

---

```
function [] = FOEB(z,p,k)

% We declare the transfer function
H = zpkm(-z,-p,k);

% We find the number of poles and zeros
np = length(p);
nz = length(z);

% We count the 0 values from the poles and zeros
cz = 0;
for i = 1:length(z)
    if eq(z(i),0)
        cz = cz + 1;
    end
end

cp = 0;
for i = 1:length(p)
    if eq(p(i),0)
        cp = cp + 1;
    end
end

% We sort the corner w
wcf = sort([z p]);

slope = [];
sum = 0;

phi = "";
for i = 1:length(wcf)

    if ismember(wcf(i),z) == logical(1)

        % We calculate the slope if the wcf is a zero
        sum = sum + 20;
        slope(i) = sum;

        % We calculate the initial gain K if the wcf is a zero
        if(wcf(i)==0)
            k = k * 1;
        else
            k = k * wcf(i);
        end

        % We construct the phase equation for zero
        elem = strcat("+atan(x/",num2str(wcf(i)),""));

    else

        % We calculate the slope if the wcf is a pole
        sum = sum - 20;
```

---

---

```

    slope(i) = sum;

    % We calculate the initial gain K if the wcf is a pole
    if(wcf(i)==0)
        k = k / 1;
    else
        k = k / wcf(i);
    end

    % We construct the phase equation for pole
    elem = strcat("-atan(x/",num2str(wcf(i)),"");
end

% We construct the powers of 10 for the samples
if wcf(i) == 0
    n(i) = floor(log10(1e-5));
else
    n(i) = floor(log10(wcf(i)));
end

% We finish building the phase equation
phi = string(phi + elem);
end

% We check to see if we have a pole or zero in origin to initialize the
% slope
if ismember(0,z)
    slope = [20 slope slope(length(slope))];
elseif ismember(0,p)
    slope = [-20 slope slope(length(slope))];
else
    slope = [0 slope];
end

phi = strcat("@(x)",phi);
n = unique(n(:).');

% We construct the logspace to have a "nice plot"
if ismember(0,wcf)
    lsp = logspace(0,(n(length(n))+2),1e2);
else
    lsp = logspace((n(1)-1),(n(length(n))+2),1e2);
end

% We construct the general wma vector for both plots
if ismember(0,wcf)
    indices = find(wcf<1);
    wcf(indices) = [];
    wma = [1 wcf 10^(n(length(n))+1) 10^(n(length(n))+2)];
else
    wma = [10^(n(1)-1) wcf 10^(n(length(n))+2)];
end

```

---

---

```

% We construct the magnitude vector for the magnitude plot
ma = [];

ma(1) = 20*log10(k);

for i = 2:length(wma)
    ma(i) = ma(i-1) + slope(i-1)*log10(wma(i)/wma(i-1));
end

% We construct the phase vector for the phase plot
fa = [];
for i = 1:length(wma)
    p = str2func(phi) ;
    fa(i) = rad2deg(p(wma(i)));
end

% We plot the real bode plot alongside the Approx. bode plot
[mag,fre] = bode(H,lsp);
mv(1:le2,1) = mag(:, :, :);
fv(1:le2,1) = (fre(:, :, :));
subplot(211);
semilogx(lsp, 20*log10(mv), 'b', wma, ma, 'ro-'); grid

title('Magnitude characteristics');
xlabel('\omega (rad/sec)'); ylabel('|H(j/\omega)|^dB')

subplot(212);
semilogx(lsp, fv, 'b', wma, fa, 'ro-'); grid
title('Phase characteristics');
xlabel('\omega (rad/sec)'); ylabel('\angle H(j\omega) (degrees)');

end

```

*Published with MATLAB® R2023a*