作业3

姓名：余书阳
学号：3220103741

Problem 1

> Show that the discrete-time ZOH equivalent state-space representation of the continuous-time system
>
> $$\dot{x}(t)=\begin{bmatrix}0 & \omega \\ -\omega & 0\end{bmatrix}x(t)+\begin{bmatrix}0 \\ \omega\end{bmatrix}u(t)$$
>
> is
>
> $$x(k+1)=\begin{bmatrix}\cos(\omega T) & \sin(\omega T) \\ -\sin(\omega T) & \cos(\omega T)\end{bmatrix}x(k)+\begin{bmatrix}1-\cos(\omega T) \\ \sin(\omega T)\end{bmatrix}u(k)$$

Solution

对于连续时间线性时不变系统：$\dot{x}(t)=Ax(t)+Bu(t)$,ZOH 离散化方法的状态空间表示为：$x(k+1)=A_dx(k)+B_du(k)$
其中,

$$A_d=e^{AT},\quad B_d=\int_0^T e^{A\tau}Bd\tau$$

1. **求解 A_d:**

$$(sI-A)^{-1}=\frac{1}{s^2+\omega^2}\begin{bmatrix}s & \omega \\ -\omega & s\end{bmatrix}=\begin{bmatrix}\frac{s}{s^2+\omega^2} & \frac{\omega}{s^2+\omega^2} \\ \frac{-\omega}{s^2+\omega^2} & \frac{s}{s^2+\omega^2}\end{bmatrix}$$

则,

$$e^{AT}=\mathcal{L}^{-1}\left\{(sI-A)^{-1}\right\}=\begin{bmatrix}\cos(\omega t) & -\sin(\omega t) \\ \sin(\omega t) & \cos(\omega t)\end{bmatrix}$$

带入 $t=T$ 得到离散系统矩阵：

$$A_d=e^{AT}=\begin{bmatrix}\cos(\omega T) & -\sin(\omega T) \\ \sin(\omega T) & \cos(\omega T)\end{bmatrix}$$

2. **求解 B_d:**

$$B_d=\int_0^T e^{A\tau}Bd\tau=\int_0^T\begin{bmatrix}\cos(\omega\tau) & -\sin(\omega\tau) \\ \sin(\omega\tau) & \cos(\omega\tau)\end{bmatrix}\begin{bmatrix}0 \\ \omega\end{bmatrix}d\tau$$
$$=\int_0^T\begin{bmatrix}-\omega\sin(\omega\tau) \\ \omega\cos(\omega\tau)\end{bmatrix}d\tau=\begin{bmatrix}1-\cos(\omega T) \\ \sin(\omega T)\end{bmatrix}$$

3. **最终 ZOH 离散化系统为:**

$$x(k+1)=\begin{bmatrix}\cos(\omega T) & -\sin(\omega T) \\ \sin(\omega T) & \cos(\omega T)\end{bmatrix}x(k)+\begin{bmatrix}1-\cos(\omega T) \\ \sin(\omega T)\end{bmatrix}u(k)$$

Problem 2

> For a stable and minimum-phase system
>
> $$G(z) = \frac{z-0.5}{z-1}$$
>
> a unity-feedback controller
>
> $$C(z) = K\frac{z-1}{z^n-1}\frac{z-1}{z-0.5}$$
>
> was suggested to deal with the tracking control of periodic reference input signals (period = n samples, $n > 1$).
> (a) Assuming that the stable pole/zero cancellation is exact, derive the closed-loop transfer function. (5%)
> (b) It is known that under the ideal condition, the stability bound is $0 < K < 2$, regardless of the value of n. However, when the plant inversion is not exact, this stability bound can change (usually reduce). Find out the range of K for a stable closed-loop when the actual plant is
>
> $$G(z) = \frac{z-0.7}{z-1}, \quad \text{for the case } n = 1. \quad (10\%)$$

Solution

(a)

开环传递函数 $L(z)$ 为:

$$L(z) = C(z)G(z) = K\frac{z-1}{z^n-1}\frac{z-1}{z-0.5}\frac{z-0.5}{z-1}$$

由于极点/零点对消是精确的，$(z-0.5)$ 项和其中一个 $(z-1)$ 项被消去:

$$L(z) = K\frac{z-1}{z^n-1}$$

闭环传递函数 $T(z)$ 为:

$$T(z) = \frac{L(z)}{1+L(z)} = \frac{K\frac{z-1}{z^n-1}}{1+K\frac{z-1}{z^n-1}}$$

$$\implies T(z) = \frac{K(z-1)}{(z^n-1)+K(z-1)}$$

(b)

控制器（n=1）为:

$$C(z) = K\cdot\frac{z-1}{z-1}\cdot\frac{z-1}{z-0.5} = \frac{K(z-1)}{z-0.5}$$

实际的被控对象是:

$$G(z)=\frac{z-0.7}{z-1}$$

则开环传递函数,

$$L(z)=C(z)G(z)=\frac{K(z-1)}{z-0.5}\cdot\frac{z-0.7}{z-1}=\cdot\frac{K(z-0.7)}{z-0.5}$$

闭环传递特征方程为:

$$P(z)=(z-0.5)+K(z-0.7)=0$$

得到极点

$$z_p=\frac{0.5+0.7K}{1+K}$$

为了使闭环系统稳定，该极点必须位于单位圆内部，即 $|z_p|<1$。
则只需要满足:

$$|0.5+0.7K|<|1+K|\implies(0.5+0.7K)^2<(1+K)^2\implies 0.51K^2-1.3K-0.51>0$$

则,

$$K<-\frac{5}{3}\quad\text{或}\quad K>-\frac{15}{17}$$

讨论$K=-1$：-1 位于 $(-\frac{5}{3},-\frac{15}{17})$ 之间，已经被排除。
综上,

$$K<-\frac{5}{3}\quad\text{或}\quad K>-\frac{15}{17}$$

Problem 3

> For a continuous-time lead controller
>
> $$C(s)=\frac{as+1}{s+1}$$
>
> find its discrete-time equivalent $C(z)$ using:
> (a) The Tustin approximation (10%)
> (b) The step invariant method (5%)

Solution

(a) Tustin Approximation

将 s 替换为 Tustin 变换的形式:

$$s = \frac{2}{T} \cdot \frac{z-1}{z+1}$$

则控制器变为:

$$C(z) = \frac{a \cdot \frac{2}{T} \cdot \frac{z-1}{z+1} + 1}{\frac{2}{T} \cdot \frac{z-1}{z+1} + 1}$$

整理化简，可以得到,

$$C(z) = \frac{(2a+T)z + (T-2a)}{(2+T)z + (T-2)}$$

(b) Step Invariant Method

$$\frac{C(s)}{s} = \frac{as+1}{s(s+1)} = \frac{1}{s} + \frac{a-1}{s+1}$$

连续时间的单位阶跃响应为:

$$h(t) = \mathcal{L}^{-1}\left\{\frac{1}{s} + \frac{a-1}{s+1}\right\} = \left(1 + (a-1)e^{-t}\right)u(t)$$

对其进行采样，得到

$$h(kT) = 1 + (a-1)e^{-kT}, \quad for \quad k = 0, 1, 2, \ldots$$

Z变换后得到,

$$H(z) = \frac{z}{z-1} + (a-1)\frac{z}{z-e^{-T}}$$

则离散时间控制器为:

$$C(z) = (1-z^{-1})H(z) = 1 + (a-1)\frac{z-1}{z-e^{-T}}$$

Problem 4

> For the regions in the s-plane described below, find out the corresponding region in the z-plane using MATLAB (assuming the sampling time $T = 0.1$ sec).
> (a) The rectangular region, $s = \sigma + j\omega$, $\sigma \in [-5, -20]$, $\omega \in [-6, 6]$ (10%)
> (b) The pizza-slice shaped regions (complex conjugate) governed by $\zeta \in [0.5, 0.9]$, $\omega_n \in [0, 20]$ (10%)

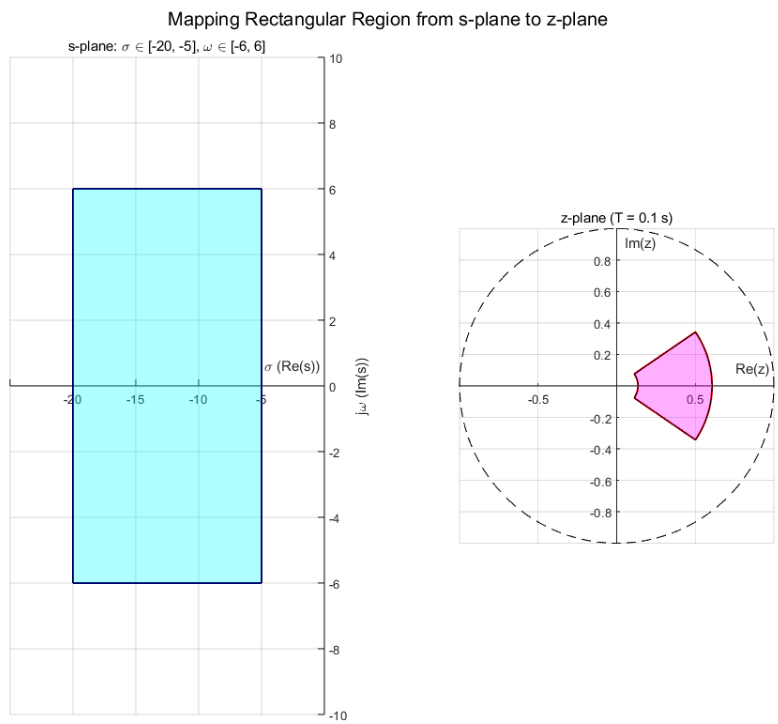
Solution

(a) Rectangular Region

```
T = 0.1;
sigma_vals = linspace(-20, -5, 100);
omega_vals = linspace(-6, 6, 100);
s_line1 = -20 + 1j*omega_vals;         % sigma = -20
s_line2 = -5 + 1j*omega_vals;          % sigma = -5
s_line3 = sigma_vals + 1j*6;           % omega = 6
s_line4 = sigma_vals - 1j*6;           % omega = -6
s_boundary_a = [s_line1, s_line3(end:-1:1), s_line2(end:-1:1), s_line4];
s1 = sigma_vals - 6j;
s2 = -5 + 1j*omega_vals;
s3 = fliplr(sigma_vals) + 6j;
s4 = -20 + 1j*fliplr(omega_vals);
z1 = exp(s1*T);
z2 = exp(s2*T);
z3 = exp(s3*T);
z4 = exp(s4*T);
figure;
subplot(1,2,1);
hold on;
plot(real(s1), imag(s1), 'b', 'LineWidth', 1.5);
plot(real(s2), imag(s2), 'b', 'LineWidth', 1.5);
plot(real(s3), imag(s3), 'b', 'LineWidth', 1.5);
plot(real(s4), imag(s4), 'b', 'LineWidth', 1.5);
% Fill the region for clarity
s_corners = [-20-6j, -5-6j, -5+6j, -20+6j];
fill(real(s_corners), imag(s_corners), 'c', 'FaceAlpha', 0.3);
grid on;
xlabel('\sigma (Re(s))');
ylabel('j\omega (Im(s))');
title('s-plane: \sigma \in [-20, -5], \omega \in [-6, 6]');
axis([-25 0 -10 10]);
ax = gca;
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
subplot(1,2,2);
hold on;
plot(real(z1), imag(z1), 'r', 'LineWidth', 1.5);
plot(real(z2), imag(z2), 'r', 'LineWidth', 1.5);
plot(real(z3), imag(z3), 'r', 'LineWidth', 1.5);
plot(real(z4), imag(z4), 'r', 'LineWidth', 1.5);
z_fill = exp([s1 s2(2:end) s3(2:end) s4(2:end-1)]*T);
fill(real(z_fill), imag(z_fill), 'm', 'FaceAlpha', 0.3);
grid on;
xlabel('Re(z)');
ylabel('Im(z)');
title(['z-plane (T = ', num2str(T), ' s)']);
axis equal;
th = 0:pi/50:2*pi;
plot(cos(th), sin(th), 'k--');
ax = gca;
ax.XAxisLocation = 'origin';
```

```
ax.YAxisLocation = 'origin';

sgtitle('Mapping Rectangular Region from s-plane to z-plane');
```

得到以下结果:



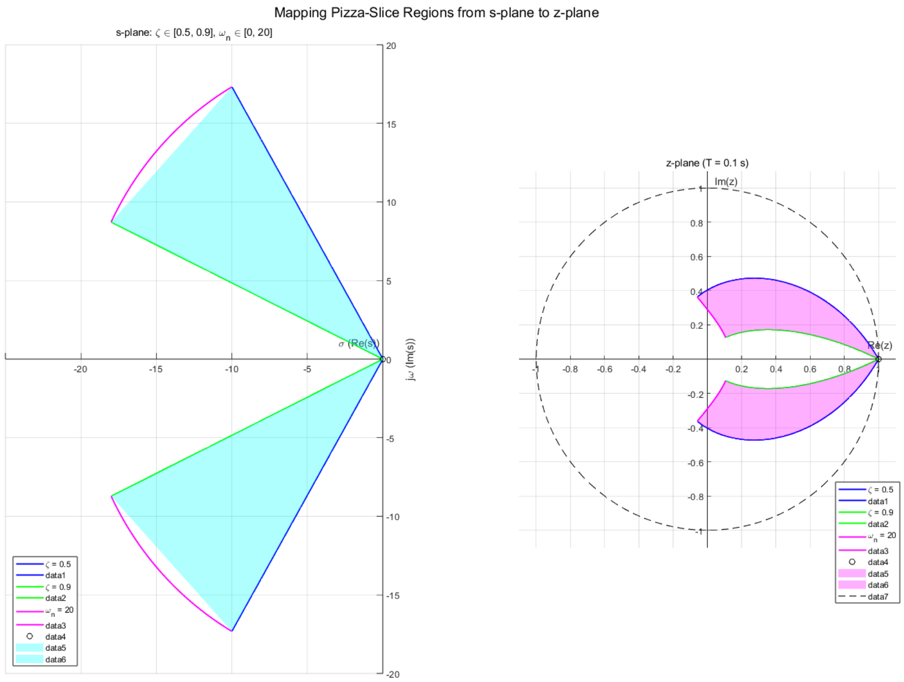
>

(b) Pizza-Slice Shaped Regions

```
T = 0.1;

zeta_vals = [0.5, 0.9];
omega_n_vals = linspace(0, 20, 200); % From 0 to 20 rad/s
s_zeta1_pos = -zeta_vals(1)*omega_n_vals + 1j*omega_n_vals*sqrt(1-zeta_vals(1)^2);
s_zeta1_neg = -zeta_vals(1)*omega_n_vals - 1j*omega_n_vals*sqrt(1-zeta_vals(1)^2);
s_zeta2_pos = -zeta_vals(2)*omega_n_vals + 1j*omega_n_vals*sqrt(1-zeta_vals(2)^2);
s_zeta2_neg = -zeta_vals(2)*omega_n_vals - 1j*omega_n_vals*sqrt(1-zeta_vals(2)^2);
zeta_arc_vals = linspace(zeta_vals(1), zeta_vals(2), 100);
s_omega_n_max_pos = -zeta_arc_vals*20 + 1j*20*sqrt(1-zeta_arc_vals.^2);
s_omega_n_max_neg = -zeta_arc_vals*20 - 1j*20*sqrt(1-zeta_arc_vals.^2);
s_origin = 0;
z_zeta1_pos = exp(s_zeta1_pos*T);
z_zeta1_neg = exp(s_zeta1_neg*T);
z_zeta2_pos = exp(s_zeta2_pos*T);
z_zeta2_neg = exp(s_zeta2_neg*T);
z_omega_n_max_pos = exp(s_omega_n_max_pos*T);
z_omega_n_max_neg = exp(s_omega_n_max_neg*T);
z_origin = exp(s_origin*T); % This will be z=1
```

```matlab
figure;
subplot(1,2,1);
hold on;
plot(real(s_zeta1_pos), imag(s_zeta1_pos), 'b', 'LineWidth', 1.5, 'DisplayName', '\zeta = 0.5');
plot(real(s_zeta1_neg), imag(s_zeta1_neg), 'b', 'LineWidth', 1.5);
plot(real(s_zeta2_pos), imag(s_zeta2_pos), 'g', 'LineWidth', 1.5, 'DisplayName', '\zeta = 0.9');
plot(real(s_zeta2_neg), imag(s_zeta2_neg), 'g', 'LineWidth', 1.5);
plot(real(s_omega_n_max_pos), imag(s_omega_n_max_pos), 'm', 'LineWidth', 1.5, 'DisplayName', '\omega_n = 20');
plot(real(s_omega_n_max_neg), imag(s_omega_n_max_neg), 'm', 'LineWidth', 1.5);
plot(real(s_origin), imag(s_origin), 'ko'); % Origin
% Fill the region
fill([real(s_zeta1_pos) real(s_omega_n_max_pos(end:-1:1)) real(fliplr(s_zeta2_pos))], ...
     [imag(s_zeta1_pos) imag(s_omega_n_max_pos(end:-1:1)) imag(fliplr(s_zeta2_pos))], 'c', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
fill([real(s_zeta1_neg) real(s_omega_n_max_neg(end:-1:1)) real(fliplr(s_zeta2_neg))], ...
     [imag(s_zeta1_neg) imag(s_omega_n_max_neg(end:-1:1)) imag(fliplr(s_zeta2_neg))], 'c', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
grid on;
xlabel('\sigma (Re(s))');
ylabel('j\omega (Im(s))');
title('s-plane: \zeta \in [0.5, 0.9], \omega_n \in [0, 20]');
legend('show', 'Location', 'southwest');
axis([-25 0 -20 20]);
ax = gca;
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
subplot(1,2,2);
hold on;
plot(real(z_zeta1_pos), imag(z_zeta1_pos), 'b', 'LineWidth', 1.5, 'DisplayName', '\zeta = 0.5');
plot(real(z_zeta1_neg), imag(z_zeta1_neg), 'b', 'LineWidth', 1.5);
plot(real(z_zeta2_pos), imag(z_zeta2_pos), 'g', 'LineWidth', 1.5, 'DisplayName', '\zeta = 0.9');
plot(real(z_zeta2_neg), imag(z_zeta2_neg), 'g', 'LineWidth', 1.5);
plot(real(z_omega_n_max_pos), imag(z_omega_n_max_pos), 'm', 'LineWidth', 1.5, 'DisplayName', '\omega_n = 20');
plot(real(z_omega_n_max_neg), imag(z_omega_n_max_neg), 'm', 'LineWidth', 1.5);
plot(real(z_origin), imag(z_origin), 'ko'); % Mapped origin (z=1)

fill([real(z_zeta1_pos) real(z_omega_n_max_pos(end:-1:1)) real(fliplr(z_zeta2_pos))], ...
     [imag(z_zeta1_pos) imag(z_omega_n_max_pos(end:-1:1)) imag(fliplr(z_zeta2_pos))], 'm', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
fill([real(z_zeta1_neg) real(z_omega_n_max_neg(end:-1:1)) real(fliplr(z_zeta2_neg))], ...
     [imag(z_zeta1_neg) imag(z_omega_n_max_neg(end:-1:1)) imag(fliplr(z_zeta2_neg))], 'm', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
grid on;
xlabel('Re(z)');
ylabel('Im(z)');
title(['z-plane (T = ', num2str(T), ' s)']);
legend('show', 'Location', 'southeast');
axis equal;
xlim([-1.1 1.1]); ylim([-1.1 1.1]);

th = 0:pi/50:2*pi;
plot(cos(th), sin(th), 'k--');
ax = gca;
ax.XAxisLocation = 'origin';
ax.YAxisLocation = 'origin';
sgtitle('Mapping Pizza-Slice Regions from s-plane to z-plane');
```

得到以下结果：



Mapping Pizza-Slice Regions from s-plane to z-plane
s-plane: ζ ∈ [0.5, 0.9], ωn ∈ [0, 20]
σ (Re(s))
jω (Im(s))
ζ = 0.5
data1
ζ = 0.9
data2
ωn = 20
data3
data4
data5
data6
z-plane (T = 0.1 s)
Im(z)
Re(z)
ζ = 0.5
data1
ζ = 0.9
data2
ωn = 20
data3
data4
data5
data6
data7