Main script (Question-2)

Table of Contents

Basic Variables

```
tb=2; % seperation between successive transmitted pulses
fb=1/tb;%  freq of seperation between successive transmitted pulses
td=0.01; % time axis jump
ts=10; % Time axis limit
fd=0.01;% time axis jump
fs=2*fb; %freq axis limit
f=0:fd:fs; % freq axis
T=-1*ts:td:ts; % Time axis
```

Generating outputs

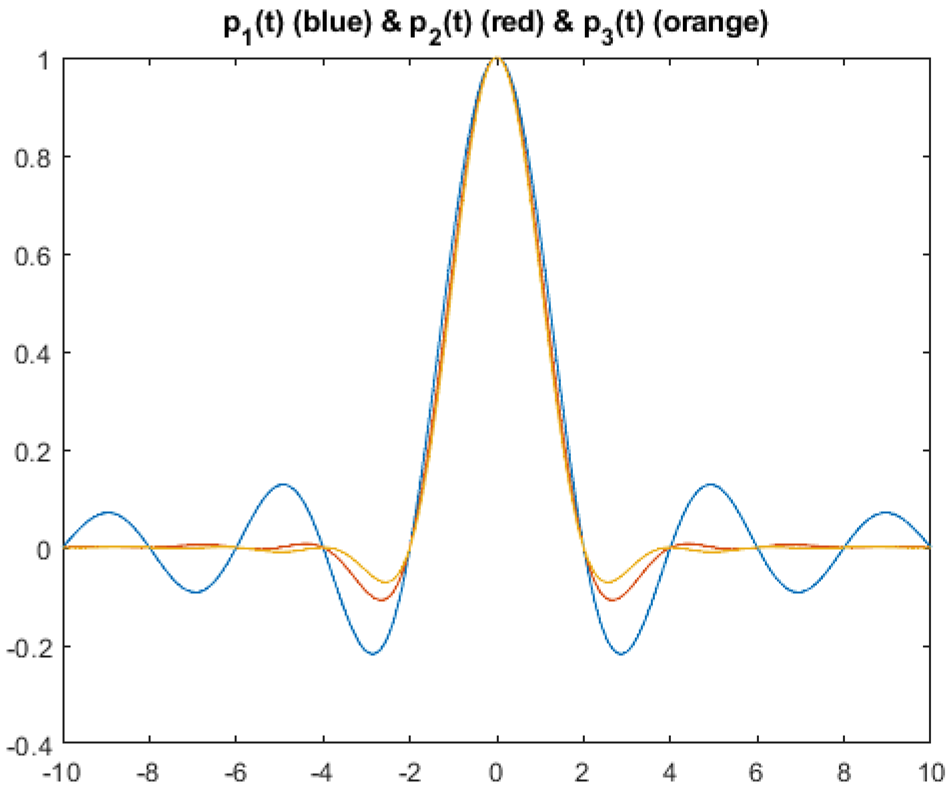
p1 -> r=0 p2 -> r=0.6 p3 -> r=0.75

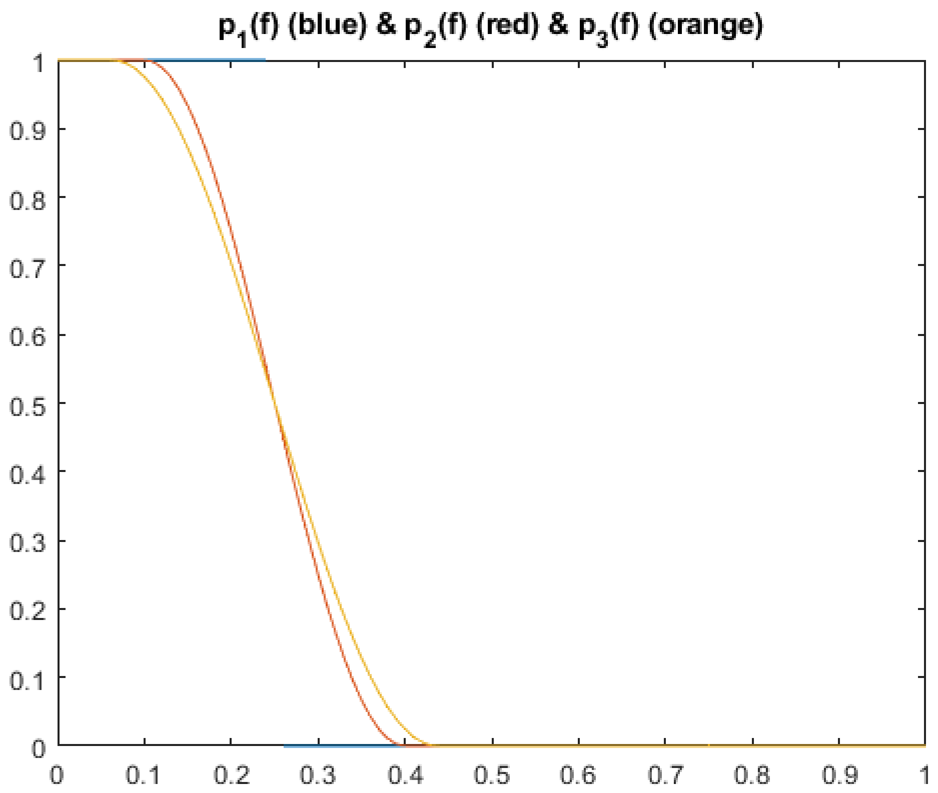
```
p1=p_timeGenerator(T,0,tb,td,ts);
p2=p_timeGenerator(T,0.6,tb,td,ts);
p3=p_timeGenerator(T,0.75,tb,td,ts);
% Above ones were Time domain Outputs

% Below ones are freq domain Outputs
p1f=p_freqGenerator(f,0,fb,fd,fs);
p2f=p_freqGenerator(f,0.6,fb,fd,fs);
p3f=p_freqGenerator(f,0.75,fb,fd,fs);
```

Ploting the Results

```
figure('Name',"part-1");
plot(T,p1);
hold on;
plot(T,p2);
plot(T,p3);
hold off;
```

```
title("p_1(t) (blue) & p_2(t) (red) & p_3(t) (orange)");
figure('Name',"part-2");
plot(f,p1f);
hold on;
plot(f,p2f);
plot(f,p3f);
hold off;
title("p_1(f) (blue) & p_2(f) (red) & p_3(f) (orange)");
```





Aditional Bandwidth (fx=Rb*r/2)

```
disp("Aditional Bandwidth (fx)")
disp("Case-1 ->  r=0 : fx = "+ fb*0/2);
disp("Case-2 ->  r=0.6 : fx = "+ fb*0.6/2);
disp("Case-3 ->  r=0.75 : fx = "+ fb*0.75/2);
disp("");
disp("Thus we see as r increases fx increases");
disp(" Hence min fx is for r=0 and max fx is for r=0.75");
```

```
Aditional Bandwidth (fx)
Case-1 ->  r=0 : fx = 0
Case-2 ->  r=0.6 : fx = 0.15
Case-3 ->  r=0.75 : fx = 0.1875
Thus we see as r increases fx increases
 Hence min fx is for r=0 and max fx is for r=0.75
```

Function to generate time domain p(t)

Function below is a logic to implement p(t) where the P(f) formula has been taken from B.P.Lathi Textbook and p(t) is the Fourier Transform of P(f). Also I am attaching the reference of the research Paper from where I have Taken p(t) formula.

```
function p=p_timeGenerator(T,r,tb,td,ts)
p=T;
```

```
for i = 1:length(T)
    m=(i-1)*td/tb;
    m=m-(ts/tb);
    if(m==(-1/(2*r))||m==(1/(2*r)))
        p(i)=pi*(sin(pi/(2*r))/(pi/(2*r)))/(4*tb);
    else
        p(i)=((sin(pi*m)/(pi*m))*cos(r*pi*m))/(1-(2*r*m)^2);
    end
end
end
```

Function to generate freq domain P(f)

Function below is a logic to implement P(f) where the P(f) formula has been taken from B.P.Lathi Textbook.

```
function p=p_freqGenerator(f,r,fb,fd,fs)
p=f;
for i = 1:length(f)
    m=(i-1)*fd;
%     m=m-fs;
    if(abs(m)<((fb/2)*(1-r)))
        p(i)=1;
    else
        if(abs(m)>((fb/2)*(1+r)))
            p(i)=0;
        else
            p(i)=(1-sin(pi*((m-(fb/2))/(fb*r))))/2;
        end
    end
end
end
```

References

1. https://en.wikipedia.org/wiki/Raised-cosine_filter

2. https://www.nonstopsystems.com/radio/pdf-hell/article-raised-cosine.pdf

End of the Question

Published with MATLAB® R2022b