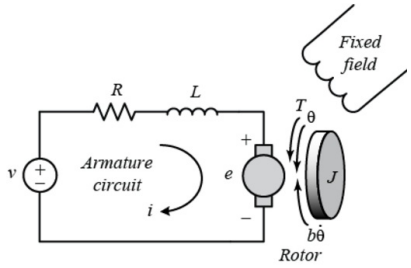
1. **系统建模**（System Modeling）

直流电机的等效电路图如下所示：




1.1> 系统方程

磁场恒定，产生的电机扭矩正比于电枢电流: $T = K_t i$ ;

反电动势正比于转轴转速: $e = K_e\dot{\theta}$

由上及牛顿第二定律和KVL得（设$K_t = K_e = K$)：

$$J\ddot{\theta} + b\dot{\theta} = Ki$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

1.2> 传递函数

由所得的系统方程进行拉普拉斯变换，得

$$s(Js + b)\Theta(s) = KI(s)$$

$$(Ls + R)I(s) = V(s) - Ks\Theta(s)$$

以电压为输入，电机转轴转速为输出，结合两式消除中间变量I(s)。因此传递函数为：

$$P(s) = \frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \qquad [\frac{rad/sec}{V}]$$

确定系统参数及Matlab建模

```
>> J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;
s = tf('s');
P_motor = K/((J*s+b)*(L*s+R)+K^2)
```

```
P_motor =

              0.01
  ----------------------------
  0.005 s^2 + 0.06 s + 0.1001

Continuous-time transfer function.
```

1.3> 状态位置空间

选择转轴转速与通过电枢的电流作为状态变量，关心的输出仍然为转轴转速，则状态空间方程为：

$$\frac{d}{dt}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} = \begin{bmatrix}-\frac{b}{J} & \frac{K}{J}\\ -\frac{K}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}V$$

$$y = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix}$$

```
>> A = [-b/J   K/J
    -K/L   -R/L];
B = [0
    1/L];
C = [1   0];
D = 0;
motor_ss = ss(A,B,C,D)
```

```
motor_ss =

 a =
        x1     x2
  x1   -10      1
  x2  -0.02    -2

 b =
      u1
  x1   0
  x2   2

 c =
      x1  x2
  y1   1   0

 d =
      u1
  y1   0
```

最后确定一下控制目标，对于系统的单位阶跃响应，输出应满足：

- 调节时间 Ts < 2s
- 超调量 %OS < 5%
- 稳态误差 < 1%

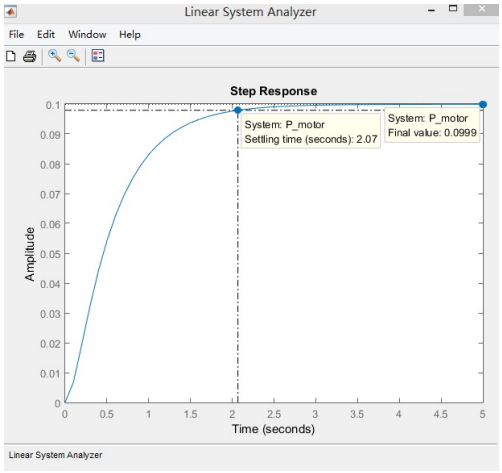
2. **系统分析**（System Analysis）

2.1> 开环响应

为了观察开环系统的响应，我们这次使用之前介绍的LTI GUI： ltiview

```
ltiview('step', P_motor, 0:0.1:5);
```

右击出现的响应图线，Characteristics选择我们关心的Settling time和Steady error。

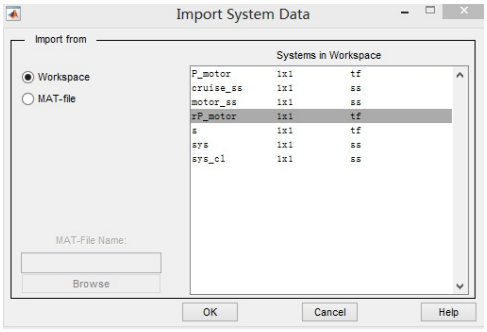



作用于1v的电压仅可得到0.1rad/s的最大转速，并且调节时间也略大于我们的要求。

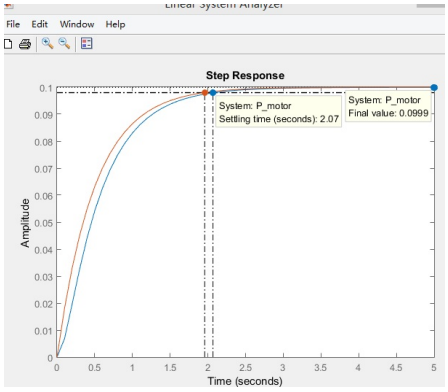
2.2> LTI模型特征

右击得到的阶跃响应图，Plot type -> Pole/zero，我们就得到了开环系统的零极点图，可以观察到有两个负实根极点：-2和-10，两个极点都是实数，因此系统的动态响应无振荡（或超调），两个极点幅值相差五倍，因此响应较慢（幅值较小）的那个极点即-2将主导动态响应，我们用只有一个极点-2的一阶系统的阶跃响应图来和=进行比较

```
>> rP_motor = 0.1/(0.5*s+1)

rP_motor =

      0.1
  ---------
  0.5 s + 1

Continuous-time transfer function.
```




在Matlab中定义rP_motor后，LTI View窗口中选择 File -> Import -> Workspace -> rP_motor，这样也会把新模型导入到LTI View中，右击图像Plot type -> Step恢复到阶跃响应图像，得到下图

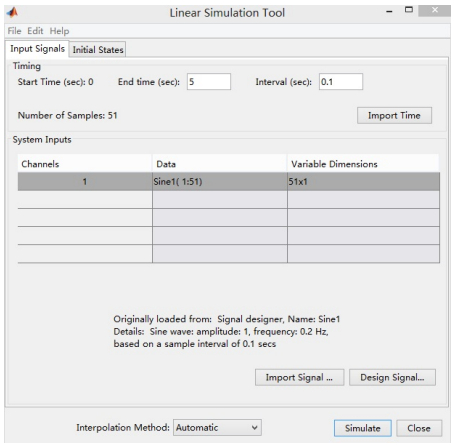


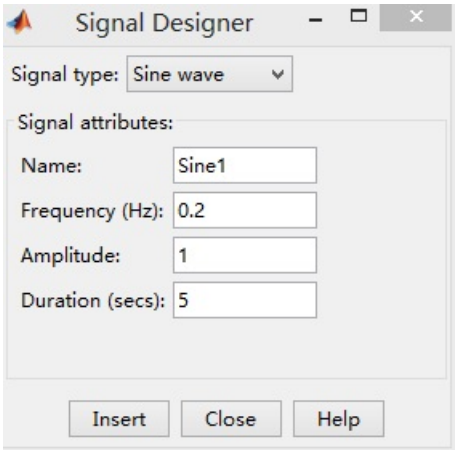

如我们所猜想，两系统动态响应十分接近。一阶系统的Ts = 4t = 4*0.5 = 2s 接近2.07s。

2.3> 其他输入类型的响应

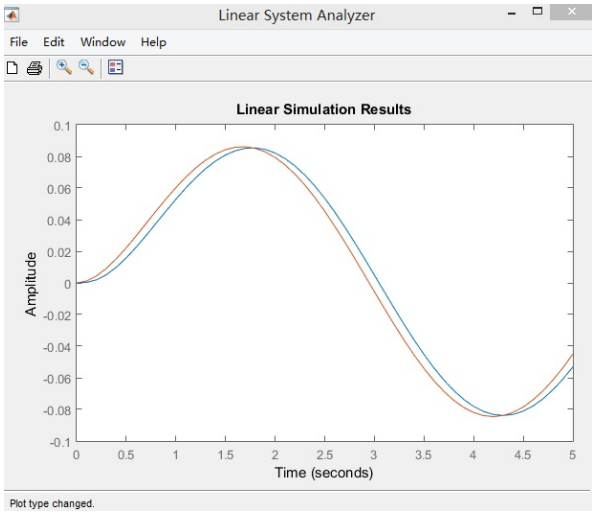
在实际过程中，经常会遇到其他类型的输入，这时可以借助于Simulink或Matlab中的 lsim 命令，也可以继续使用LTI view，

右击图像 -> Plot type -> Linear Simulation，会出现Linear Simulation Tool



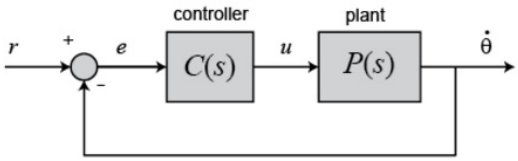




设置End time及Interval，然后导入信号，点击Import Signal，选择信号（目前有 Sine wave, Square wave, Step function, White noise四种），然后Insert到Linear Simulation Tool后点击Simulate在LTI view中显示响应




3. **PID控制**（PID control）

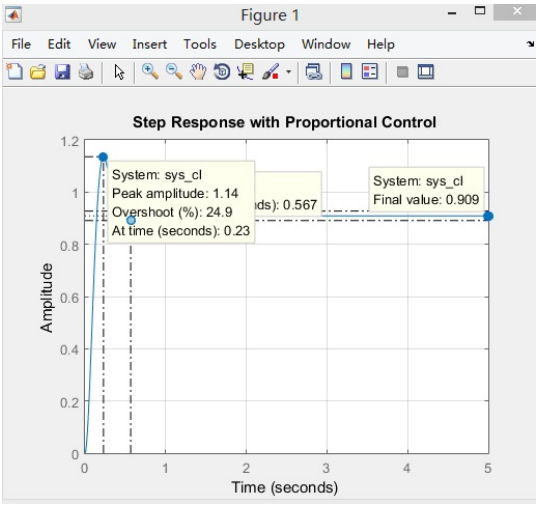
PID控制框图如下




3.1> P控制器

设置Kp = 100，观察系统响应

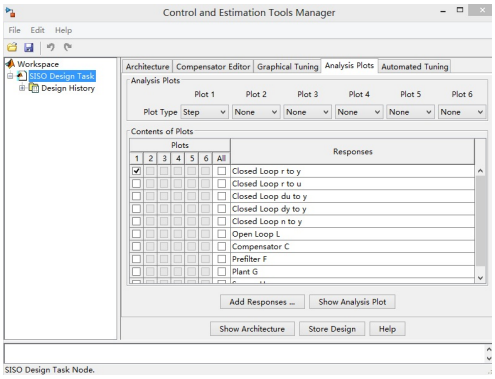
```
>>  Kp = 100;
C = pid(Kp);
sys_cl = feedback(C*P_motor,1);
t = 0:0.01:5;
step(sys_cl,t)
grid
title('Step Response with Proportional Control')
```

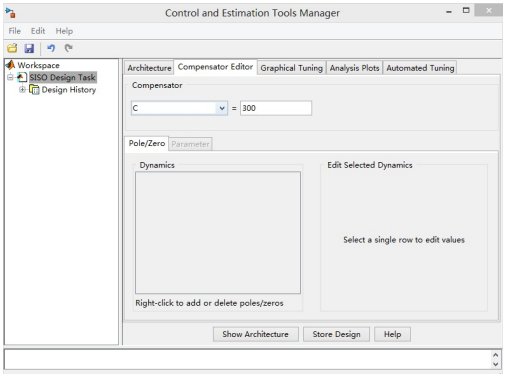


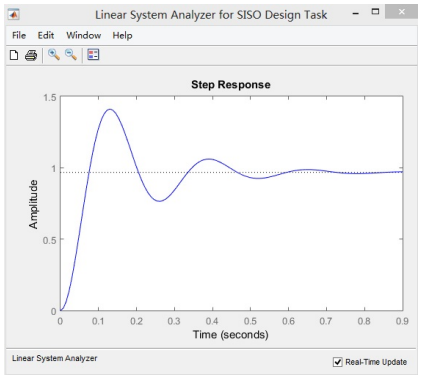

如上说示，虽然系统的动态响应明显加快，但也大大增加了超调，且超过了我们的要求，我们可以通过减小Kp值达到要求，可以输入

```
sisotool(P_motor)  // 或 pidtool(P_motor,'p') 打开PID Tuner 来调节，此处其实更为方便
```

打开Introduction介绍的SISO Design Tool ，然后从Control and Estimation Tools Manager窗口打开Analysis Plots，Plot 1 选择Step，然后打开Compensator Editor，改变增益C的值来观察对应系统闭环的阶跃响应输出



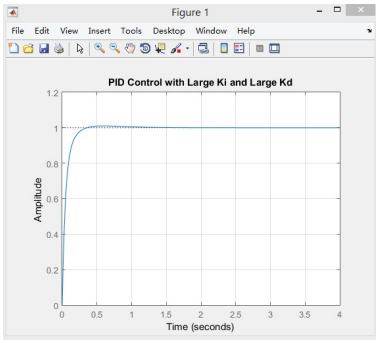




调节不同范围的值会发现，仅依靠P控制器不能同时满足要求，我们需要增加积分或微分环节来设计PI控制器或PD控制器。

3.2> PID控制器

先设置初值，然后按照introfuction中的PID对系统的影响表来改变响应值，或直接使用 `pidtool(P_motor,'pid')` 来取得期望值。一组比较合理的取值及对应的响应如下

```
>>   Kp = 100;
 Ki = 200;
Kd = 10;
C = pid(Kp,Ki,Kd);
sys_cl = feedback(C*P_motor,1);
step(sys_cl, 0:0.01:4)
grid
title('PID Control with Large Ki and Large Kd')
```

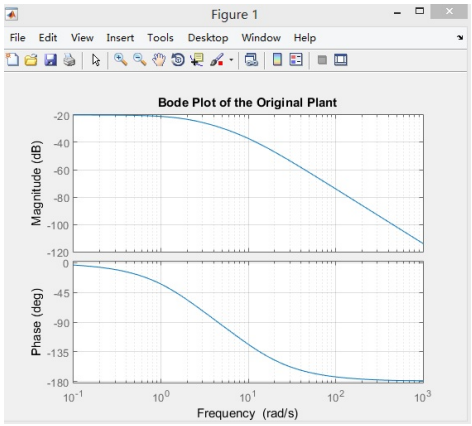



可见，满足我们的要求。

4. **控制器设计中的频域法**（Frequency Domain Methods for Controller Design）

观察原开环系统的bode图

```
>> bode(P_motor)
grid
title('Bode Plot of the Original Plant')
```



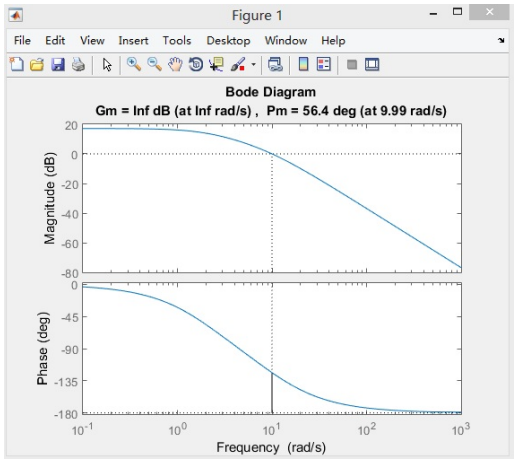

在任何频率下相位、幅值都为负值，即幅值裕度和相位裕度都是无穷大，表明系统的鲁棒性（robust）很好、超调量很小。相位裕度为无穷大，则系统不能无误差地跟踪不同的输入信号，因此我们在保证相位裕度足够的情况下来增大增益。通常60deg的相位裕度对于一个系统来说已经能够保证足够的稳定性了。从bode图可以观察到60deg的相位裕度对应于10rad/s的频率作为增益穿越频率（gain crossover frequency，Wgc），此时幅值约为-40dB，具体值可以给bode函数增加第二个参数：频率。

```
[mag,phase,w] = bode(P_motor,10)
```

```
mag =

    0.0139


phase =

 -123.6835


w =

    10
```

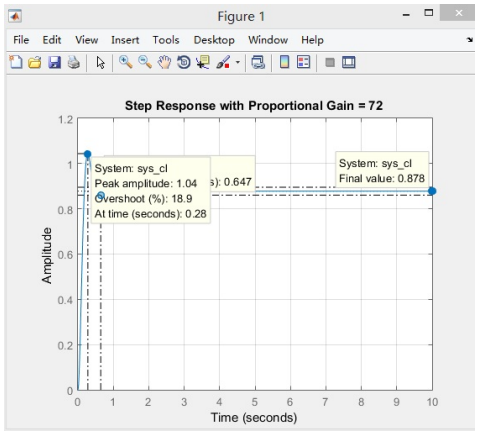
20log0.0139 = -37.1dB，因此要是Wgc = 10,必须增加37.1dB的幅值，即需要给开环系统增加 1/0.0319 = 72的比例增益。使用margin函数可以直接得到带有裕度的bode图

```
C = 72;
margin(C*P_motor);
```




如上图，如我们所期待，此时相位裕度为 56.4deg，接近60deg。观察这时单位闭环系统的阶跃响应

```
>> sys_cl = feedback(C*P_motor,1);
t = 0:0.01:10;
step(sys_cl,t), grid
title('Step Response with Proportional Gain = 72')
```

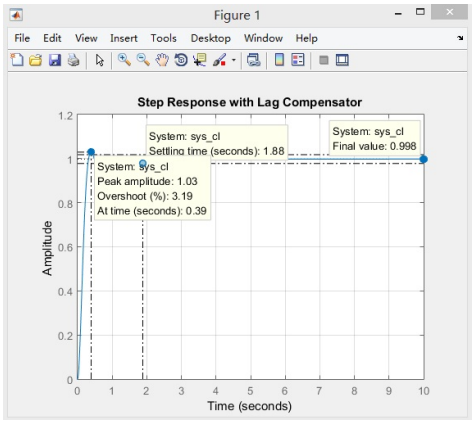



系统响应很快，但是超调量及稳态误差太大了。可以减小比例增益的数值来减小超调，此时的相位裕度也会增大，但是稳态误差会变得更大。这时可以添加一个**滞后补偿器**（Lag Compensator， Extra: Designing Lead and Lag Compensators）来解决这个问题。
考虑这样一个滞后补偿器：

$$C(s) = \frac{(s+1)}{(s+0.01)}$$

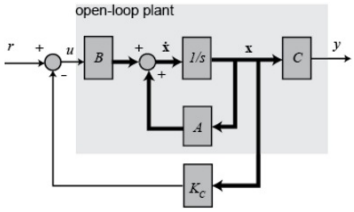
补偿器的增益为 1/0.01 = 100，而这个增益会使得闭环系统的稳态误差减小，甚至此时可以允许一定范围内减小72这个开环系统的比例增益来减小超调且增大相位裕度，取45。

```
>> C = 45*(s + 1)/(s + 0.01);
sys_cl = feedback(C*P_motor,1); t = 0:0.01:10;
step(sys_cl,t), grid
title('Step Response with Lag Compensator')
```




如上，系统的闭环响应满足我们的要求。

5. **控制器设计中的状态空间法**（State-Space Methods for Controller Design）

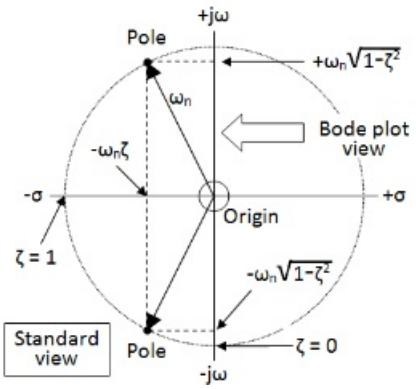



同其他的状态空间法一样，通过改变K值来改变 A-BK （2X2）的特征值即极点位置，进而改变系统的响应。Matlab提供了 **order( )** 函数来确定系统的极点个数，同时 **ctrb(A,B)**对于形位空间系统确定可控矩阵（Controllability matrix），求其秩可以得到极点个数

```
sys_order = order(motor_ss)
sys_rank = rank(ctrb(A,B))  //与上句作用相同
```

```
sys_order =

    2


sys_rank =

    2
```

可控矩阵是满秩，因此系统是可控的。设置极点为-5+i 和 -5-i，




则 sigma = 5，$W_n$ = 5.099，系统的阻尼比zeta = 0.98，

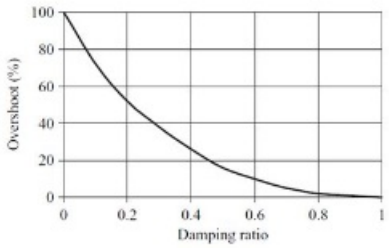
如下图超调与阻尼比的关系所示，此时超调接近0

## The percent overshoot (*Mp*)

- For 2nd order system, the percent overshoot is calculated as

$$M_p = \exp(-\xi\pi / \sqrt{1-\xi^2})$$

- The amount of overshoot depends on the damping ratio (ζ) and directly indicates the relative stability of the system. The lower is the damping ratio, the higher the is maximum overshoot.




而调节时间

$$T_s = \frac{4}{\zeta\omega_n}$$

求得调整时间0.8s。

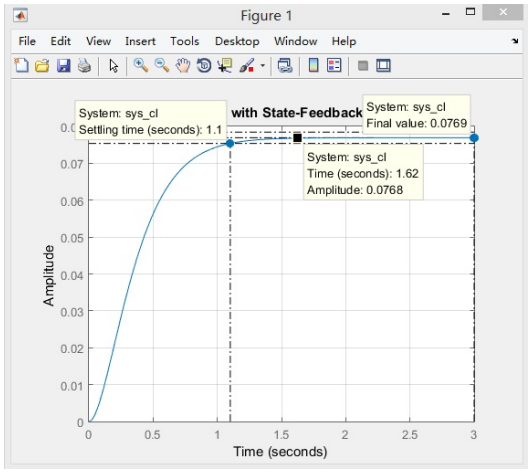
```
>> p1 = -5 + 1i;
p2 = -5 - 1i;
Kc = place(A,B,[p1 p2])

Kc =

   12.9900   -1.0000
```

新的状态空间方程为：

$$\dot{\mathbf{x}} = (A - BK_c)\mathbf{x} + Br$$

$$y = C\mathbf{x}$$

```
>> t = 0:0.01:3;
sys_cl = ss(A-B*Kc,B,C,D);
step(sys_cl,t)
grid
title('Step Response with State-Feedback Controller')
```




满足要求。

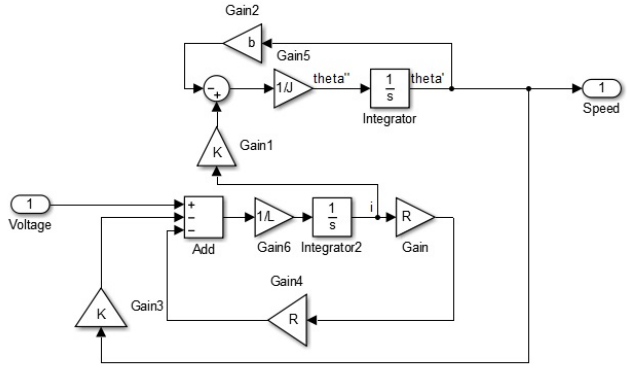
6. **Simulink建模**（Simulink Modeling）

6.1> 采用Simulink建模

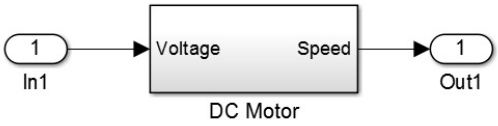
$$J\frac{d^2\theta}{dt^2} = T - b\frac{d\theta}{dt} \Longrightarrow \frac{d^2\theta}{dt^2} = \frac{1}{J}(K_t i - b\frac{d\theta}{dt})$$

$$L\frac{di}{dt} = -Ri + V - e \Longrightarrow \frac{di}{dt} = \frac{1}{L}(-Ri + V - K_e\frac{d\theta}{dt})$$

由系统方程式进行Simulink建模如图，其中将输入电压、输出转速用input 与 output block，以封装为子系统




封装起来，如图所示

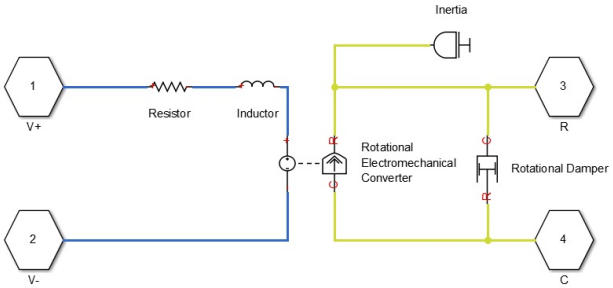



6.2> 采用Simscape建模

Simulink中的Simscape库是对其的拓展，Simscape Library中的模块代表实际的物理元素，因此复杂的多领域模块能够被建立而不用具体写出其数学方程形式。

取出下列模块建立Simscape模型

- Simscape/Foundation Library/Electrical/Electrical Elements library：Resistor, Inductor and Rotational Electromechanical Converter blocks
- Simscape/Foundation Library/Mechanical/Rotational Elements library：Rotational Damper and Inertia blocks
- Simscape/Utilities library：Four Connection Port blocks




设置参数

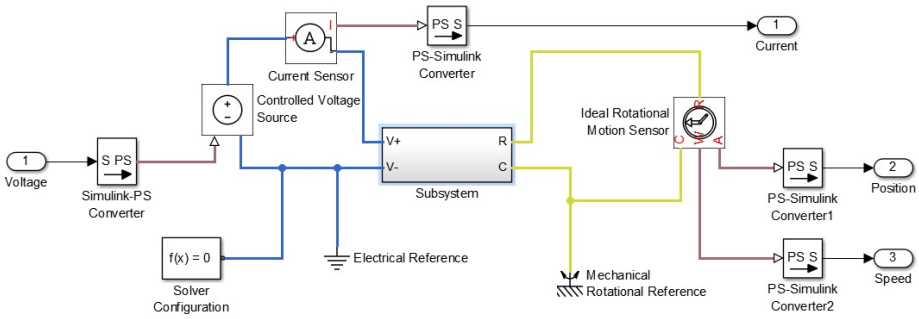
V+ V- 端口位置为 left, R C 端口位置为 Right

Resisitor = R, Inductor = L;  Inertia = J;Constant of proportionality = K; Damping coefficient = b。参数值可在Matlab命令窗口设置。

然后封装起来。

为了仿真系统，还需要添加传感器。取出下列模块建立如下模型

- Simscape/Foundation Library/Electrical/Electrical Sensors library： Current Sensor block
- Simscape/Foundation Library/Electrical/Electrical Sources library：Controlled Voltage Source block
- Simscape/Utilities library：Two PS-Simulink Converter blocks and a Solver Configuration block
- Simscape/Foundation Library/Electrical/Electrical Elements library：Electrical Reference block
- Simscape/Foundation Library/Mechanical/Mechanical Sensors library：Ideal Rotational Motion Sensor block
- the Simscape/Foundation Library/Mechanical/Rotational Elements library：Mechanical Rotational Reference block
- Simulink/Ports & Subsystems library：Three Out1 blocks and one In1 block

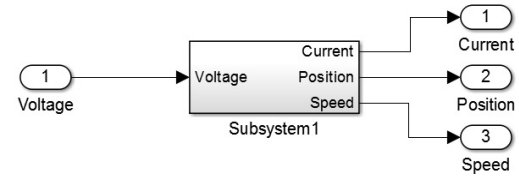



Electrical Reference、Mechanical Rotational Reference分别为电路参考端口（地）和机械旋转参考端口（支架或地）

PS-Simulink Conventer将物理信号（带单位）转换为Simulink输出信号（不带单位），Simulink-PS Conventer与之相反，将Simulink输入信号转换为物理信号

The Solver Configuration来定义用于模型仿真的数值求解器

其他模块可以双击模块查看说明。同样将其封装




DC_Motor_Simsacpe.slx

7. **Simulink控制器设计**（Simulink Controller Design）

仿真、设计上节的模型

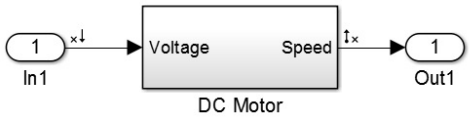
在Matlab命令窗口中输入参数对应值

```
>> J = 0.01;
b = 0.1;
K = 0.01;
R = 1;
L = 0.5;
```

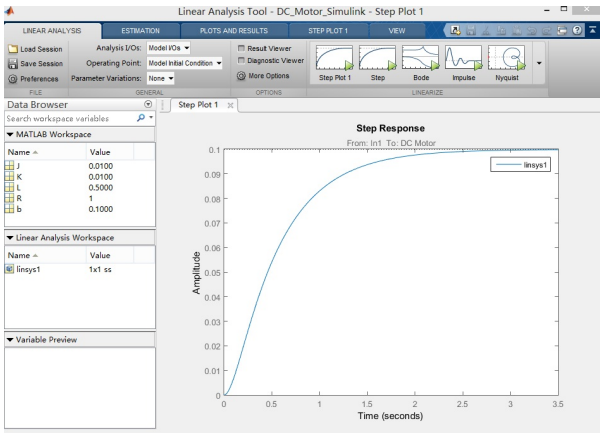
7.1> 抽取线性模型到Matlab

将所建的Simulink模型抽取到Matlab中进行分析，可以使用 linmod 函数 或 如下直接从Simulink中抽取

右击代表电压输入的信号线选择Linear Analysis Points -> open-loop input，同样右击转速输出信号线选择Linear Analysis Points -> open-loop output




然后在Simulink模型窗口顶部选择 Analysis -> Control Desogn -> Linear Analysis打开Linear Analysis Tool，点击step获得系统阶跃响应




将Linear Analysis Workspace中的linsys1对象拖入Matlab Workspace即可将模型导入Matlab
在Matlab中命令行输入

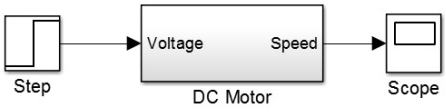
```
zpk(linsys1)
```

```
ans =

  From input "In1" to output "DC Motor":
           2
  -------------------
  (s+9.997) (s+2.003)

Name: Linearization at model initial condition
Continuous-time zero/pole/gain model.
```

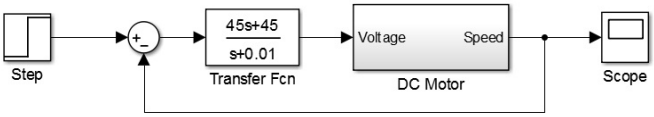
等效于

```
>> s = tf('s');
P_motor = K/((J*s+b)*(L*s+R)+K^2);
zpk(P_motor)
```

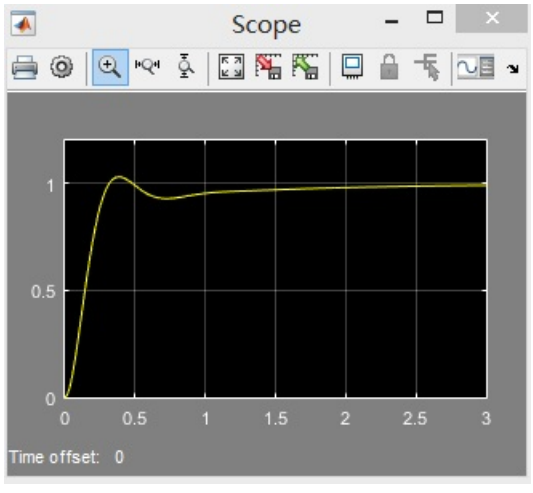
7.2> 开环响应
直接在Simukink模型中得到开环响应，一出 input、output block，输入换位Step block，输出换位Scope block




双击Step Block设置step time为0
点击Run，双击查看Scope，得到开环响应同上。
7.3> 闭环响应
**滞后补偿器**（lag compensator）在系统频域法设计控制器时为了闭环系统同时满足所有响应要求，加了一个滞后补偿器，在Simulink中建模如下




仿真时间设为3s，得到响应



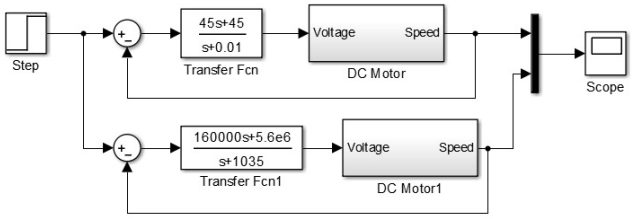
满足要求。

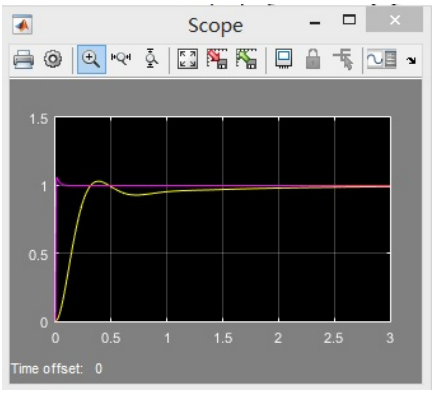
**前置补偿器**（lead compensator）

设计一个前置补偿器

$$C_{lead}(s) = 160000\frac{s+35}{s+1035}$$

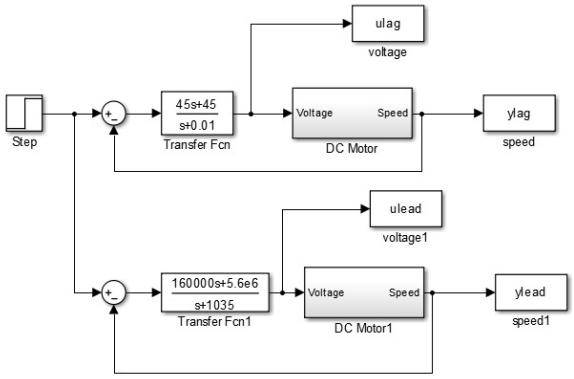
建立Simulink模型




观察输出并与带滞后补偿器的闭环系统比较，带前置补偿器的闭环系统比带滞后补偿器的响应快得多，但是我们还是会选择滞后控制器，滞后控制器比前置控制器所需的控制能量要小，因此器件的尺寸可以做的更小。为了观察这一差别，重建Simulink模型，使其将相关信号导入到Matlab中，添加

4个To Workplace blocks(Sink Library），参数 save format 设为 array ，即输出为矩阵，这些变量在仿真过程中的默认自变量为tout



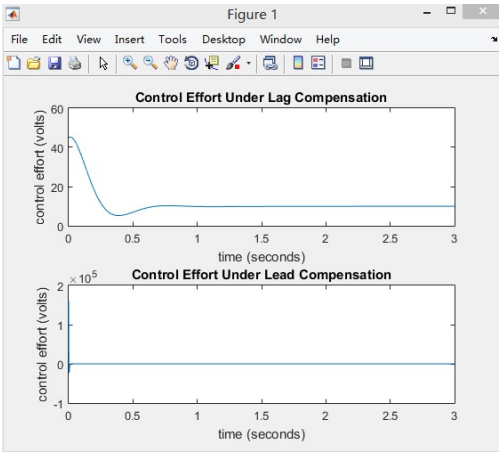

DC_Motor_Simulink.slx

点击Run，

在Matlab命令行中输入

```
>> subplot(2,1,1)
plot(tout,ulag);
xlabel('time (seconds)')
ylabel('control effort (volts)')
title('Control Effort Under Lag Compensation')
subplot(2,1,2)
plot(tout,ulead);
xlabel('time (seconds)')
ylabel('control effort (volts)')
title('Control Effort Under Lead Compensation')
```

比较两者输入电压值




显然带 Lead Compensation 的闭环系统大得多，最高值达到了1.5X10^5v，远超过了直流电机的耐压极限。