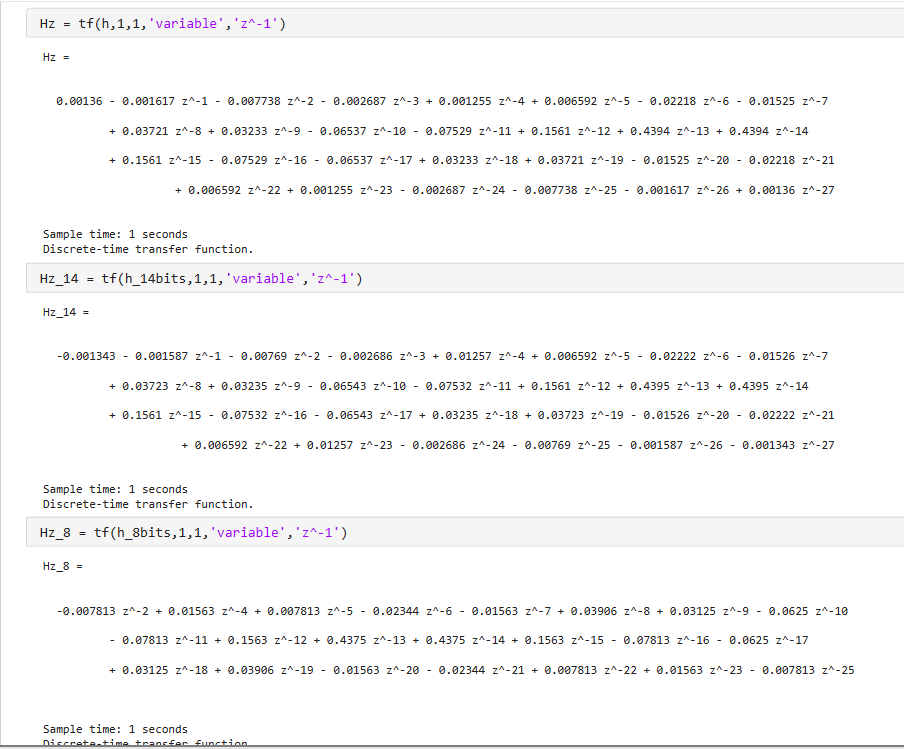
1. 未量化(32位量化)，14位量化，8位量化系数求系统函数；

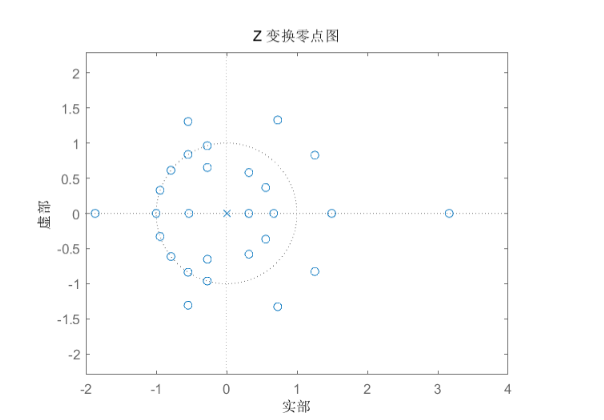
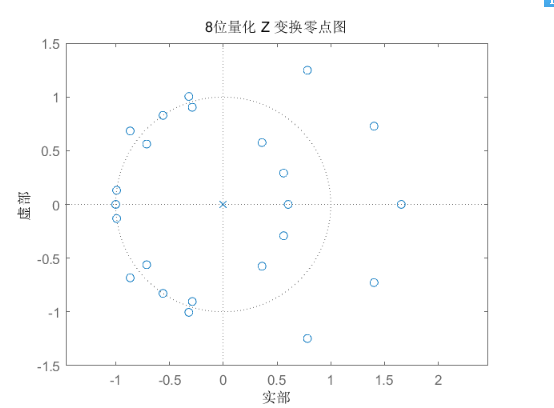
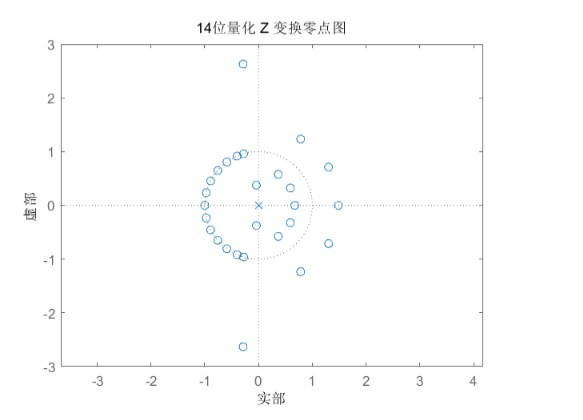
图：



代码：

1. format long
2. h1 = [1.359657e-3,-1.616993e-3,-7.738032e-3,-2.686841e-3,1.255246e-3,...
3. 6.591530e-3,-2.217952e-2,-1.524663e-2,3.720668e-2,3.233332e-2,-6.537057e-2,-7.528754e-2,...
4. 1.560970e-1,4.394094e-1];
5. h2 = flip(h1);
6. h = [h1,h2];
7. %14bits
8. format long
9. h1 = [-11,-13,-63,-22,103,54,-182,-125,305,265,-536,-617,1279,3600]\*2^(-13);
10. h2 = flip(h1);
11. h\_14bits = [h1,h2];
12. %8bits
13. format long
14. h1 = [0,0,-1,0,2,1,-3,-2,5,4,-8,-10,20,56]\*2^(-7);
15. h2 = flip(h1);
16. h\_8bits = [h1,h2];
17. % 第一题
18. syms z
19. n = 0:27;
20. Hz = sum(h.\*z.^(-n));
21. Hz\_8 = sum(h\_8bits.\*z.^(-n));
22. Hz\_14 = sum(h\_14bits.\*z.^(-n));
23. 系统函数，求出零点（分母设为z-27 ），并画零点图；

图：



代码：

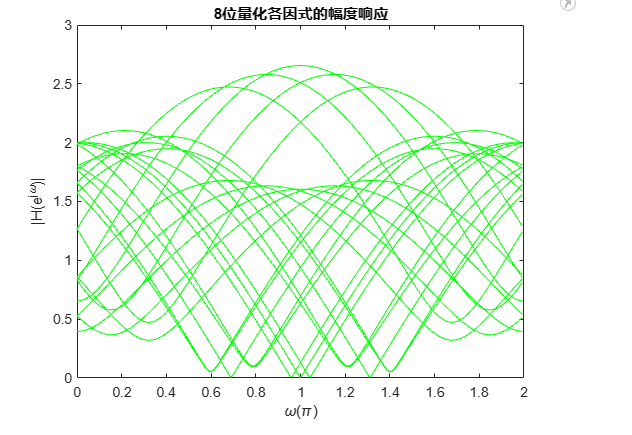
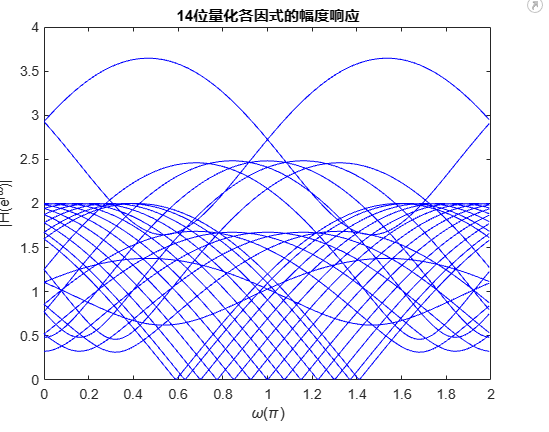
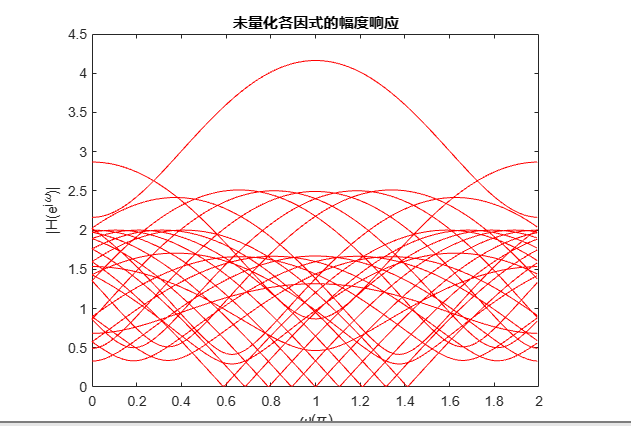
1. zero\_n = roots(h);
2. zero\_14 = roots(h\_14bits);
3. zero\_8 = roots(h\_8bits);
4. %plot the map
5. figure;
6. zplane(h,1);
7. title('Z 变换零点图');
8. xlabel('实部');
9. ylabel('虚部');
10. figure;
11. zplane(h\_14bits,1);
12. title('14位量化 Z 变换零点图');
13. xlabel('实部');
14. ylabel('虚部');
15. figure;
16. zplane(zero\_8,1);
17. title('8位量化 Z 变换零点图');
18. xlabel('实部');
19. ylabel('虚部');
20. 零点，求出系统函数因式形式；

图：



1. %no quantilize
2. tf\_n = 1;
3. for n = 1:27
4. tf\_n = (z-zero\_n(n)).\*tf\_n;
5. end
6. tf\_n
7. %14bits
8. tf\_14 = 1;
9. for n = 1:27
10. tf\_14 = (z-zero\_14(n)).\*tf\_14;
11. end
12. tf\_14
13. %8bits
14. tf\_8 = 1;
15. for n = 1:25
16. tf\_8 = (z-zero\_8(n)).\*tf\_8;
17. end
18. tf\_8
19. 系统函数各子因式，画出各因式对应的幅度响应|Hk(ejω)|；

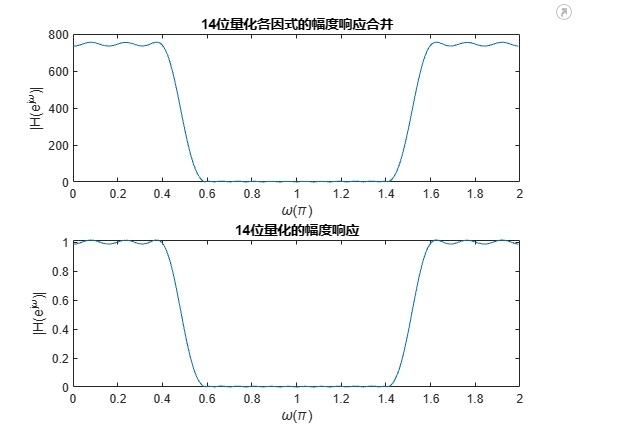
图：



1. % no quantilize
2. figure;
3. clear title;
4. H\_factor = zeros(27,256);
5. for i = 1:27
6. [H\_subfactor,w] = freqz([1 -zero\_n(i)],1,256,'whole');
7. H\_factor(i,:) = H\_subfactor';
8. end
9. for j = 1:27
10. plot(w,abs(H\_factor(j,:)));
11. hold on;
12. end
13. title('未量化')
14. figure;
15. H\_factor = zeros(25,256);
16. for i = 1:25
17. [H\_subfactor,w] = freqz([1 -zero\_14(i)],1,256,'whole');
18. H\_factor(i,:) = H\_subfactor';
19. end
20. for j = 1:25
21. 8
22. plot(w,abs(H\_factor(j,:)));
23. hold on;
24. end
25. title('十四位量化')
26. figure;
27. H\_factor = zeros(25,256);
28. for i = 1:25
29. [H\_subfactor,w] = freqz([1 -zero\_8(i)],1,256,'whole');
30. H\_factor(i,:) = H\_subfactor';
31. end
32. for j = 1:25
33. plot(w,abs(H\_factor(j,:)));
34. hold on;
35. end
36. title('八位量化')

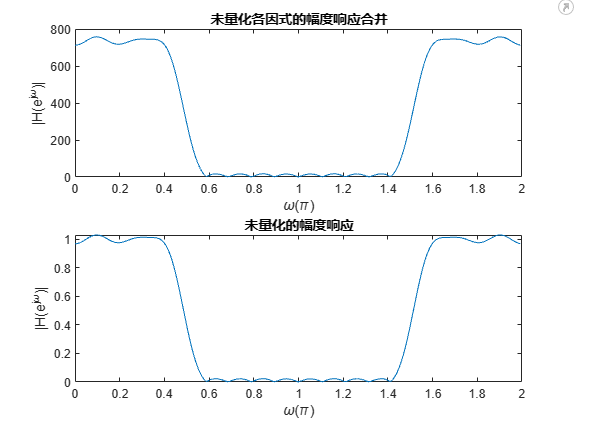
⑤画出 ④生成的由14位量化各因式幅度响应合并的系统幅度响应|H14(ejω)|，并与由①中求得的系统函数直接获得的幅度响应进行比较，给出结论描述；

图：



1. H\_factor\_14\_product = prod(abs(H\_factor\_14),1);
2. figure(7);
3. subplot(2,1,1)
4. plot(w/pi,abs(H\_factor\_14\_product));
5. title('14位量化各因式的幅度响应合并');
6. xlabel('\omega(\pi)');
7. ylabel ('|H(e^j^\omega)|');
8. subplot(2,1,2)
9. [H\_f14,~] = freqz(h\_14bits,1,256,'whole');
10. plot(w/pi,abs(H\_f14));
11. title('14位量化的幅度响应');
12. xlabel('\omega(\pi)');
13. ylabel ('|H(e^j^\omega)|')
14. 对未量化系统，重做步骤⑤过程，并与14位量化幅度响应进行比较，并给出结论描述。

图



代码:

1. H\_factor\_n\_product = prod(abs(H\_factor),1);
2. figure(8);
3. subplot(2,1,1)
4. plot(w/pi,abs(H\_factor\_n\_product));
5. title('未量化各因式的幅度响应合并');
6. xlabel('\omega(\pi)');
7. ylabel ('|H(e^j^\omega)|');
8. subplot(2,1,2)
9. [H,~] = freqz(h,1,256,'whole');
10. plot(w/pi,abs(H));
11. title('未量化的幅度响应');
12. xlabel('\omega(\pi)');
13. ylabel ('|H(e^j^\omega)|');