In this week, I mainly focus on the algorithm of calculating the phase difference between signals. By calculating the phase difference between the signal waves then we can estimate the position of the signal arrival angle.

First, for phase difference calculation, we use the Pearson product-moment correlation coefficient which is is a "measure of the linear correlation (dependence) between two variables X and Y, giving a value between +1 and −1 inclusive, where 1 is total positive correlation, 0 is no correlation, and −1 is total negative correlation".(wiki)

The correlation coefficient ranges from −1 to 1. A value of 1 implies that a linear equation describes the relationship between X and Y perfectly, with all data points lying on a line for which Y increases as X increases. A value of −1 implies that all data points lie on a line for which Y decreases as X increases. A value of 0 implies that there is no linear correlation between the variables. More generally, note that (Xi − X)(Yi − Y) is positive if and only if Xi and Yi lie on the same side of their respective means. Thus the correlation coefficient is positive if Xi and Yi tend to be simultaneously greater than, or simultaneously less than, their respective means. The correlation coefficient is negative if Xi and Yi tend to lie on opposite sides of their respective means.

The formula is

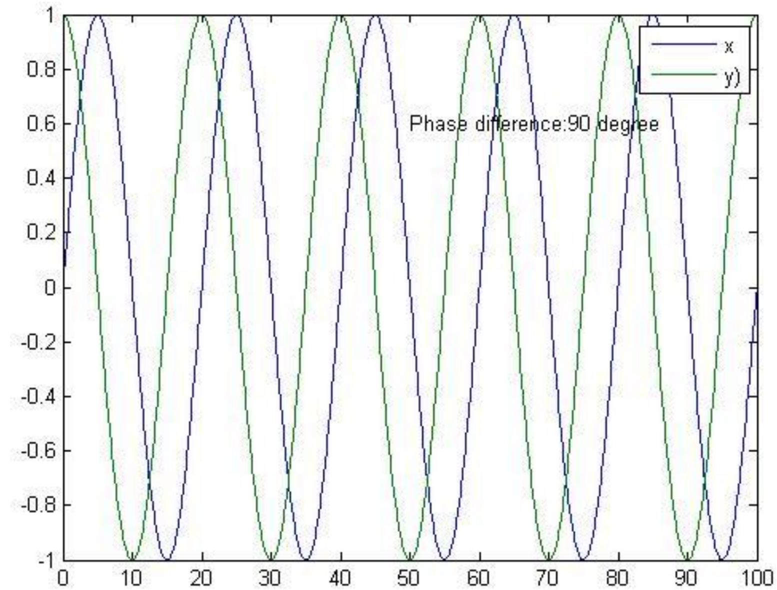
$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

However, if we look in the Geometric interpretation, it is more like the dot product. The theta angle is the phase difference we are looking for between two signals

$$\cos\theta = \frac{\mathrm{Re}(\mathbf{a}\cdot\mathbf{b})}{\|\mathbf{a}\|\,\|\mathbf{b}\|}.$$

In the matlab, I have done some simulation

First signal is x= sin(2πt) and the second signal is y=cos(2πt)



By applying the Pearson algorithm

We can get the phase difference theta easily, which is 90 degree.

The code is included:

```
num=100;
t=0:1:num;
l=linspace(0,100,1000);
x=sin(2*pi*50*l);
y=cos(2*pi*50*l);
num=1000;
Ix=sum(x.^2)/num;
Iy=sum(y.^2)/num;
Ixy=sum(x.*y)/num;
c=180*acos(2*Ixy/(4*Ix*Iy)^0.5)/pi;
plot(l,x,l,y);
legend('x','y');
text(50,0.6,strcat('Phase difference: ',strcat(num2str(c),' degree')));?
```