计算机控制系统大作业

院(系)名称	高等理工学院
学　　号	16231235
姓　　名	李谨杰
指导教师	周锐

2019 年 05 月

综合习题一

已知： $D(s)=\dfrac{3}{s+3}$

1）试用 Z 变换、一阶向后差分、向前差分、零极点匹配、Tustin 变换和预修正的 Tustin（设关键频率＝3）变换等方法将 D(s)离散化，采样周期分别取为 0.05s 和 0.3s；

- Z 变换法：查表 $Z[\dfrac{1}{s+a}]=\dfrac{z}{z-e^{-aT}}$，代入 a=3，T，增益乘 3 即可。之后要配增益，由于采样的原因，连续域和离散域的频率响应有关系 $H_D\left(e^{j\Omega}\right)=\dfrac{1}{T}\sum_{n=-\infty}^{\infty}H_A\left(j\dfrac{\Omega-2\pi n}{T}\right)$，故需要对数字响应乘上比例因子 T，这样才能使得脉冲响应的值相等。
- 一阶向后差分法：令 $s=\dfrac{1-z^{-1}}{T}$，代入 D(s)即可。
- 一阶向前差分法：令 $s=\dfrac{1-z^{-1}}{Tz^{-1}}$，代入 D(s)即可。
- 零极点匹配法：s 域有极点-3，$z=e^{sT}$；同时 D(s)分子阶次小于分母阶次 1 次，故 D(z)分子上应配（z+1）因子；增益按照 $D(s)\big|_{s=0}=D(z)\big|_{z=1}$ 配置。按此规则计算的 D(z)填在表格中。
- Tustin变换法：令 $s=\dfrac{2}{T}\dfrac{(1-z^{-1})}{(1+z^{-1})}$，代入D(s)即可。Matlab中用 `c2dm(num,den,T,'tustin');`指令配置。
- 预修正的 Tustin（设关键频率＝3rad/s）变换：在ω1=3rad/s 处，保证幅值和相位均不失真。令 s=$\dfrac{\omega_1}{\tan(\omega_1 T/2)}\dfrac{z-1}{z+1}$，代入 D(s)即可。

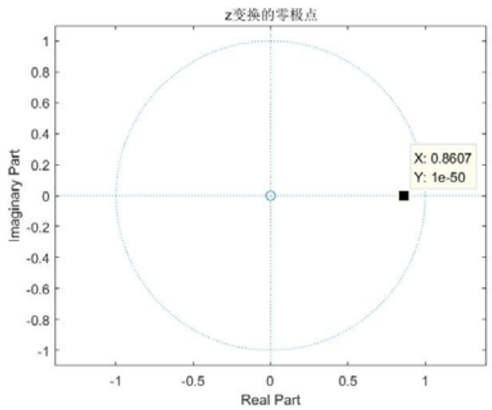
	T=0.05s 时的 D (z)	T=0.3s 时的 D (z)
Z 变换	$\dfrac{3z}{z-0.8607}$，配完增益为：$\dfrac{0.15z}{z-0.8607}$	$\dfrac{3z}{z-0.4066}$ 配完增益为：$\dfrac{0.9z}{z-0.4066}$

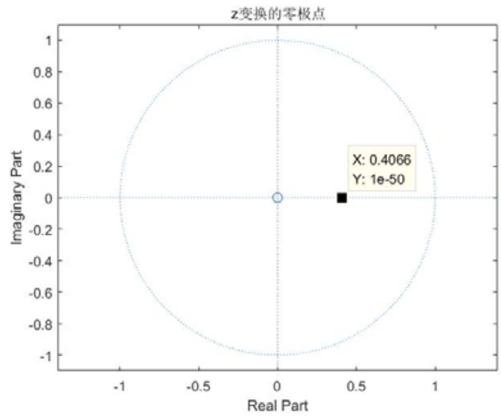
一阶向后差分	$\frac{0.1304z}{z-0.8696}$	$\frac{0.4737z}{z-0.5263}$
向前差分	$\frac{0.15}{z-0.85}$	$\frac{0.9}{z-0.1}$
零极点匹配	$\frac{0.06965(z+1)}{z-0.8607}$	$\frac{0.2976(z+1)}{z-0.4066}$
Tustin 变换	$\frac{0.06977z+0.06977}{z-0.8605}$	$\frac{0.3103z+0.3103}{z-0.3793}$
预修正的 Tustin（ω=3）	$\frac{0.0699z+0.0699}{z-0.8602}$	$\frac{0.3257z+0.3257}{z-0.3486}$

2）将 D(z)的零极点标在 Z 平面图上。

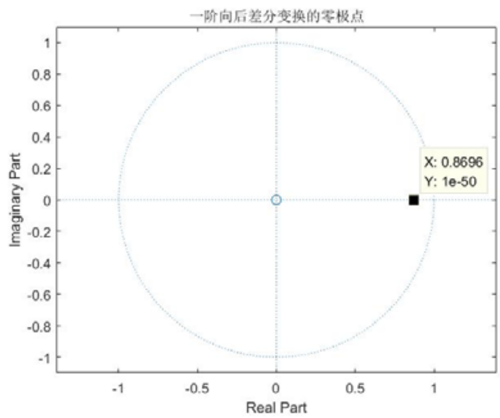
T=0.05s　　T=0.3s

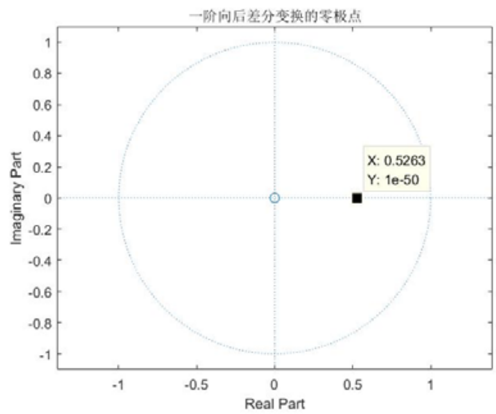
Z 变换



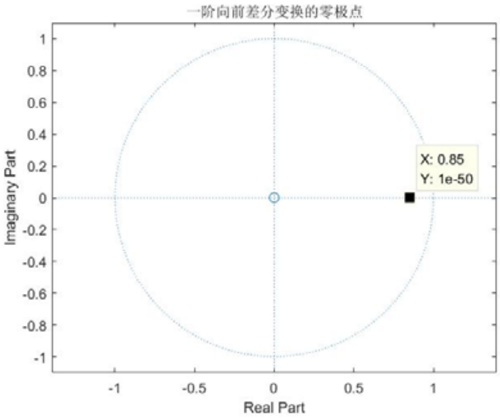


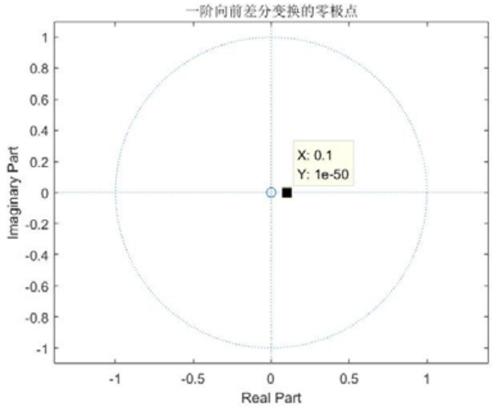
一阶向后差分



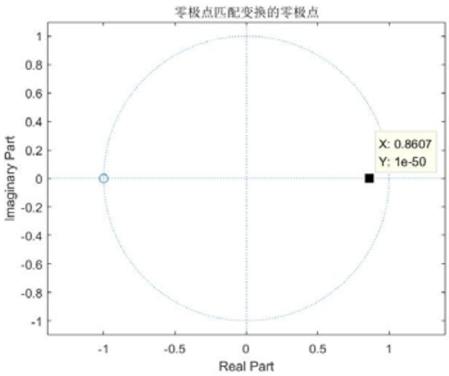


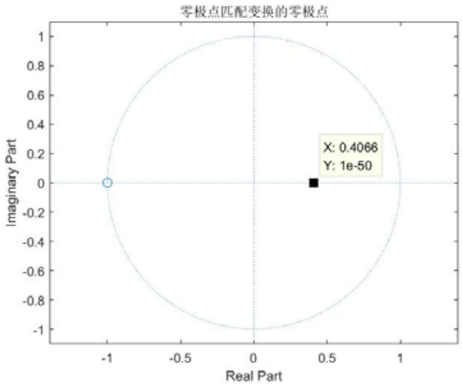
向前差分



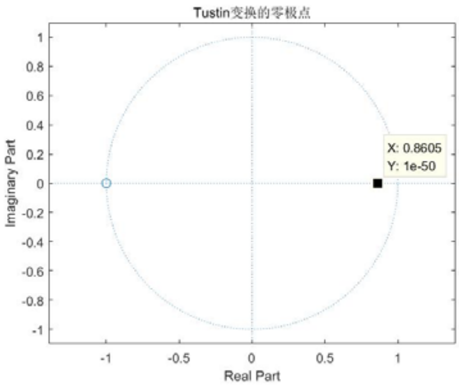


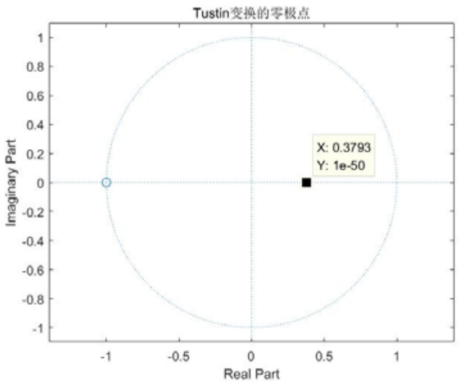
零极点匹配



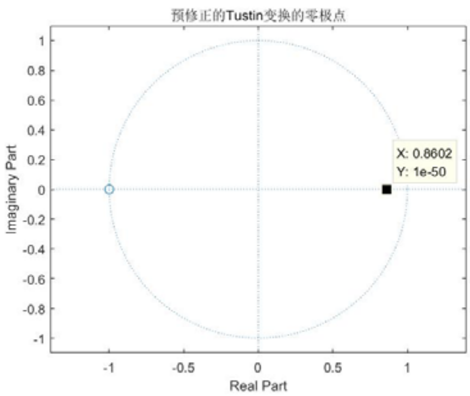


Tustin变换





预修正的Tustin（ω=3）



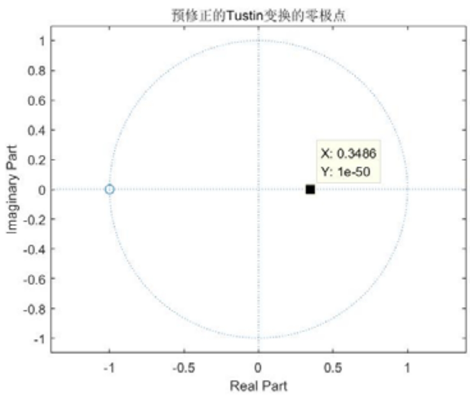


图 1.1 各种变换在 Z 域上的零极点

3）计算 $\mathbf{D(j\omega)}$和各个$\mathbf{D(e^{j\omega T})}$的幅频和相频特性并绘图（等频率轴），ω由

0~30 rad，平均计算 31 个点，每个 T 绘一张图（Z 变换方法单画），共 4 张。

通过 Matlab 指令作图。以 Tustin 变换的指令为例：

```
clear;clc;
T=0.3;
w=0:1:30;
num=3;
den=[1,3];
[n5,d5]=c2dm(num,den,T,'tustin');
Dztus=tf(n5,d5,T);
[m5,p5]=dbode(n5,d5,T,w);
```

dbode 函数返回的幅值为十进制，需要用指令 20*log10(mag)转化为分贝。

T=0.05s↓

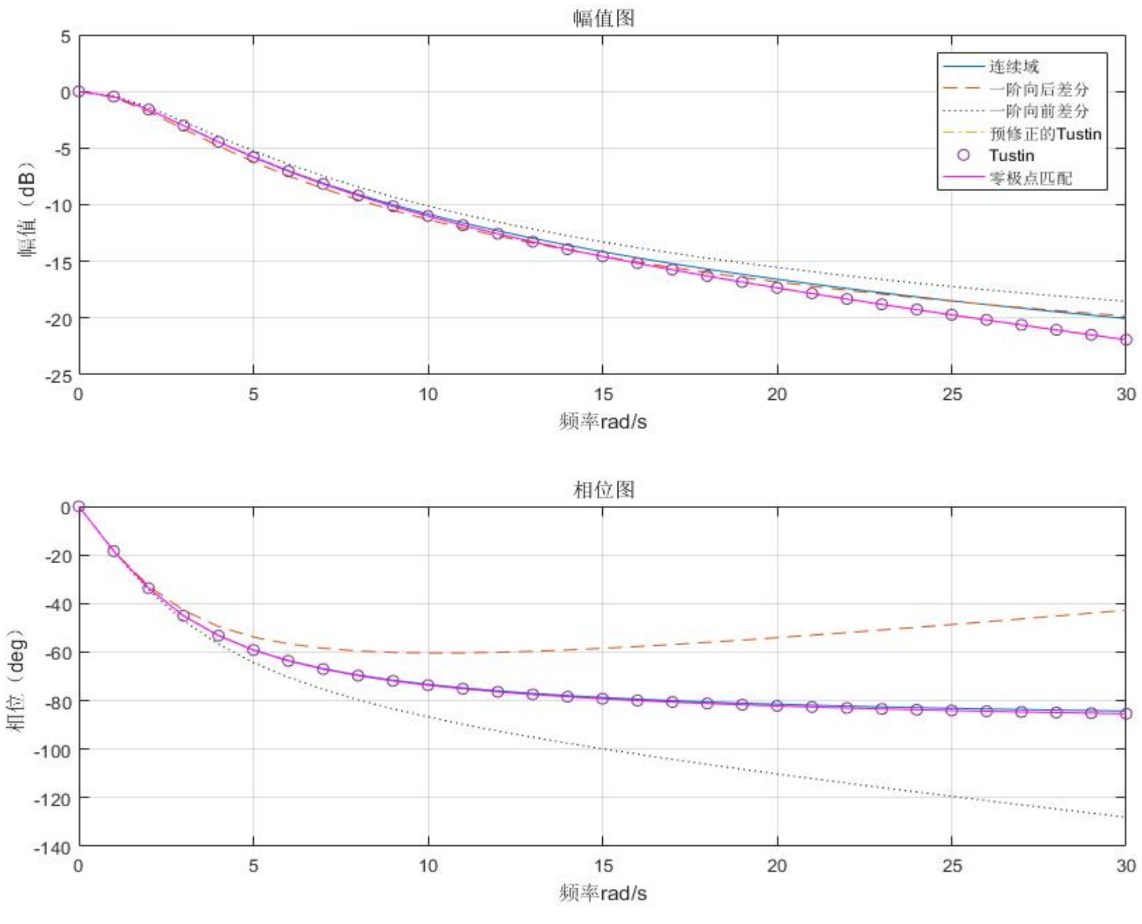


图 1.2 T=0.05s 时不同变换的幅值和相位图

T=0.3s↓

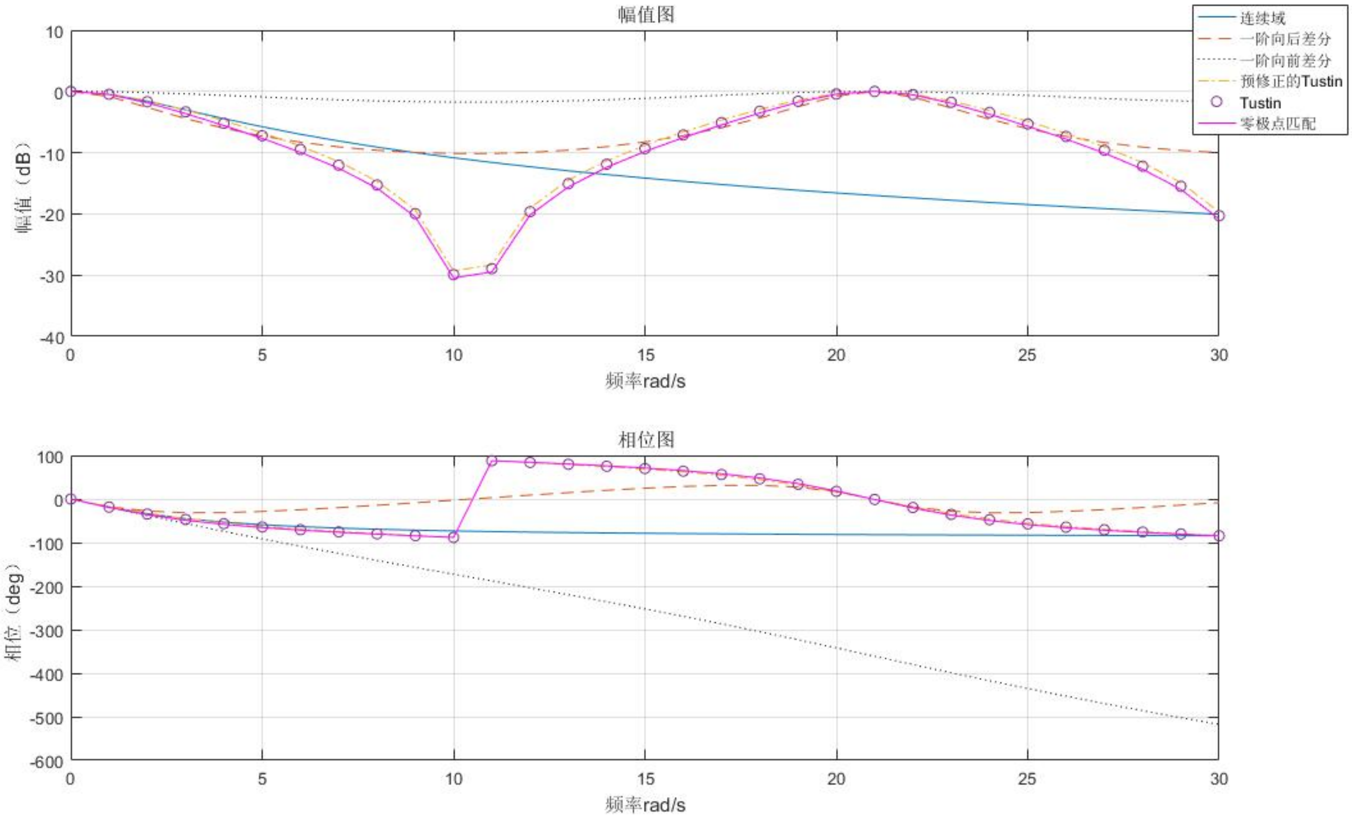
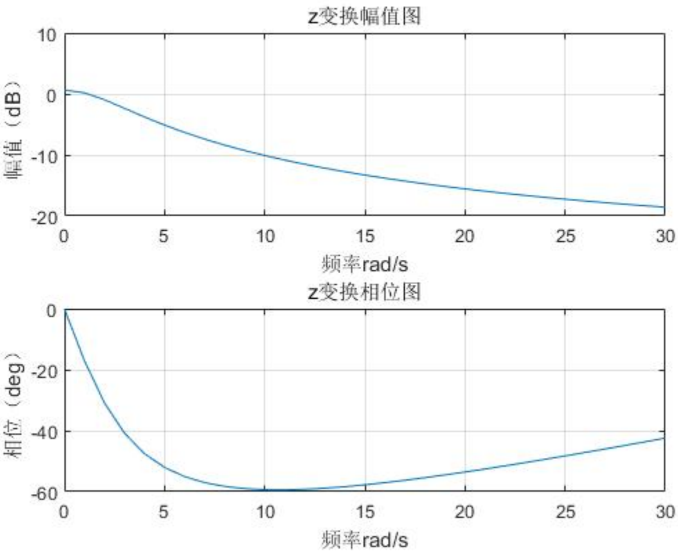


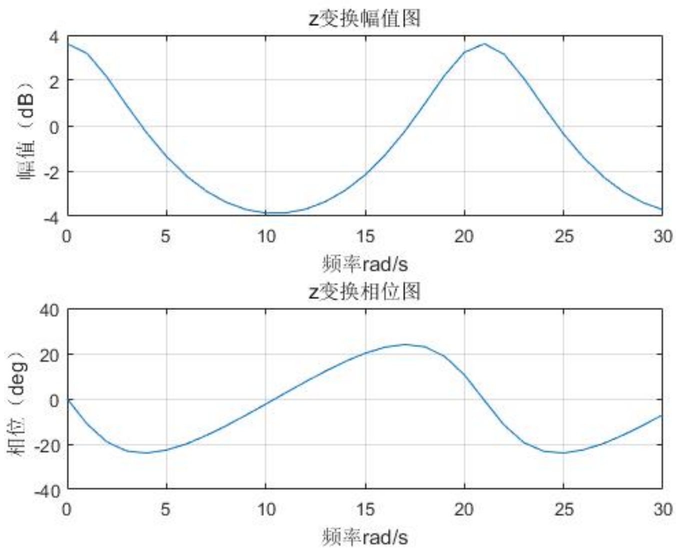
图 1.3 T=0.3s 时不同变换的幅值和相位图

Z 变换：

T=0.05s



T=0.3s



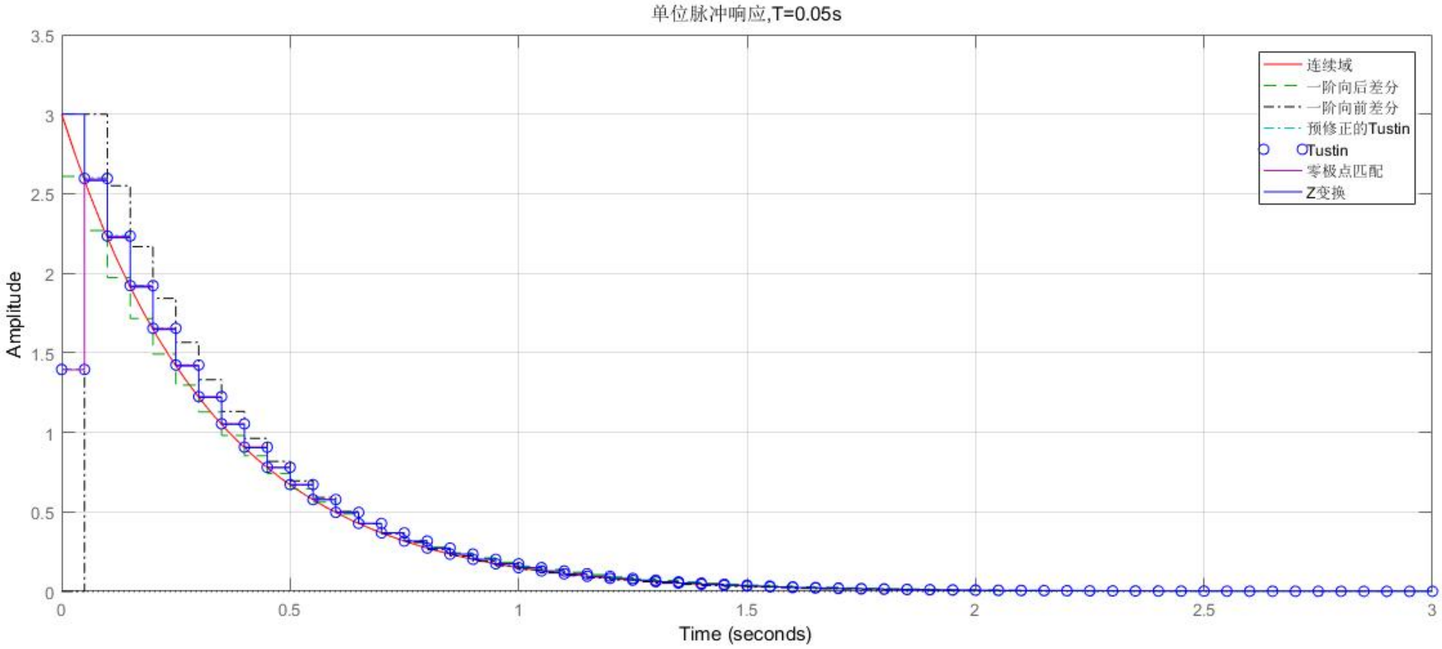
4）计算 D(s)及 T=0.05s 和 T=0.3s 时 D(z)的单位脉冲响应，运行时间为 3

秒。

针对此题，使用 impulse 函数。在已经构建各系统的前提下，部分代码如下：

```
impulse(Gs,'r-',Dzhou,'g--
',Dzqian,'k-.',Dzxiu,'c-.',Dztus,'bo',Dlingjidian,'m',3);
```

单位脉冲响应，T=0.05s



单位脉冲响应，T=0.3s

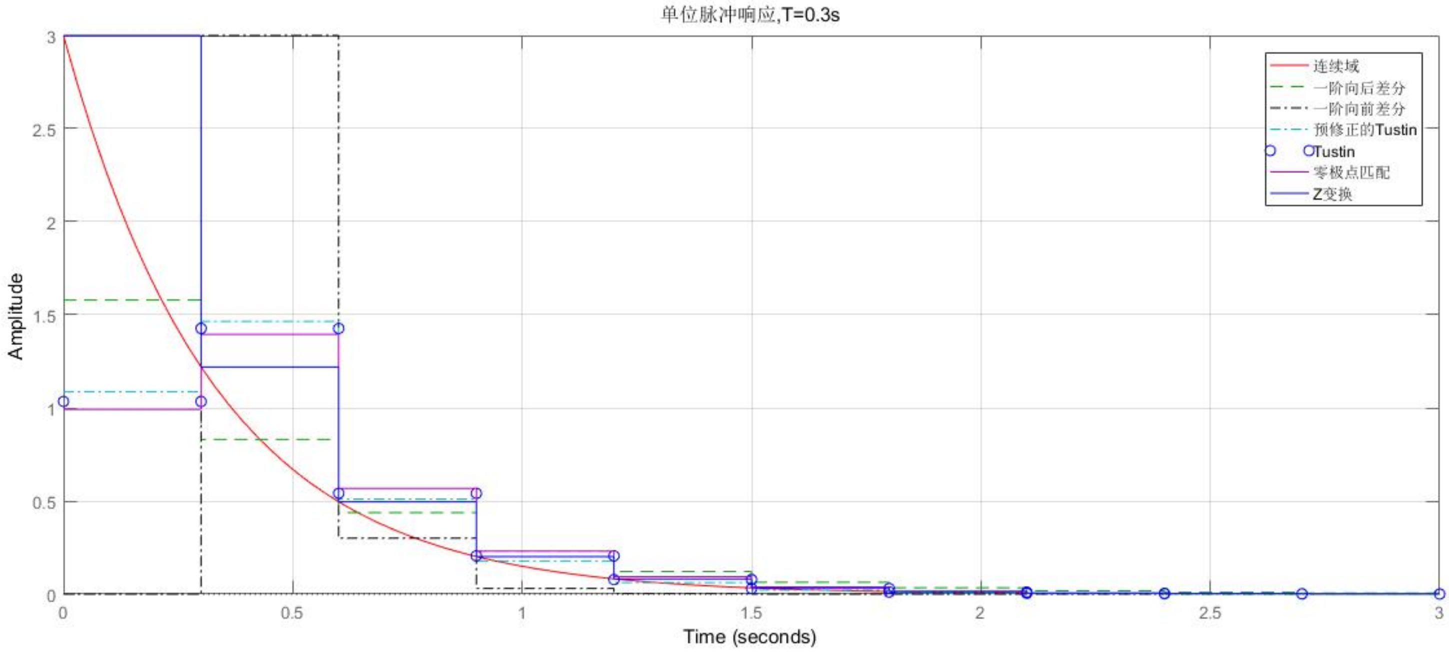


图 1.4 不同变换的单位脉冲响应图

5）结合所得的结果讨论分析各种离散化方法的特点。

1. 从零极点位置可以看出，零极点匹配法、Tustin 变换法和预修正的 Tustin

变换法均在(-1,0)处引入零点，而 Z 变换、一阶向后差分、向前差分法引入的零点在原点。就极点的位置来说，T 两种取值状态下系统均稳定。T=0.05s 时，快速性方面，一阶向后差分方法最差，一阶向前差分方法最好（最接近原点），其他差不多，总的来说快速性相差不大；T=0.3s 时，向前差分方法快速性最好，极点距原点最近，其他极点位置相差不大，Tustin 变换和预修正的 Tustin 变换的快速性要稍好一些。

2. 从变换前后的幅频和相频特性来看，可以得出以下结论：

（1）当采样时间较小时，幅值方面不论哪种离散化方法都有很好的结果，与原系统相差不大；相位方面，预修正的 Tustin 变换和一阶向前变换有一定的误差，其他变换几乎没有误差。

（2）当采样时间较大时，幅值方面，Z 变换法（脉冲响应不变法）出现频谱混叠现象，导致在频率等于 22rad/s 左右的地方出现第二个峰值。Tustin 变换、预修正的 Tustin 变换和零极点匹配法有相似的幅频特性，频率大于 5rad/s 后失真严重，出现了频率畸变现象，因此适用于低通环节。一阶向后差分变换的效果最好，在 10rad/s 以后才有失真。一阶向前差分失真最严重，整个频率图中幅值在 0dB 左右徘徊。同时由下图可以观察到与 Tustin 变换相比，预修正的 Tustin 变换在ω=3rad/s 时的幅频和相频均没有失真，与设计初衷相符。

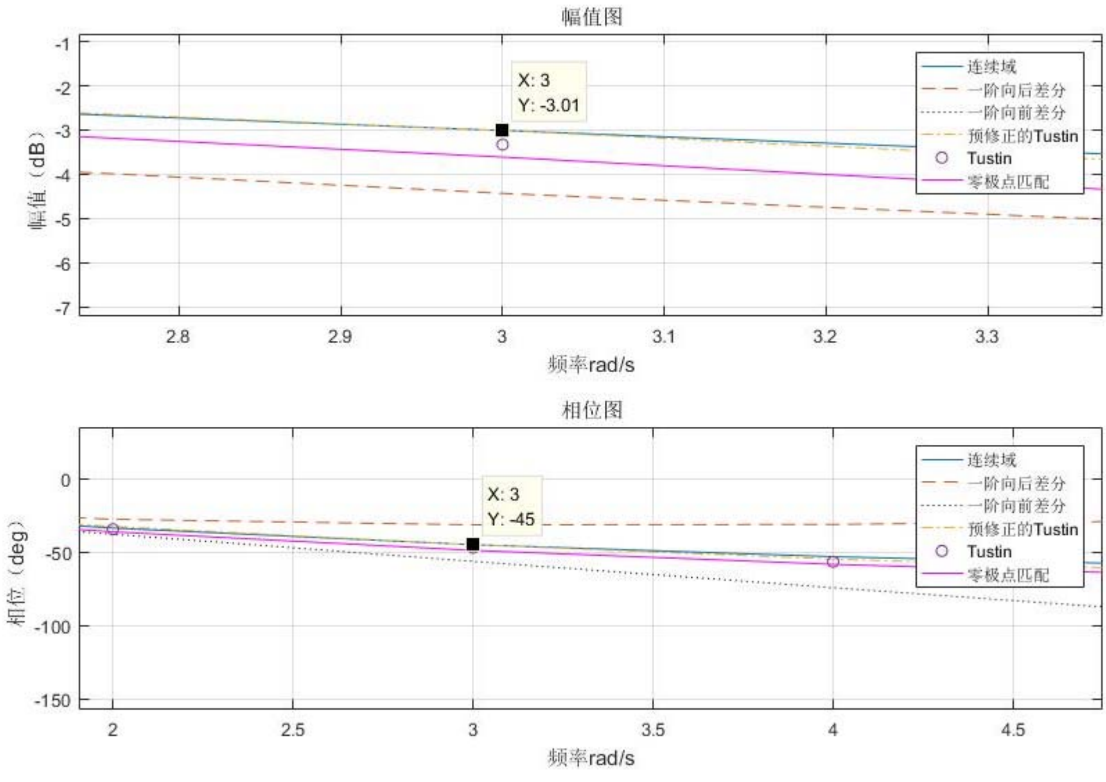


图 1.5 预修正的 Tustin 变换在频率等于 3rad/s 时的局部图

3. 从单位脉冲响应可以得出如下结论：

（1）除了脉冲响应不变法，其他各种变换方法在第一个周期内都不能很好地拟合原曲线。

（2）当采样时间较小时，各种变换方法得到的结果相似，都与连续系统响应接近。

（3）当采样时间较大时，脉冲响应不变法得到的结果完全体现为对连续响应的采样；除第一个周期外，其他的变换方法均略有失真，一阶向前差分方法的失真最大，一阶向后差分方法的失真较大。因此在实际工程实践中，若采样周期较大应尽量避免使用一阶向前差分法，一阶向后差分法。

6）收获与感悟

经过这次大作业，我深刻理解了不同离散方法的优缺点，同时掌握了MATLAB 的求离散传递函数和作图的方法，收获颇丰。

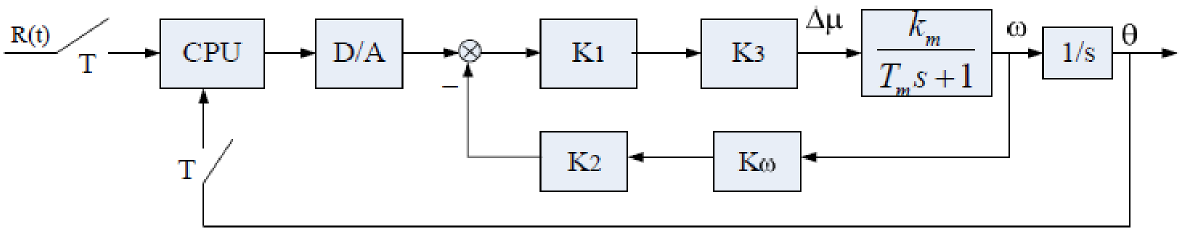
7）源代码

```
clear;clc;
T=0.3;
w=0:1:30;
num=3;
den=[1,3];
Gs=tf(num,den);
[m0,p0]=bode(num,den
,w);

[n,d]=c2dm(num,den,T
,'imp');
% n=[0.5934,0];
Dz=tf(n,d,T);
[m1,p1]=dbode(n,d,T,
w);
% figure(1);
% zplane(n,d);
% title('z变换的零极点
');

n2=[3*T,0];
d2=[1+3*T,-1];
Dzhou=tf(n2,d2,T);
[m2,p2]=dbode(n2,d2,
T,w);
% figure(2);
% zplane(n2,d2);
% title('一阶向后差分变
换的零极点');

n3=3*T;
d3=[1,3*T-1];
Dzqian=tf(n3,d3,T);
[m3,p3]=dbode(n3,d3,
T,w);
% figure(3);
% zplane(n3,d3);
% title('一阶向前差分变
换的零极点');

a=1/tan(1.5*T);
n4=[1,1];
d4=[1+a,1-a];
Dzxiu=tf(n4,d4,T);
[m4,p4]=dbode(n4,d4,
T,w);
% figure(4);
% zplane(n4,d4);
% title('预修正的
Tustin变换的零极点');

[n5,d5]=c2dm(num,den
,T,'tustin');
Dztus=tf(n5,d5,T);
[m5,p5]=dbode(n5,d5,
T,w);
% figure(5);
% zplane(n5,d5);
% title('Tustin变换的
零极点');

if T==0.05
    a=0.06965;
else if T==0.3
        a=0.2976;
    end
end
n6=[a,a];
Dlingjidian=tf(n6,d,
T);
[m6,p6]=dbode(n6,d,T
,w);
% figure(6);
% zplane(n6,d);
% title('零极点匹配变换
的零极点');

figure(7);
subplot(2,1,1);
plot(w,20*log10(m0),
'-
',w,20*log10(m2),'--
',w,20*log10(m3),'k:
',w,20*log10(m4),'-.
',w,20*log10(m5),'o'
,w,20*log10(m6),'m')
,title('幅值图');
xlabel('频率
rad/s');ylabel('幅值
（dB）');grid on;
legend('连续域','一阶
向后差分','一阶向前差分
','预修正的
Tustin','Tustin','零
极点匹配');

subplot(2,1,2);
plot(w,p0,'-
',w,p2,'--
',w,p3,'k:',w,p4,'-.
',w,p5,'o',w,p6,'m')
,title('相位图');
```

```
xlabel('频率
rad/s');ylabel('相位
（deg）');grid on;
legend('连续域','一阶
向后差分','一阶向前差分
','预修正的
Tustin','Tustin','零
极点匹配');

% figure(8);
% subplot(2,1,1);
%
plot(w,20*log10(m1))
,title('z变换幅值图');
% xlabel('频率
rad/s');ylabel('幅值
（dB）');grid on;
% subplot(2,1,2);
%
plot(w,p1),title('z变
换相位图');
% xlabel('频率
rad/s');ylabel('相位
（deg）');grid on;

%这一部分对应阶跃响应
% figure(9);
% impulse(Gs,'r-
',Dzhou,'g--
',Dzqian,'k-.',Dzxiu
,'c-.',Dztus,'bo',Dl
ingjidian,'m'...
%    ,Dz,'b',3);grid
on;title('单位脉冲响
应,T=0.3s');
% legend('连续域','一
阶向后差分','一阶向前差
分','预修正的
Tustin','Tustin','零
极点匹配','Z变换');

% figure(10);
%
impulse(Gs,'r',Dz,'b
',3);grid on;title('
单位脉冲响
应,T=0.05s');
% legend('连续域','Z变
换');
```

综合习题二 计算机伺服控制系统设计

1. 已知：被控对象为一个带有均质圆盘负载的直流力矩电机，其伺服系统方框图如下：



其中，电机传递函数为角速率$\omega/\Delta u$和转角$\theta/\Delta u$；模拟控制器由 K1、K2、K3 组成，数字控制器由采样、CPU（控制律）和 D / A 组成。
给定参数如下：

- 电机传递函数$G(s)=\dfrac{\theta(s)}{u(s)}=\dfrac{k_m}{s(T_m s+1)}$ ， $k_m=2rad/s$， $T_m=0.1s$
- 电机启动电压$\Delta u_A=1.7\mathbf{v}$
- 测速机传递系数$k_\omega=1\mathbf{v}/\mathbf{rad}/\mathbf{s}$
- 电位计最大转角为 345° ，输出±5v
- 功放 KA=2=K3
- 采样周期 T=0.010s

2. 设计要求：

1） D/A 输出 120mv，电机启动： $\Delta u_A=1.7\mathbf{v}$

2） D/A 输出 5v，电机转速$\omega=26\mathrm{rad/s}$

3） 设计状态反馈增益 K，使系统闭环极点$\zeta\ge0.9$， $\omega_n\ge20\mathrm{rad/s}$

4） 设 θ 可测，设计降维观测器（求 L），取观测器衰减速率是系统闭环衰减速率的 4 倍。

5） 求调节器的离散控制律 D(z)=U(z)/Y(z)。

6） 将 D(z)进行实现，配置适当的比例因子，编制相应的程序流程图。

7） 仿真验证调节器的控制效果。假设系统受到扰动，初试状态为：初速$\omega_0=0$，初始角度$\theta_0=10°=0.175rad$。看看是否经过一定时间后，系统状态回到平衡的零态。

8） （选作）引进指令信号，设计相应的指令跟踪控制器，仿真给出闭环系统的阶跃响应曲线。

设计步骤如下：

1．**确定基本参数。**

因为启动时ω=0，D/A 输出为 0.12V，δu = 1.7V，故 $K1\cdot 2=\frac{1.7v}{0.12v}=14.17\approx 15$，解得 K1=7.5。

因为 D/A 输出 5V，电机可以稳定在ω=26rad/s。由系统结构图得：$G(j\omega')=\frac{30}{K_2\cdot 30+0.1j\omega'+1}$，这里的ω’指输入信号的频率。因为输入与输出稳定，可以看做常值信号，故ω’=0，即 $G(j0)=\frac{30}{K_2\cdot 30+1}=\frac{\omega}{U_{DA}}=\frac{26}{5}$。解得 K_2=0.159。

因此，D/A 环节到输出之间的传递函数为 $G(s)=\frac{300}{(s+57.7)s}$。

2．**设计状态反馈**

根据 $\zeta\geq 0.9,\ \ \omega_n\geq 20\mathrm{rad/s}$ 的指标和二阶系统极点公式：

$s_{1,2}=-\zeta\omega_n\pm\omega_n\sqrt{\zeta^2-1}$ 可求得期望极点为：$s_{1,2}=-18\pm j20\sqrt{0.19}$

Z 域上由 $z=e^{sT}$ 可得 $z_{1,2}=0.8321\pm 0.0727j$

取 $x_1=\theta,x_2=\omega=\dot{\theta}$ 配置状态矩阵：

$$A=\begin{pmatrix}0 & 1\\ 0 & -57.7\end{pmatrix}\quad B=\begin{pmatrix}0\\ 300\end{pmatrix}\quad C=\begin{pmatrix}1 & 0\end{pmatrix}$$

由MATLAB指令`[F,G,Cd,Dd]=c2dm(A,B,C,0,T)`得到

$$F=\begin{pmatrix}1 & 0.0076\\ 0 & 0.5616\end{pmatrix}\quad G=\begin{pmatrix}0.0125\\ 2.2795\end{pmatrix}\quad C=\begin{pmatrix}1 & 0\end{pmatrix}\quad \mathrm{D=0}$$

取 $\mathbf{k}=[\mathrm{k}_1,\mathrm{k}_2]$

由MATLAB指令：`Pd=exp(P*T);k=place(F,G,Pd);`得到配置完的状态反馈增益矩阵K=[1.4688 -0.0531]，状态方程如下：

$$\mathbf{x}(\mathrm{k}+1)=(\mathrm{F}-\mathrm{GK})\mathbf{x}(\mathrm{k})+\mathrm{Gu}(\mathrm{k})$$

$$\mathbf{x}(\mathrm{k}+1)=\begin{pmatrix}0.9817 & 0.0083\\ -3.3480 & 0.6825\end{pmatrix}\mathbf{x}(\mathrm{k})+\begin{pmatrix}0.0125\\ 2.2795\end{pmatrix}\mathrm{u}(\mathrm{k})$$

3．**设计降维观测器**

衰减速率是四倍，则 $\sigma_2=-18\times 4=-72$，期望极点 $z=e^{\sigma_2 T}=0.4868$

F_{11}=1，F_{12}=0.0076，F_{21}=0，F_{22}=0.5616，G_1=0.0125，G_2=2.2795

由 $F_{22}-LF_{12}=0.5616-0.0076L=0.4868$　故　$L=9.8421$

所以 $F_{21}-LF_{11}=-9.8421$，$G_2-LG_1=2.1565$

观测器方程为：$\hat{x}_2(k+1)=0.4868\hat{x}_2(k)-9.8421y(k)+2.1565u(k)+9.8421y(k+1)$

Z 变换为: $\hat{x}_2(\mathrm{z})=\dfrac{1}{z-0.4868}(9.8421(\mathrm{z}-1)\,\mathrm{Y}(\mathrm{z})-2.1565U(\mathrm{z}))$

4. 求调节器的离散控制律 D(z)=U(z)/Y(z)。

$\mathrm{u(k)}=-\mathrm{k}\,\mathbf{x}=-1.4688x_1+0.0531\hat{x}_2=-1.4688y(\mathrm{k})+0.0531\hat{x}_2(\mathrm{k})$

Z 变换得：$\mathrm{U(z)}=-1.4688Y(\mathrm{z})+0.0531\hat{x}_2(\mathrm{z})$

与 3 中方程 $\hat{x}_2(\mathrm{z})=\dfrac{1}{z-0.4868}(9.8421(\mathrm{z}-1)\,\mathrm{Y}(\mathrm{z})-2.1565U(\mathrm{z}))$ 联立：

$$D(\mathrm{z})=\frac{z-0.4868}{z-0.4868-0.0531\times 2.1565}\times\left(-1.4688+\frac{0.0531\times 9.8421(z-1)}{z-0.4868}\right)$$

化简得：$D(\mathrm{z})=\dfrac{-0.9462z+0.1924}{z-0.5833}=-0.9462\dfrac{z-0.2033}{z-0.6165}$

5. 将 D(z)进行实现，配置适当的比例因子，编制相应的程序流程图。

零极点法实现：

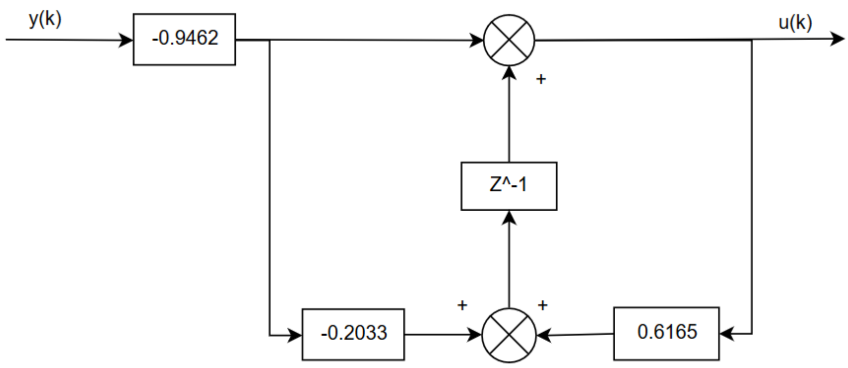


图 2.1 零极点法实现 D(z)

总的编排图如下：

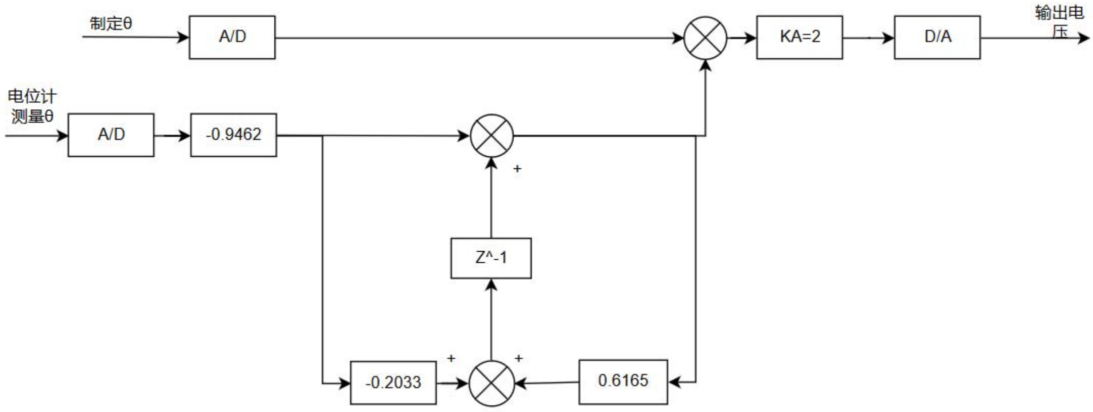


图 2.2 数字控制部分的结构图

选择比例因子：

（1） $D(z)\big|_{z\to -1} = -0.7043$， $D(z)\big|_{z\to 1} = -1.9657$，配比例因子 2

（2）系数没有超过 1，不用配。

（3） $\theta_{L\max} = 345^\circ/2 = \pm 3\,\text{rad} \approx \pm 2.5\text{rad}$

$\theta_R = \pm 2.5\,\text{rad}$

$A/D: \pm 5V \to \pm 2.5\,\text{rad} \to \pm 1$
$D/A: \pm 1 \to \pm 5V \to \pm 2.5\,\text{rad}$

故 A/D，D/A 方面不需要配比例因子

配完以后的编排图：

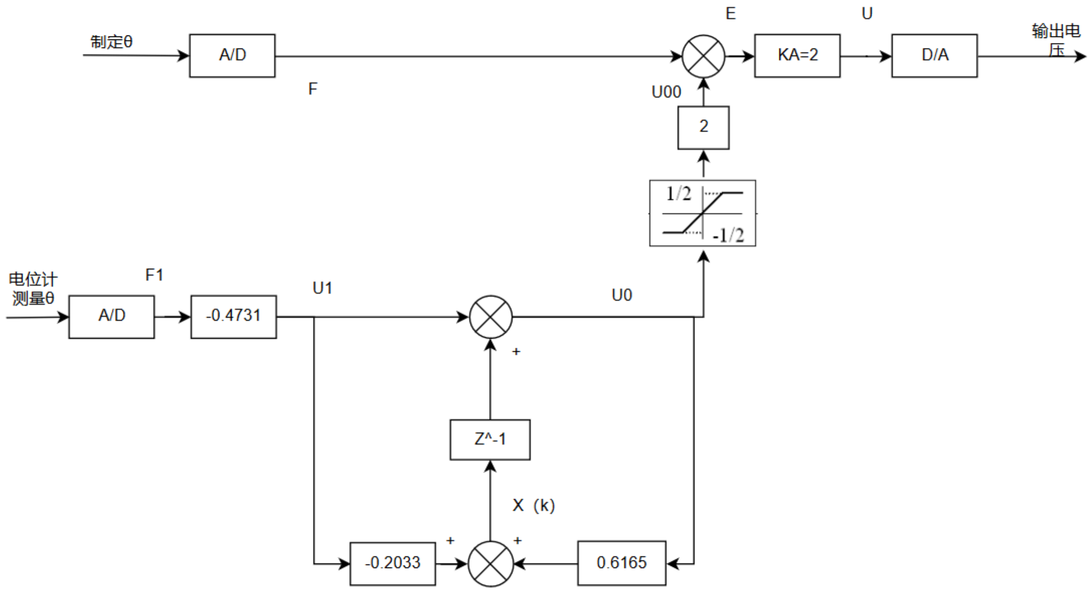


图 2.3 配置完比例因子的数字控制部分结构图

算法 1：$U_0(k)=-0.4731*F1(k)+X(k-1)$

$$U_{00}(k)=\begin{cases}1 & U_0(\mathrm{k})>\frac{1}{2}\\ 2U_0(\mathrm{k}) & |U_0(\mathrm{k})|<\frac{1}{2}\\ -1 & U_0(\mathrm{k})<-\frac{1}{2}\end{cases}$$

U(k)=F(k)+U_{00}(k)

算法 2：X(k)=0.0961*F1(k)+0.6165*U_0(k)

流程图：

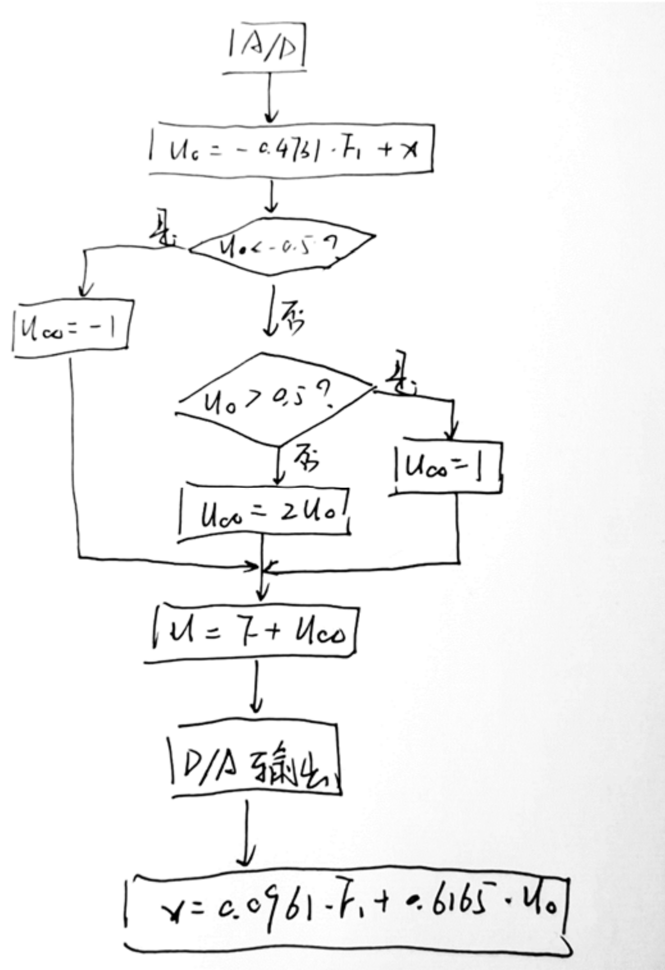


图 2.4 算法流程图

6.仿真验证调节器的控制效果

初试状态为：初速 $\omega_0=0$，初始角度 $\theta_0=10°=0.175rad$

仿真结构图如下：

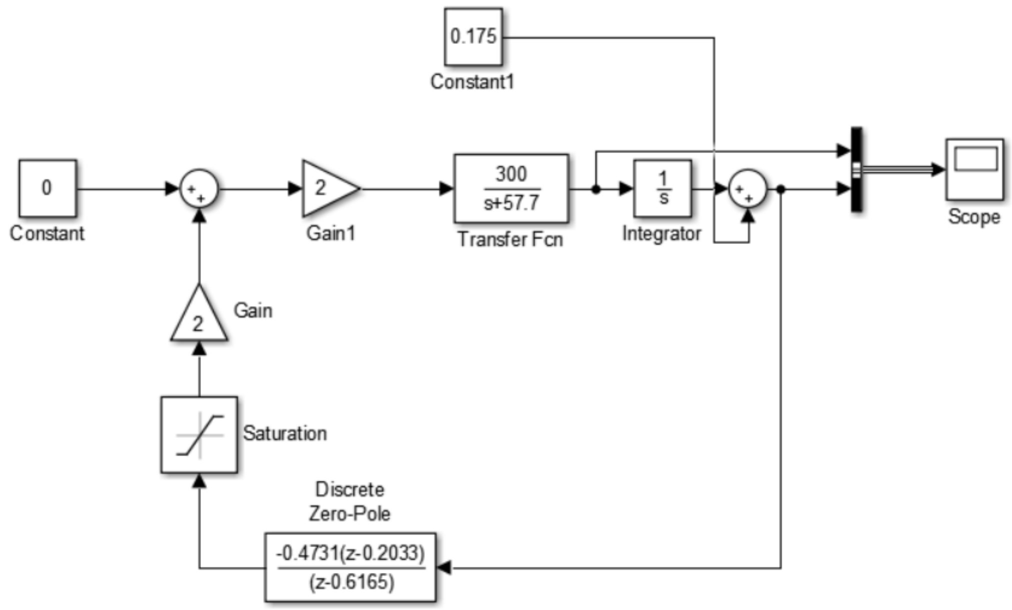


图 2.5 验证调节器效果 MATLAB 仿真结构图

仿真结果：

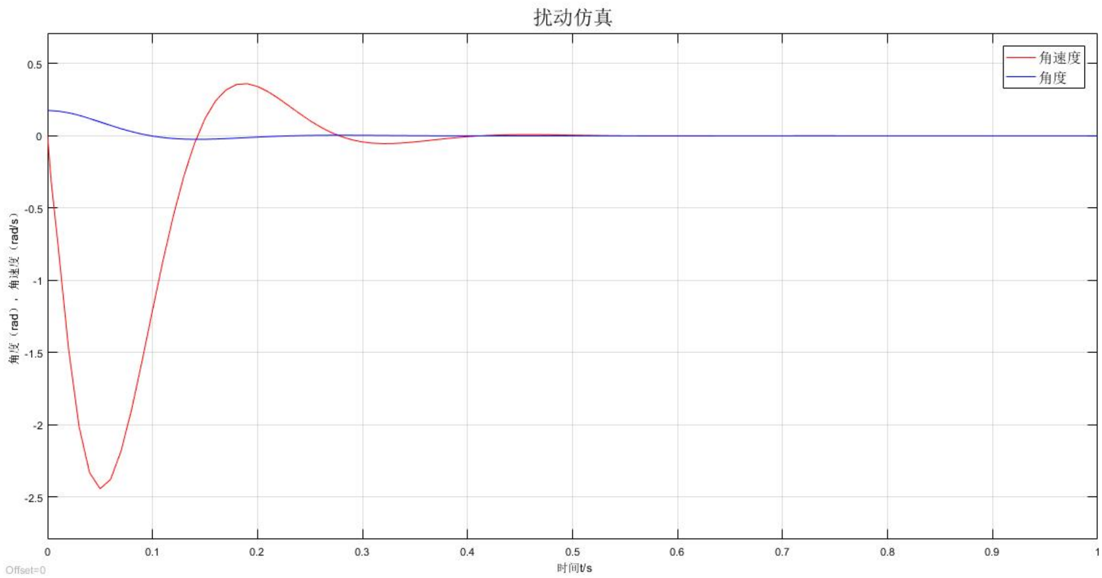


图 2.6 验证调节器效果，角度阶跃仿真图

由图可知，经过 0.2s 左右，受干扰系统即可以恢复零点。系统设计达到稳定要求。

7. 引进指令信号，设计相应的指令跟踪控制器，仿真给出闭环系统的阶跃响应曲线。

仿真结构图：

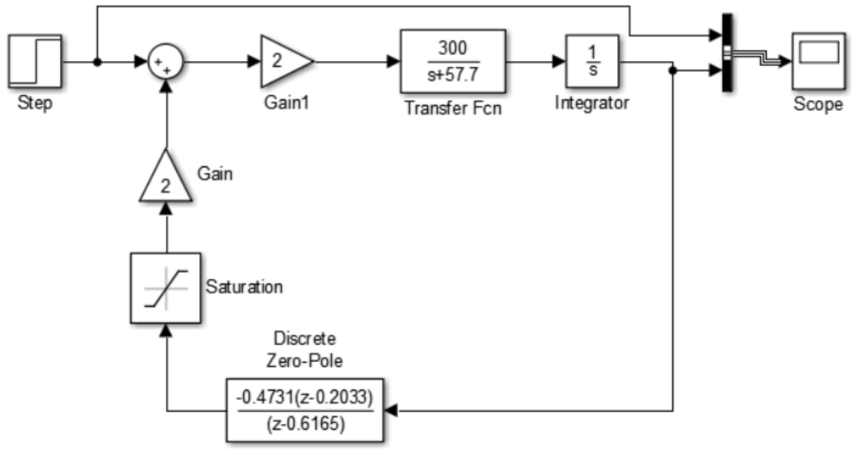


图 2.7 指令信号 MATLAB 仿真结构图

仿真结果：

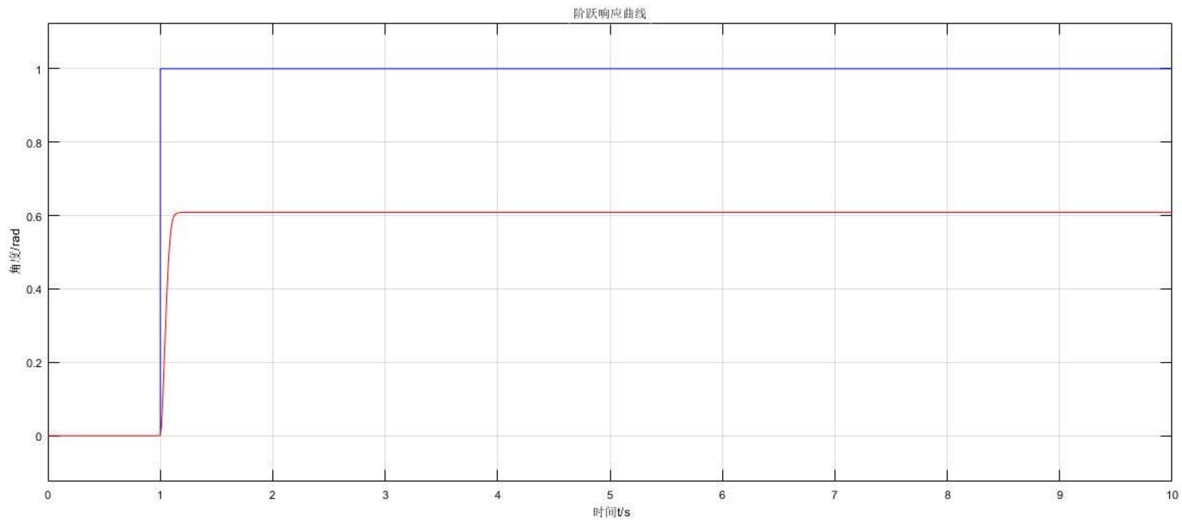


图 2.8 引进指令信号的角度阶跃仿真图

角速度：

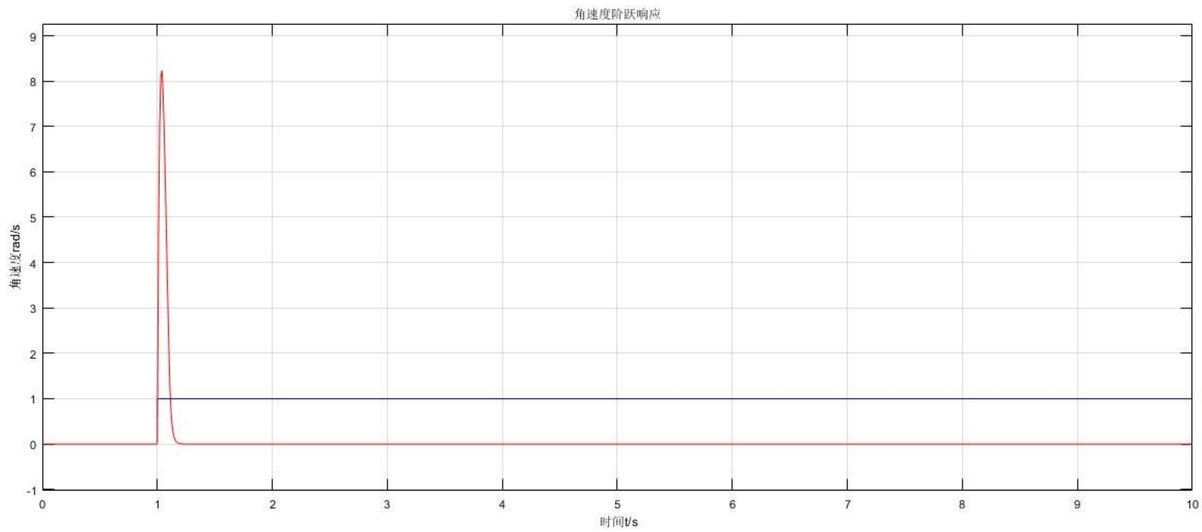


图 2.9 引进指令信号的角速度阶跃仿真图

分析仿真图，可以看到阶跃响应的调节时间约 0.2s，超调量为 0，但存在稳

态误差 0.4。稳态误差可能是由于配置 A/D 和 D/A 的系数和配置放大倍率时进行了近似而造成的。其中的原因仍需进一步探究。

MATLAB 代码：

```
clear;clc;
T=0.01;
syms s GG Gs;
A=[0 1;0 -57.7];
B=[0;300];
C=[1 0];
GG=C*inv(s*eye(2)-A)*B;
Gs=simplify(GG);

[F,G,Cd,Dd]=c2dm(A,B,C,0,T);
P=[-18+1j*20*sqrt(0.19),-18-1j*20*sqrt(0.19)];
Pd=exp(P*T);
z2=exp(-18*4*T);
k=place(F,G,Pd);

a=-1;
dz=-0.9462*(a-0.2033)/(a-0.6165)

syms z
Dz=(z-0.4868)/(z-0.4868-0.0531*2.1565)*(1.4688-
0.0531*9.8421*(z-1)/(z-0.4868));
Dz=simplify(Dz);
```