EE 313 Linear Signals & Systems (Fall 2017)

Solution Set for Mini-Project #1 on Multipath Fading

Anyesha Ghosh & Prof. Brian L. Evans

Mini-Project #1 is to examine the effects of multipath fading on a wireless communications signal. The assignment asks one to calculate the signal strength of the receiver as a function of the location of the receiver in a vehicle. On a smart phone, the received signal strength is shown graphically. Traditionally, the received signal strength indicator (RSSI) uses vertical signal bars using five levels (0-4 bars). More recent operating systems display the RSSI as a continual value using a ramp shape.

The assignment is from the following reference:

> J. H. McClellan, R. W. Schafer & M. A Yoder, "Section 6 Lab Exercise: Multipath Fading",
> *Lab P-2: Introduction to Complex Exponentials: Multipath*,
> http://dspfirst.gatech.edu/chapters/DSP1st2eLabs/MultipathLab.pdf

The first paragraph below and the figure that follows it are from the above reference.

> In a mobile radio system (e.g., cell phones or AM radio), there is one type of degradation that is a common problem. This is the case of *multipath fading* caused by reflections of the radio waves which interfere destructively at some locations. Consider the scenario diagrammed in Fig. 1 where a vehicle traveling on the roadway receives signals from two sources: directly from the transmitter and reflections from another object such as a large building. This multipath problem can be modeled easily with sinusoids. The total received signal at the vehicle is the sum of two signals which are themselves delayed versions of the transmitted signal, *s(t)*.
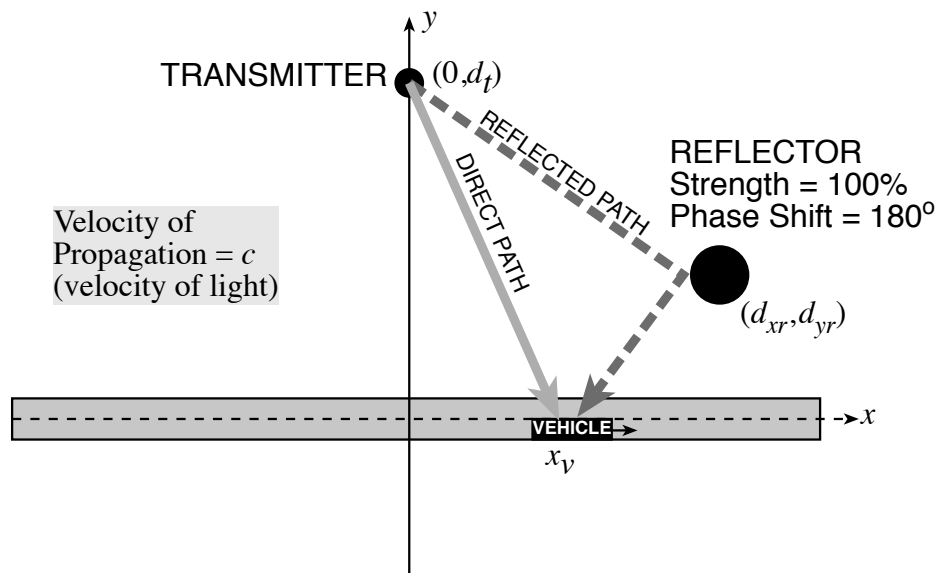

Figure 1: Scenario for multipath in mobile radio. A vehicle traveling on the roadway (to the right) receives signals from two sources: the transmitter and a reflector located at (d_{xr}, d_{yr}).

Assume that the speed of propagation in the medium is the speed of light, *c*.

a) $t_1(x_v)$ = distance between transmitter & vehicle/c

$$= \frac{\sqrt{d_t^2 + x_v^2}}{c}.$$

b) $t_2(x_v)$ = distance between transmitter & vehicle via reflector/c

$$= \frac{\sqrt{(d_t-d_{yr})^2+d_{xr}{}^2}+\sqrt{(d_{xr}-x_v)^2+d_{yr}{}^2}}{c}.$$

c) $r_v(t) = s(t - t_1) - s(t - t_2)$
$$= \cos\left(2\pi f_0(t - t_1)\right) - \cos\left(2\pi f_0(t - t_2)\right).$$

The reported maximum value after running the code below is 0.5737.

```
f0 = 150*10^6;
xv = 0;
c = 3*10^8;
pt = [0;1500];
pr = [100;900];
pv = [xv;0];
t1 = sqrt(sum((pt-pv).^2))/c;
t2 = [sqrt(sum((pt-pr).^2))+sqrt(sum((pr-pv).^2))]/c;

fs = 200*f0;
t = 0:1/fs:3/f0;   %T of the combined signal = 1/f0, since both added
components have the same freq.
rv = cos(2*pi*f0*(t-t1))-cos(2*pi*f0*(t-t2));
plot(t,rv)
ylabel("r_v(t)");xlabel("t(in s)");
m = max(rv)                  % This is the max. value of the waveform.
```
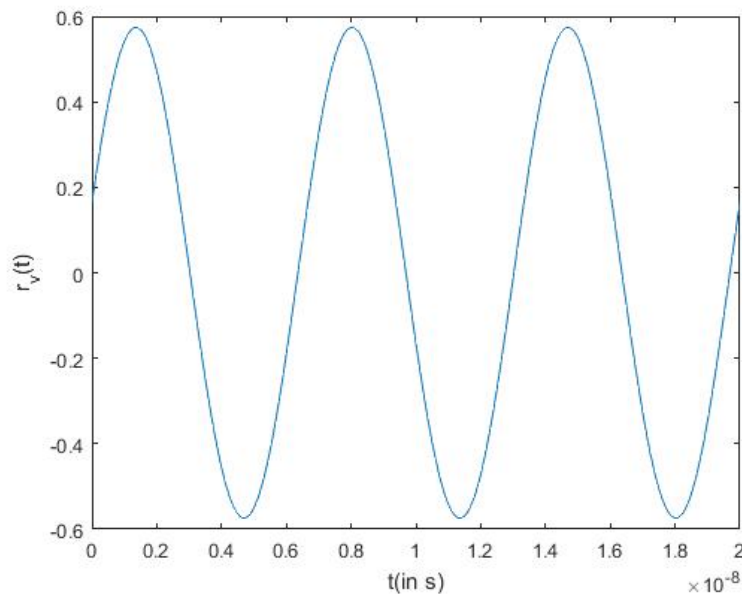

d) $r_v(t) = s(t - t_1) - s(t - t_2)$
$$= Re\{e^{j2\pi f_0(t-t_1)} - e^{j2\pi f_0(t-t_2)}\} = Re\{e^{j2\pi f_0 t}(e^{-j2\pi f_0 t_1} - e^{-j2\pi f_0 t_2})\}$$
$$= Re\{r_1 e^{j(2\pi f_0 t+\varphi)}\} \text{ , for some } r_1, \varphi.$$
$$\max r_v(t) = \max Re\{r_1 e^{j(2\pi f_0 t+\varphi)}\}$$
$$= \max r_1. \cos(2\pi f_0 t + \varphi)$$
$$= r_1. \max \cos(2\pi f_0 t + \varphi)$$
$$= r_1 \qquad (\text{as } \max \cos(2\pi f_0 t + \varphi) = 1)$$

$So, \max r_v(t) = r_1 = magnitude(r_{vc}(t))$, $r_{vc}(t) := r_1 e^{j(2\pi f_0 t+\varphi)}$

This shows how we can find the maximum value of the waveform just by taking the magnitude of the sum of two complex exponentials.

Using this derivation, the reported maximum value after running the code below is 0.5738. The graph is the same as shown above.

```
f0 = 150*10^6;
xv = 0;
c = 3*10^8;
pt = [0;1500];
pr = [100;900];
pv = [xv;0];
t1 = sqrt(sum((pt-pv).^2))/c;
t2 = [sqrt(sum((pt-pr).^2))+sqrt(sum((pr-pv).^2))]/c;

fs = 24*f0;
t = 0:1/fs:3/f0;
rv_complex = exp(2*j*pi*f0*(t-t1))-exp(2*j*pi*f0*(t-t2));
plot(t,real(rv_complex))
ylabel("r_v(t)");xlabel("t(in s)");
m = abs(rv_complex(1))      %All points in rv_complex have the same
magnitude, so we can take the absolute value of any point in rv_complex.
```

e) The complex amplitudes of the two waveforms are: $e^{-j2\pi f_0 t_1}$ & $e^{-j2\pi f_0 t_2}$.

Substituting the values of t_1 & t_2, the complex amplitudes become:

$$e^{-j2\pi f_0 \frac{\sqrt{d_t^2+x_v^2}}{c}} \& e^{-j2\pi f_0 \frac{\sqrt{(d_t-d_{yr})^2+d_{xr}^2}+\sqrt{(d_{xr}-x_v)^2+d_{yr}^2}}{c}}$$

To see why summing these complex amplitudes, and taking its magnitude will give the max value, consider the following proof.

From the previous part, we have,

$$e^{j2\pi f_0 t}\left(e^{-j2\pi f_0 t_1} - e^{-j2\pi f_0 t_2}\right) = r_1 e^{j(2\pi f_0 t+\varphi)}$$

Cancelling $e^{j2\pi f_0 t}$ from both sides, we get,

$$e^{-j2\pi f_0 t_1} - e^{-j2\pi f_0 t_2} = r_1 e^{j\varphi}$$

$$=> mag\,(e^{-j2\pi f_0 t_1} - e^{-j2\pi f_0 t_2}) = mag(r_1 e^{j\varphi}) = r_1 = \max\, r_v(t) \qquad\qquad --(1)$$

Solution #1: Using a for loop

```
f0 = 150*10^6;
xv = 0:0.01:300;
c = 3*10^8;
pt = [0;1500];
pr = [100;900];
max_val = zeros(length(xv),1);
pv = [xv(1);0];

for i=1:1:length(xv)
    pv = [xv(i);0];
    t1 = sqrt(sum((pt-pv).^2))/c;
    t2 = [sqrt(sum((pt-pr).^2))+sqrt(sum((pr-pv).^2))]/c;

    max_valc = exp(-2*j*pi*f0*t1)-exp(-2*j*pi*f0*t2);   %(1) shows how the
maximum value is calculated from this.
    max_val(i) = abs(max_valc);
end
plot(xv,max_val);
xlabel("Position of vehicle x (in m)");
ylabel("Signal strength");
```

Solution #2: Using vectorized code

```
f0 = 150*10^6;
xv = 0:0.01:300;
c = 3*10^8;
pt = repmat([0;1500],1,length(xv));
pr = repmat([100;900],1,length(xv));
pv = [xv;zeros(1,length(xv))];

t1 = sqrt(sum((pt-pv).^2))/c;
t2 = [sqrt(sum((pt-pr).^2))+sqrt(sum((pr-pv).^2))]/c;

max_valc = exp(-2*j*pi*f0*t1)-exp(-2*j*pi*f0*t2);    %(1) shows how the
maximum value is calculated from this.
max_val = abs(max_valc);
plot(xv,max_val);
xlabel("Position of vehicle x (in m)");
ylabel("Signal strength");
```

The vectorized code is much more efficient. I timed both the codes, and got these run times:

Vectorized code: 0.01627s

Code with for loop: 0.03051s

At first glance, the performance gains do not seem that dramatic (~2x). This is because of the costly repmat() operation at the beginning of the vectorized code. However, this cost gets amortized over the size of xv, making the vectorized code much more efficient for large arrays. To really see the benefits of vectorization, we need to time the core part of the code, which is the for loop in one of the codes, and the calculation of t1,t2,max_valc,max_val for the other. When just those parts are timed, I got the following run times:

Vectorized code: 0.485ms

Code with for loop: 12.07685ms

Here, we see that the vectorized code gives a *25x* speedup over the code with the for loop!

f) The code for the plot generation is given above. The derivation of how to get the peak value from the complex amplitude is also given above. The signal strength vs. xv plot is below.
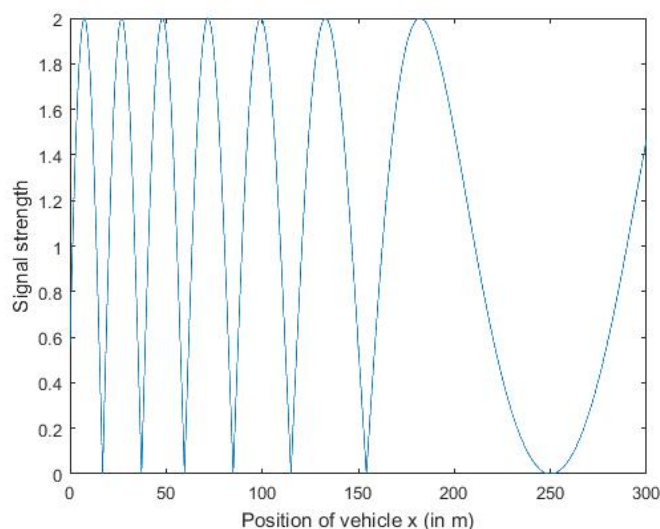
g) The largest value of the signal strength is 2 & the lowest value is 0. This makes sense as the largest value would be achieved when the two signals – directly transmitted & reflected, arrive exactly in phase with each other. In that case, the magnitudes of the two signals (magnitude of each signal = 1) add up to give a signal strength of 2.

The lowest value of the signal strength is achieved when the two signals arrive exactly (π rad) out of phase. Then, signal magnitudes get subtracted, giving a net signal strength of 0.

As we can see from the plot, complete signal cancellation occurs at a number of values of x_v. The values of x_v can be read off from the plot: 16.96m, 37.27m, 59.53m, 84.97m, 115.1m, 154.1m, 249.9m.

These values can also be calculated by equating the phase shifts.

For complete cancellation,

$$2\pi f_0. \left\{ \frac{\sqrt{(d_t-d_{yr})^2+d_{xr}^2}+\sqrt{(d_{xr}-x_v)^2+d_{yr}^2}}{c} - \frac{\sqrt{d_t^2+x_v^2}}{c} \right\} = 2n\pi, n \text{ is a whole number}$$

Substituting & simplifying, we get,

$$2\pi 150*10^\wedge 6. \left\{ \frac{\sqrt{(600^2+100^2}+\sqrt{(100-x_v)^2+900^2}}{c} - \frac{\sqrt{1500^2+x_v^2}}{c} \right\} = 2n\pi$$

$$2\pi 150*10^\wedge 8. \left\{ \frac{\sqrt{37}+\sqrt{(1-x_v/100)^2+81}}{c} - \frac{\sqrt{225+(x_v/100)^2}}{c} \right\} = 2n\pi$$

$$\sqrt{37} + \sqrt{(1-x_v/100)^2+81} - \sqrt{225+(x_v/100)^2} = n/50$$

Using Mathematica to solve this for different values of n, we get the exact zero locations: 16.99m, 37.23m, 59.58m, 84.9m, 114.9m, 154.2m, 249.8m.