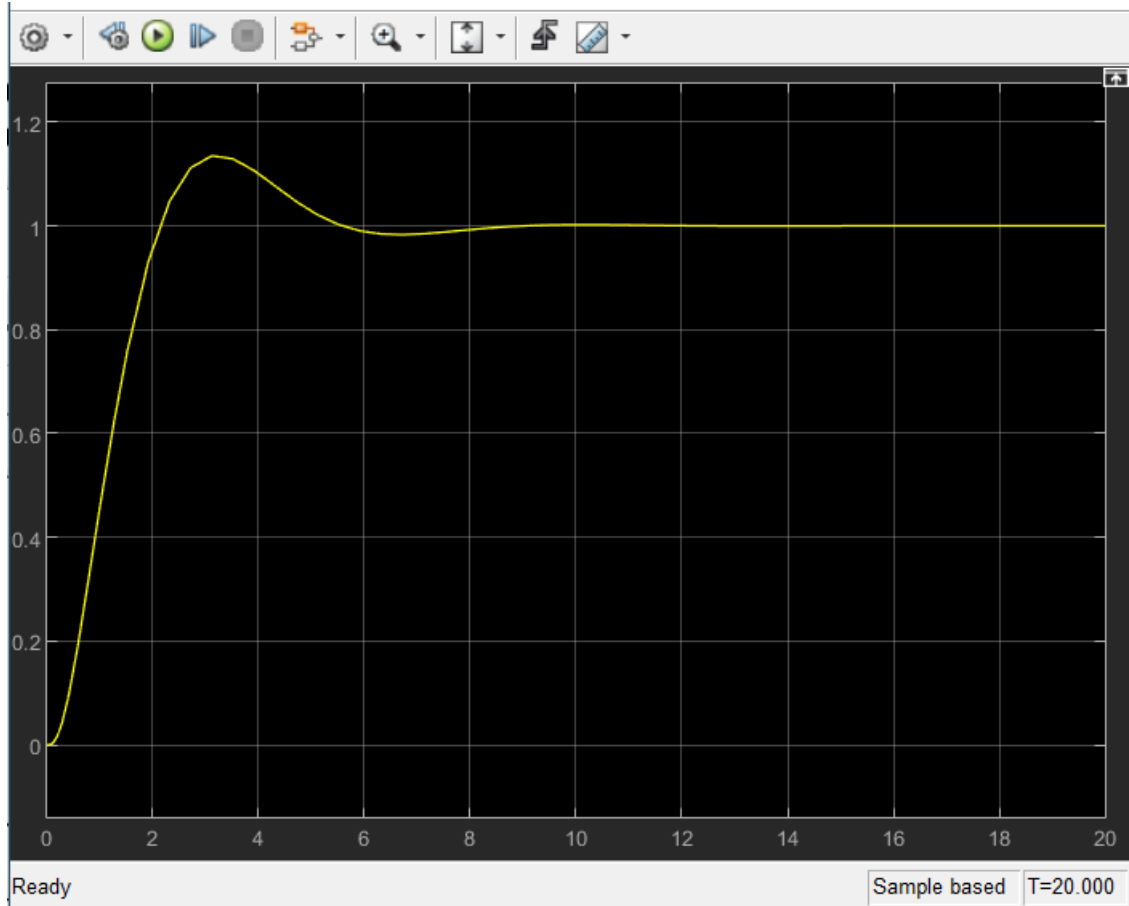
Start time:  Stop time:

Solver selection

Type:  Solver:

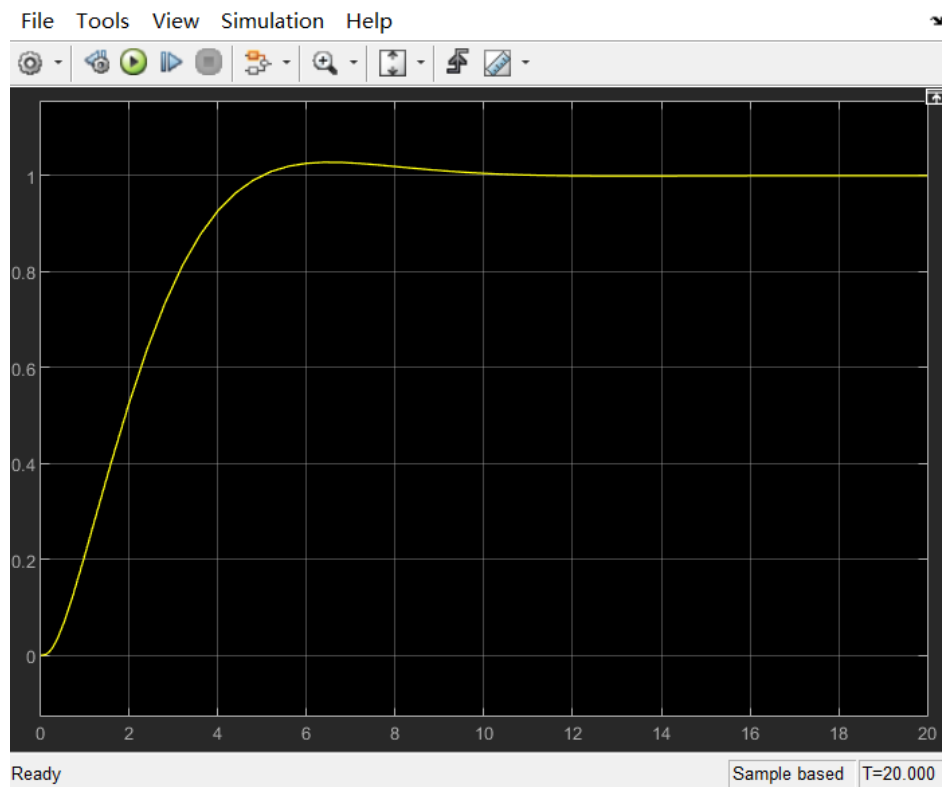
► Solver details

### 4. 得出图像



### 5. 调整参数

将Kp减小也就是增益减小



明显的观察到超调量减小，系统更加稳定

## 五、实验小结

Simulink建立仿真模型过程中，要注意微分和积分也就是s与s分之一的区别，留意引出点的不同。

Simulink是一种用于实现计算机仿真的软件工具，他是Matlab的一个附属组件，用来提供一个系统级的建模与动态仿真工作平台。用模块组合的方法为用户能够快速、准确地创建动态系统的计算机模型。

本次实验确实为我们提供了一个学习Simulink的机会，它将一些数字化的公式抽象为了图形，这是用编程语言明确地表达微分方程的仿真软件包无法比拟的。

# 实验四 二阶模糊逻辑控制系统仿真

## 一、实验目的

1. 熟悉在MATLAB下，模糊逻辑推理系统的创建、输入、输出变量设置及其论域、隶属度函数的选取设置、模糊逻辑规则的建立以及输入输出特性曲面浏览；
2. 熟悉利用模糊逻辑控制工具箱或利用模糊推理系统的基本函数来建立模糊逻辑推理系统；
3. 了解模糊推理系统在控制系统中的应用。

## 二、实验内容

假设某一工业过程可等效成以下二阶系统：

$$G(s) = \frac{20}{8s^2 + 6s + 1}$$

设计一个模糊控制器，使其能自动建立模糊规则库，保证控制规则如表1所示，这种规则可表示为：

$$U = \text{fix}\left(\frac{E+DE}{2}\right) = \text{fix}(\alpha E + (1 - \alpha)DE) \quad (1)$$

式(1)中，fix为取整函数；E为误差的模糊集；DE为误差导数的模糊集； $\alpha$ 为常数。

表1 模糊控制规则

|    | NB | NS | ZR | PS | PB |
|----|----|----|----|----|----|
| NB | PB | PB | PS | PS | ZR |
| NS | PB | PS | PS | ZR | ZR |
| ZR | PS | PS | ZR | ZR | NS |
| PS | PS | ZR | ZR | NS | NS |
| PB | ZR | ZR | NS | NS | NB |

这样表示的模糊控制系统可以通过改变 $\alpha$ 值方便地修改如表1所示的模糊控制规则，从而自动建立系统的模糊规则库。

设计模糊控制器，使其能自动建立模糊规则库，保证系统输出尽快跟随系统输入。

要求建立脚本实现上述模糊逻辑控制系统。采样时间T=0.01秒；系统输入r(t)=1.0。观察不同 $\alpha$ 取值时，阶跃响应的变化情况，找到匹配较佳动态性能的 $\alpha$ 值，并进行详细的实验分析。

## 三、实验过程及分析

1. 根据题意写出模糊控制相关代码

```

1  clear all;
2  num=20;den=[8 6 1];
3  [A,b,c,d]=tf2ss(num,den);
4  %% 系统参数
5  T=0.01;h=T;
6  N=500;R=1.0*ones(1,N);
7  uu=zeros(1,N); yy=zeros(3,N);
8  ka=1;
9  for alpha=[0.45 0.75 0.90];
10
11  %定义输入/输出变量及其隶属度函数
12
13  fisMat=newfis('n4');
14  fisMat=addvar(fisMat,'input','e',[-6,6]);
15  fisMat=addvar(fisMat,'input','de',[-6,6]);
16  fisMat=addvar(fisMat,'output','u',[-6,6]);
17  fisMat=addmf(fisMat,'input',1,'NB','trapmf',[-6 -6 -5 -3]);
18  fisMat=addmf(fisMat,'input',1,'NS','trapmf',[-5 -3 -2 0]);
19  fisMat=addmf(fisMat,'input',1,'ZR','trimf',[-2 0 2]);
20  fisMat=addmf(fisMat,'input',1,'PS','trapmf',[0 2 3 5]);
21  fisMat=addmf(fisMat,'input',1,'PB','trapmf',[3 5 6 6]);
22  fisMat=addmf(fisMat,'input',2,'NB','trapmf',[-6 -6 -5 -3]);
23  fisMat=addmf(fisMat,'input',2,'NS','trapmf',[-5 -3 -2 0]);
24  fisMat=addmf(fisMat,'input',2,'ZR','trimf',[-2 0 2]);
25  fisMat= addmf(fisMat,'input',2,'PS','trapmf',[0 2 3 5]);
26  fisMat=addmf(fisMat,'input',2,'PB','trapmf',[3 5 6 6]);
27  fisMat=addmf(fisMat,'output',1,'NB','trapmf',[-6 -6 -5 -3]);
28  fisMat=addmf(fisMat,'output',1,'NS','trapmf',[-5 -3 -2 0]);
29  fisMat=addmf(fisMat,'output',1,'ZR','trimf',[-2 0 2]);
30  fisMat=addmf(fisMat,'output',1,'PS','trapmf',[0 2 3 5]);
31  fisMat=addmf(fisMat,'output',1,'PB','trapmf',[3 5 6 6]);
32  %模糊规则矩阵
33  for i=1:5
34      for j=1:5
35          rr(i,j)=round(alpha*i+(1-alpha)*j);
36      end
37  end
38  rr=6-rr;
39  r1=zeros(prod(size(rr)),3);
40  k=1;
41  for i=1:size(rr,1)
42      for j=1:size(rr,2)
43          r1(k,:)= [i,j,rr(i,j)];
44          k=k+1;
45      end
46  end
47  [r,s]=size(r1);
48  r2=ones(r,2);
49  rulelist=[r1 r2];
50  fisMat=addrule(fisMat,rulelist);
51  %模糊控制系统仿真
52  Ke=30;Kd=0.2;
53  Ku=1.0;x=[0;0];
54  e=0;de=0;
55  for k=1:N

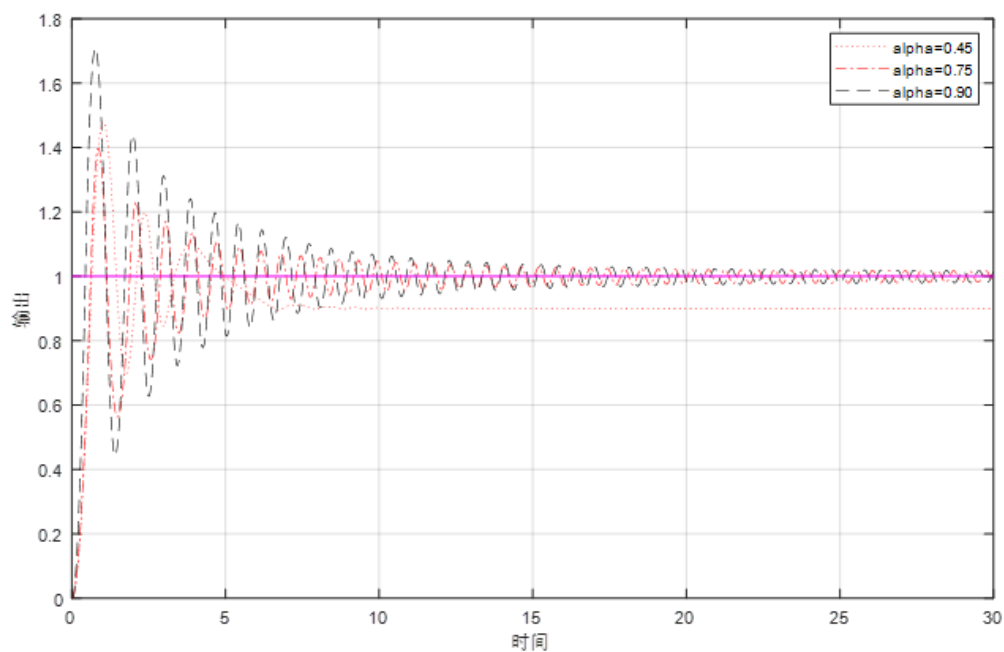
```

```

56         e1=Ke*e;
57         de1=Kd*de;
58 %将模糊控制器的输入变量变换到论域
59         if e1>=6
60             e1=6;
61         elseif e1<=-6
62             e1=-6;
63         end
64         if de1>=6;
65             de1=6;
66         elseif de1<=-6
67             de1=-6;
68         end
69
70         in=[e1 de1];
71         uu(1,k)=Ku*evalfis(in,fismat);
72         u=uu(1,k);
73
74         K1=A*x+b*u;
75         K2=A*(x+h*K1/2)+b*u;
76         K3=A*(x+h*K2/2)+b*u;
77         K4=A*(x+h*K3)+b*u;
78         x=x+(K1+2*K2+2*K3+K4)*h/6;
79         y=c*x+d*u;
80         yy(ka,k)=y;
81
82         e1=e;e=y-R(1,k);
83         de=(e-e1)/T;
84     end
85     ka=ka+1;
86 end
87
88 kk=[1:N]*T;
89 plot(kk,yy(1,:), 'r:', kk,yy(2,:), 'r-. ', kk,yy(3,:), 'k--', kk,R, 'm');
90 xlabel('时间');ylabel('输出');
91 legend('alpha=0.45', 'alpha=0.75', 'alpha=0.90');
92 grid on;
93

```

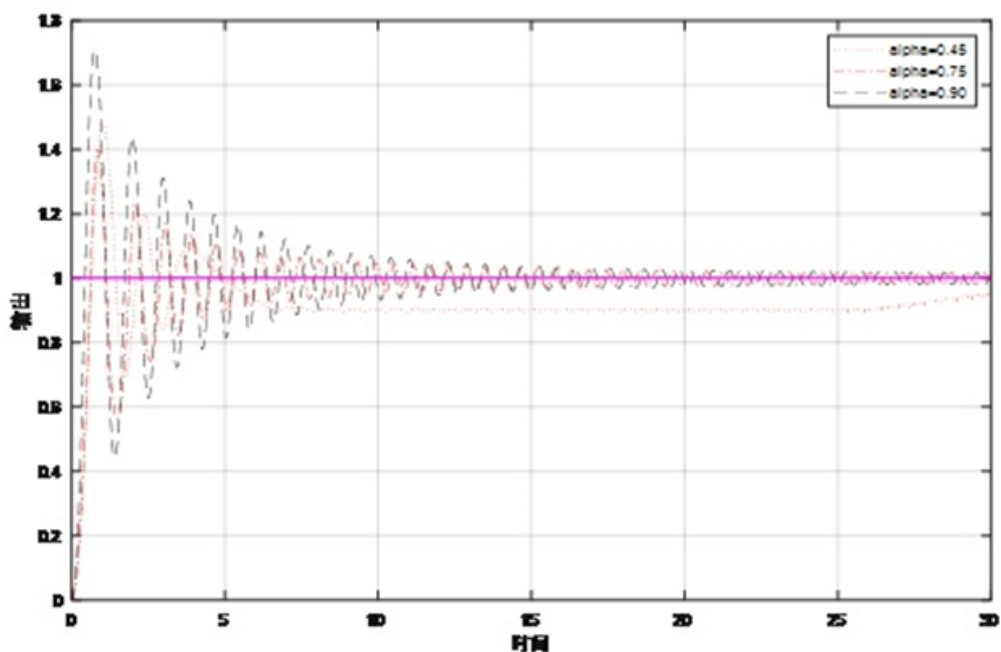
得出如下图像结果



$\alpha$ 取值分别为0.45、0.75、0.9，当 $\alpha = 0.75$ 和 $\alpha = 0.9$ 时，系统对阶跃响应的超调量都比较大，且经过多次振荡之后才能逐渐接近稳态值，但也始终在稳态值的附近振荡。当 $\alpha=0.45$ 时超调量依旧比较大，但是能够趋于稳定，然而有一定的稳态误差。

2. 加入关于误差积分的代码，引入误差 $e$ 的积分 $ie$ 来消除稳态误差，在原来的循环体中加入以下代码：

```
1 ki = 0.03;
2 ie = 0;
3 uu(1, k) = ku * evalfis(in, fisMat) - ki * ie; % 计算输出
4 ie = ie + e * T;
```



综上，匹配较佳动态性能的值 $\alpha$ 可以取0.45。

## 四、实验小结

不得不说，相对传统控制，包括经典控制理论与现代控制理论。模糊控制能避开对象的数学模型(如状态方程或传递函数等)，它目的是对人们关于某个控制问题的成功与失败和经验进行加工，总结出知识，从中提炼出控制规则，用一系列多维模糊条件语句构造系统的模糊语言变量模型，应用模糊推理方法，得到适合控制要求的控制量，可以说模糊控制是一种语言变量的控制。

实验过程中，能明显的感觉到模糊控制系统的鲁棒性特别强，干扰和参数变化对控制效果的影响能被大大减弱，特别适合于我们在自动控制原理这门课中非线性、时变及纯滞后系统的控制。

还有一个明显的优点是，模糊控制可以不需被控对象的数学模型即可实现较好的控制，这是因为被控对象的动态特性已隐含在模糊控制器输入、输出模糊集及模糊规则中。

当然，模糊控制也有缺陷，主要表现在:1. 精度不太高 2. 自适应能力有限 3. 易产生振荡现象。但是，我们几区相关的知识，调整相关的代码和参数后，能够将上述缺点造成的坏处降到最小，这也许就是模糊控制理论的最大特点了。