1) https://github.com/eriklovekin
2) The residuals for both (a) and (b) are approximately 0.
3) The error in the function definition is from publishing without inputs. As can be seen from the bode plots, the MATLAB and custom responses are similar, but the custom response is exact at the critical omega where as MATLAB is accurate at omega=0.

Table of Contents

HW1 Problem 2

```
b = RR_poly([2,-2,5,-5],1);
a = RR_poly([-1 1 -3 3 -6 6],1);
f = RR_poly([-1 -1 -3 -3 -6 -6],1);
```

(a)

```
[x,y] = RR_diophantine(a,b,f);
test = trim(a*x+b*y);
residual = norm(f-test)% test that answer is correct (should = 0)
```

```
residual =
   1.1369e-13
```

(b)

```
f_roots = RR_roots(f);
k = 6;
f2 = RR_poly([f_roots -20*ones(1,k)],1);% Add k poles to make y/x proper
[x2,y2] = RR_diophantine(a,b,f2);
test2 = trim(a*x2+b*y2);
residual2 = norm(f2-test2)% test result is correct (should = 0)
```

```
residual2 =
   1.9457e-04
```

Published with MATLAB® R2023a

ESL_C2D_matched

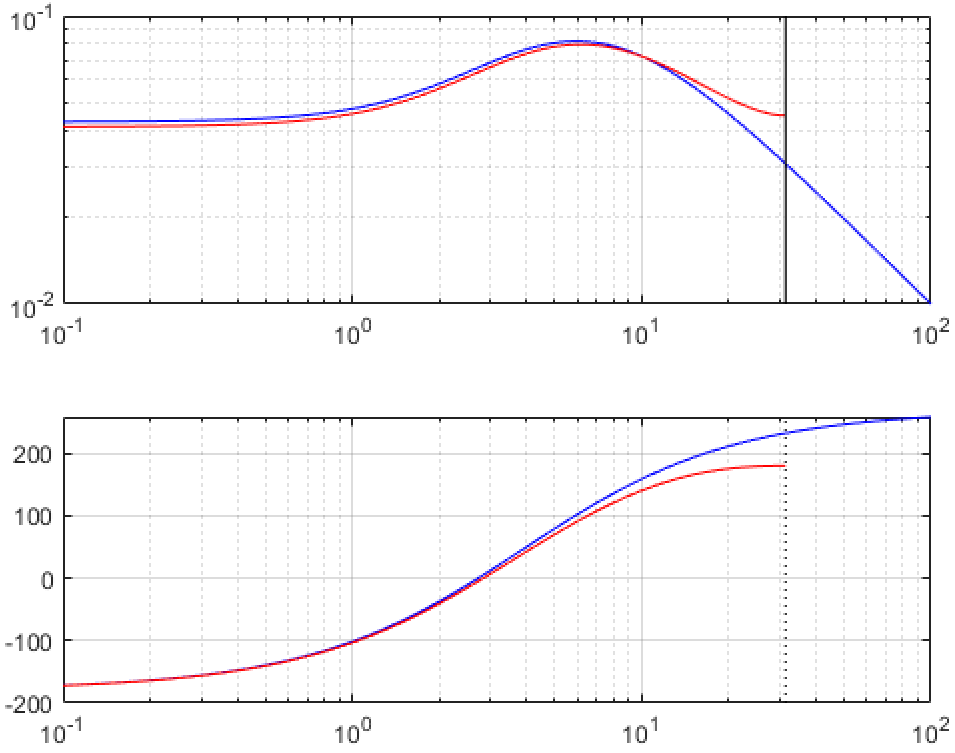
```
function [Gz] = ESL_C2D_matched(Ds,h,omegac,proper)
    % function [Gz] = ESL_C2D_matched(Ds,h)
    % Convert Ds(s) to Dz(z) using matched method with timestep h.
    % Match gain at default omega = 0 or at critical gain omegac.

    if nargin < 4 % Assume proper
        proper = true;
    elseif nargin < 3 % Assume desired omegac = 0
        omegac = 0;
        proper = true;
    end

    Gz = RR_tf(exp(Ds.z*h),exp(Ds.p*h),1);% Initialize Gz
    Gz.h = h;

    nz_inf = length(Gz.p)-length(Gz.z);% # Zeros at infinity = # poles - #
 zeros

    Gz.z = [Gz.z -1*ones(1,nz_inf-1)];% Add poles at z=-1

    if proper
        Gz.z = [Gz.z Inf(1)];% Final pole at inf
    else
        Gz.z = [Gz.z -1];% Final pole at -1
    end

    Ks = abs(RR_evaluate(Ds,omegac*1i));% Continuous gain at omegac
    Kz = abs(RR_evaluate(Gz,exp(omegac*1i*h)));% Discrete gain at omegac

    Gz = Gz*Ks/Kz;% Scale discrete gain
end
```

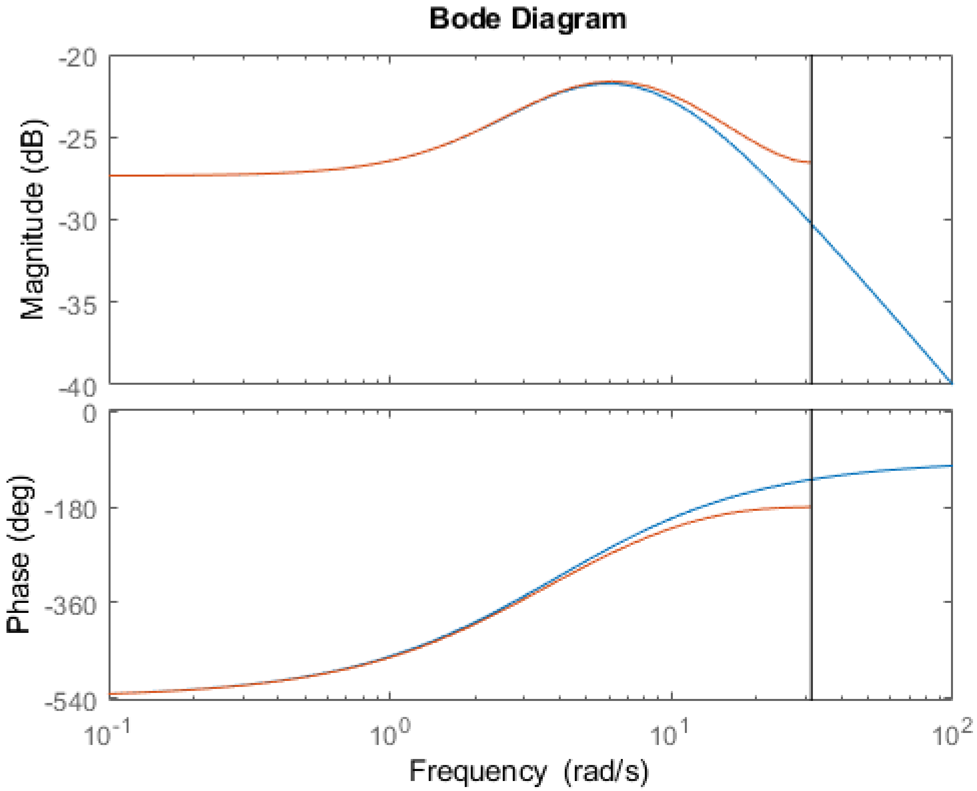
```
Not enough input arguments.
Error in ESL_C2D_matched (line 14)
    Gz = RR_tf(exp(Ds.z*h),exp(Ds.p*h),1);% Initialize Gz
```

Published with MATLAB® R2023a

ESL_C2D_matched test

```
close all
num = [1 5 6];
den = [1 -16 83 -140];
h = 0.1;% Time step size
omegac = 10;% critical frequency

Ds = RR_tf(num,den);% Define continuous functions
Ms = tf(num,den);

Dz = ESL_C2D_matched(Ds,h,omegac); %discretize
Mz = c2d(Ms,h,'matched'); %discretize with MATLAB method

figure(1)
hold on
RR_bode(Ds)
RR_bode(Dz)
hold off

figure(2)
hold on
bode(Ms)
bode(Mz)
hold off
```





Published with MATLAB® R2023a