

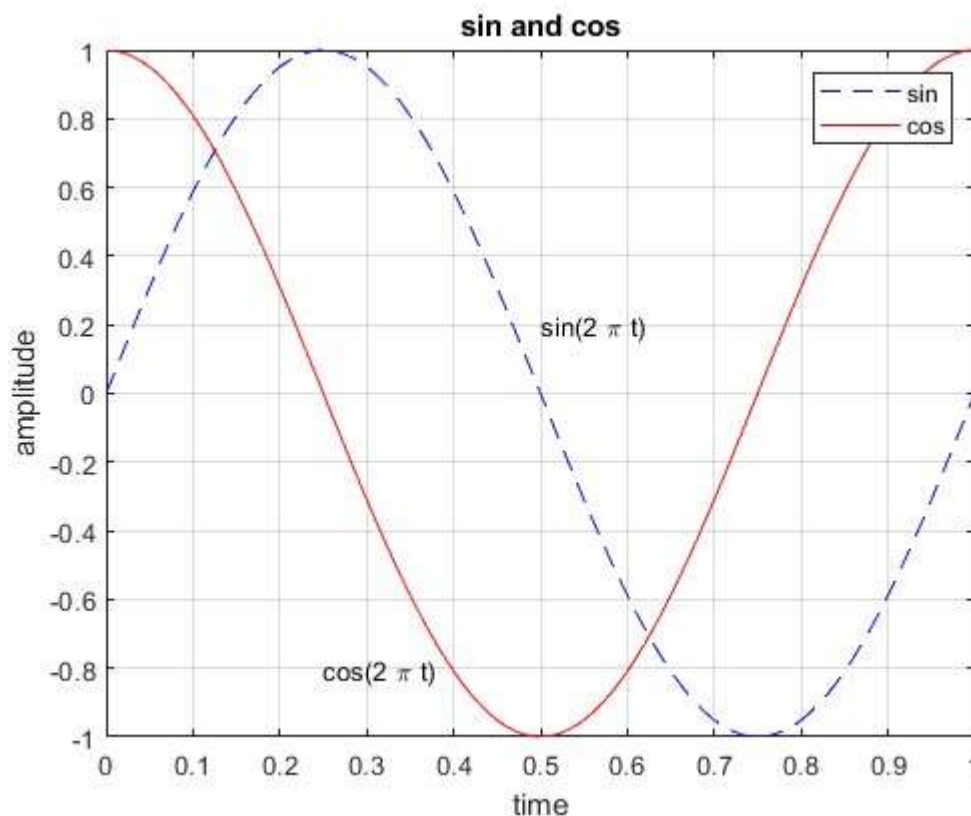
CA1

عباس خشدونی فراهانی

810101415

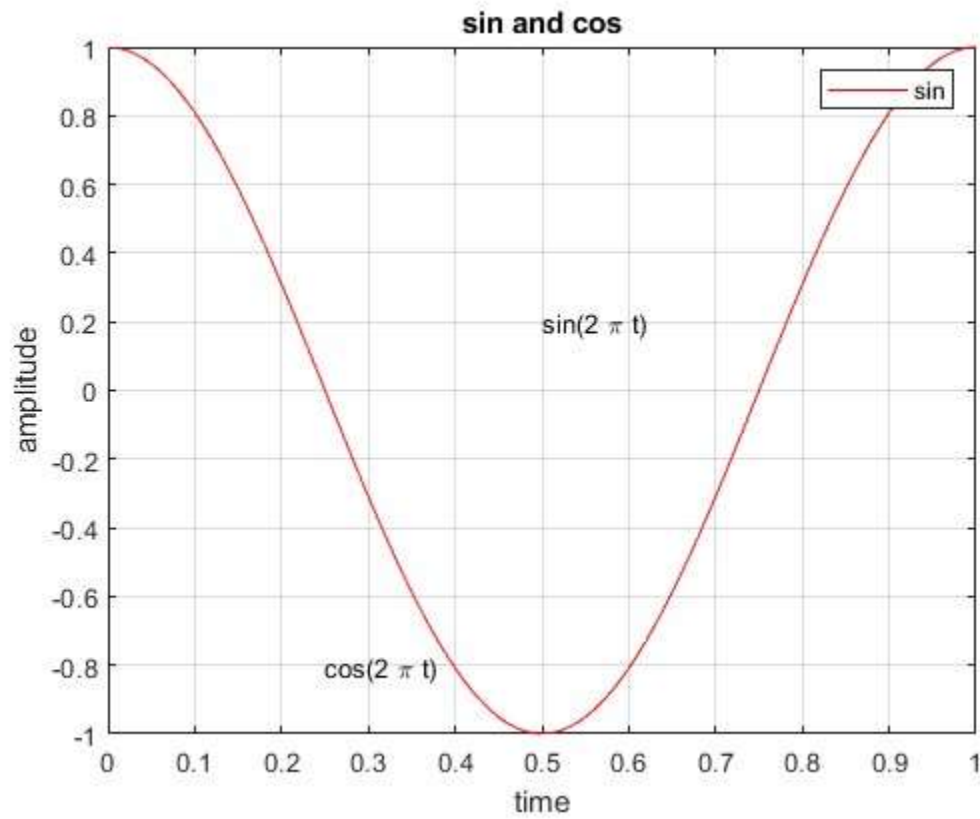
(1-1)

شکل نهایی تولید شده :

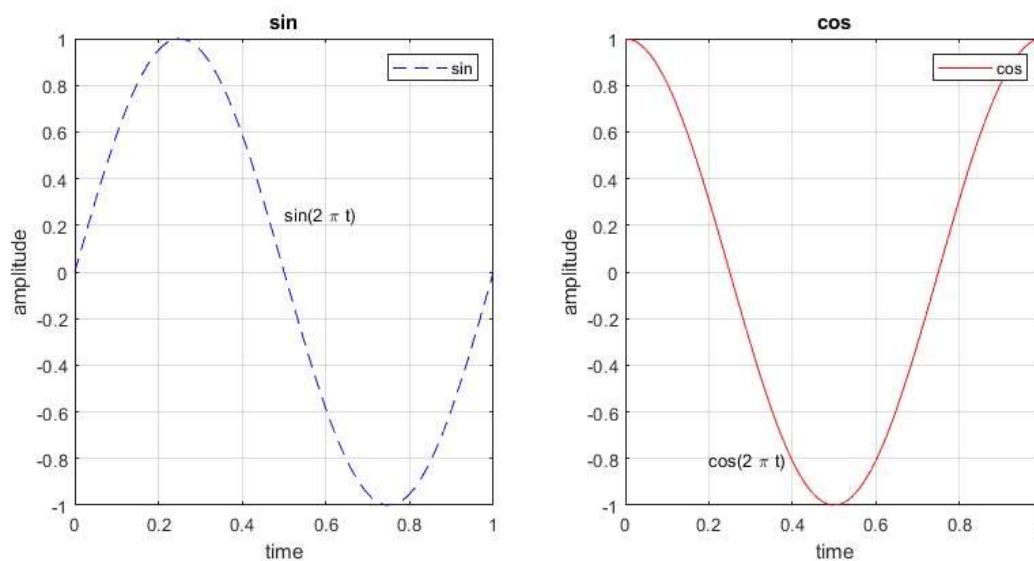


همان طور که می بینیم در صفحه دو تابع z_1 , z_2 ترسیم شده است و توسط دستور xlabel عبارت time و توسط دستور ylabel عبارت amplitude به شکل اضافه شده است. با استفاده از دستور grid on شکل واحد بندی شده است و با استفاده از دستور text به شکل حاصل در دو موقعیت x_0 , y_0 دو عبارت $\sin(2\pi t)$, $\cos(2\pi t)$ اضافه شده است و توسط عبارت legend مشخص شده است که دو خط کشیده شده ، هر کدام مربوط به کدام تابع هستند.

در صورت حذف دستور `hold on` اولین `plot(z1 = sin(2*pi*t))` رسم نمی شود. در واقع کاربرد دستور `hold on` در صورت حذف دستور `hold on` در همان صفحه ای که `plot` جدید رسم می شود است.



(1-2)

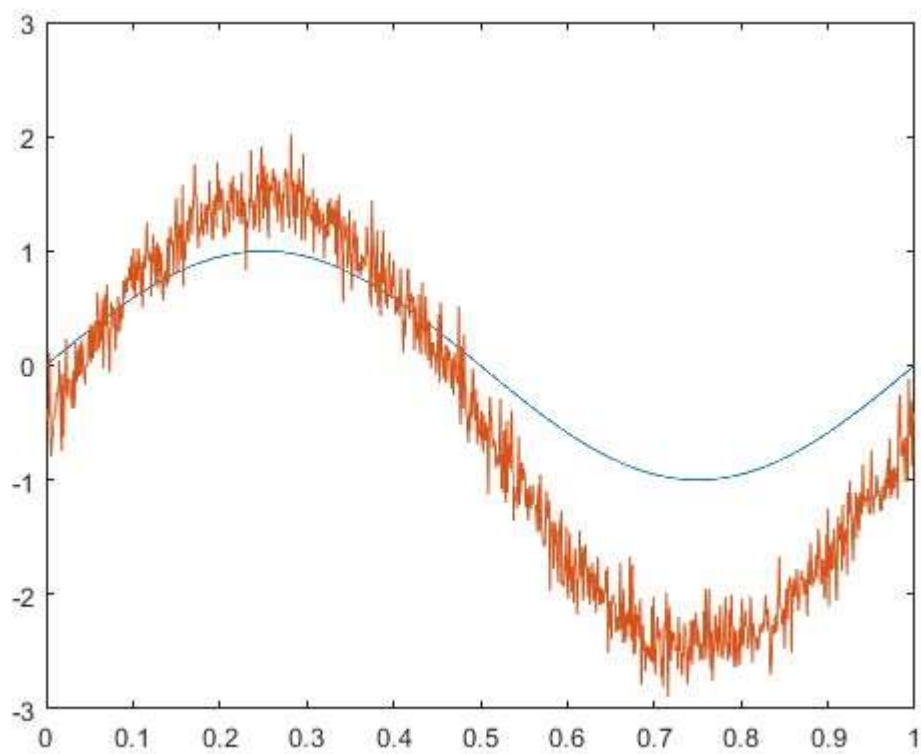


برای آنکه دو نمودار را در در شکل مجزا ولز در یک صفحه رسمی کنیم ، می توان از دستور subplot استفاده کرد. در این دستور می توان تعداد سطر و ستون و جای هر شکل را در این صفحه مشخص کرد ، به طوری که آرگومان اولی تعداد سطر و آرگومان دوم تعداد ستون و آرگومان سوم ، شماره شکل را مشخص میکند ، نکته مهم استفاده از این دستور ، نوشتن آن در کد قبل از دستور plot است .

بخش دوم:

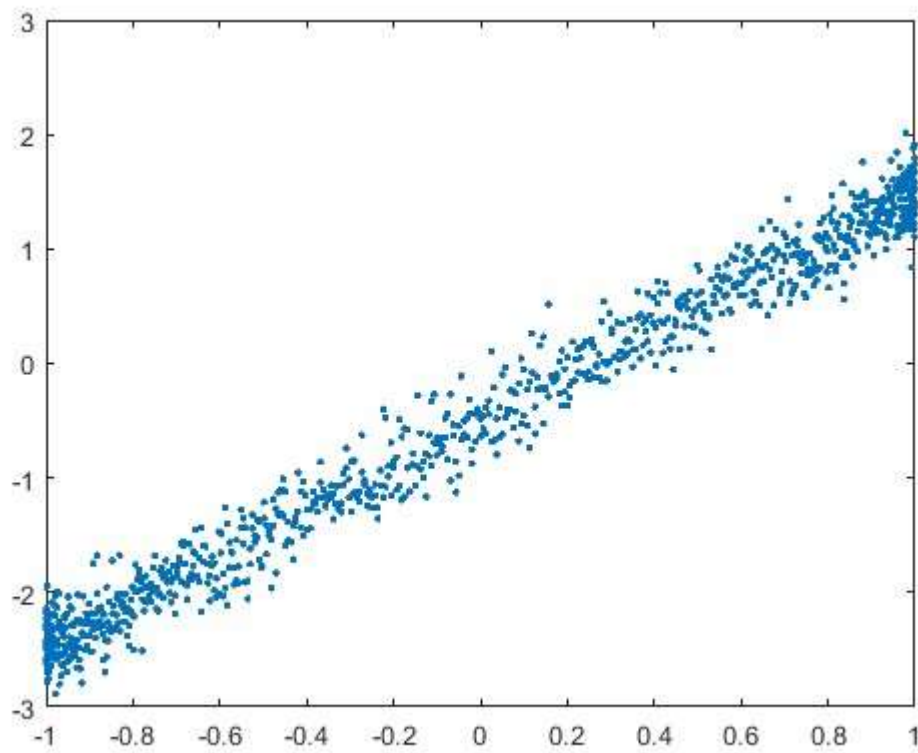
(2-1,2-2)

شکل های X , Y به طرز هم زمان در شکل زیر رسم شده است. برای رسم آن ها از دستور `plot` استفاده شده است. همان طور که از شکل مشخص است تابع Y دارای نویز است.



(2-3)

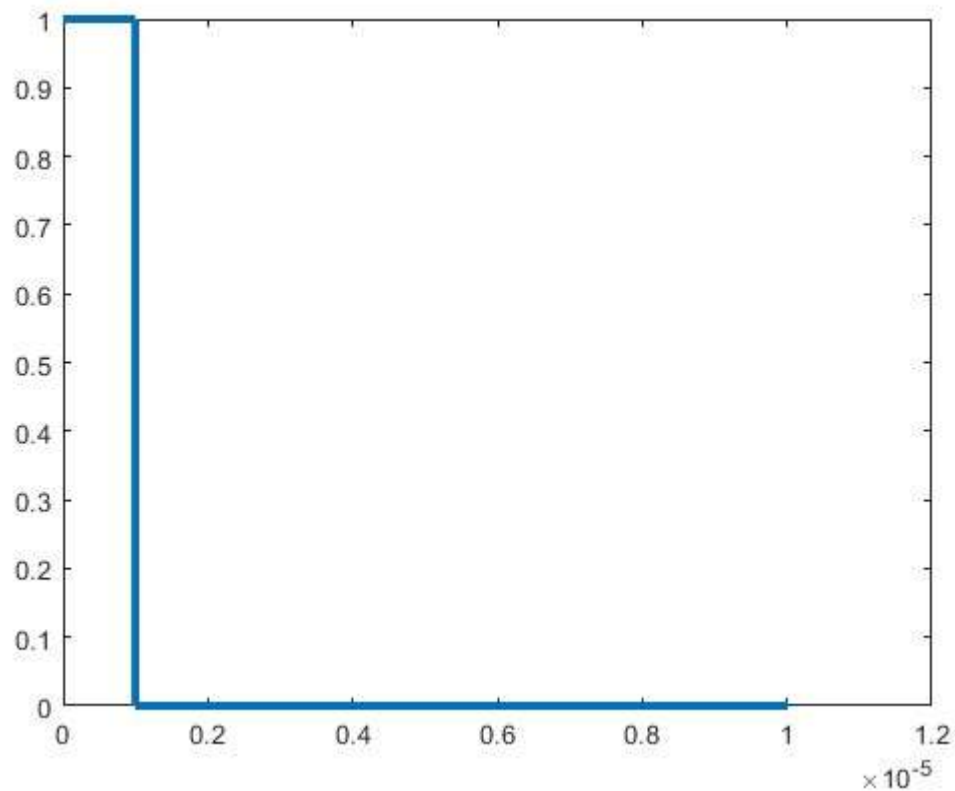
باتوجه به رابطه خطی موجود بین $x(t), y(t)$ شیب خط حاصل اندازه α و عرض از مبدا آن β را به ما می دهد.



(2-4)

بخش سوم:

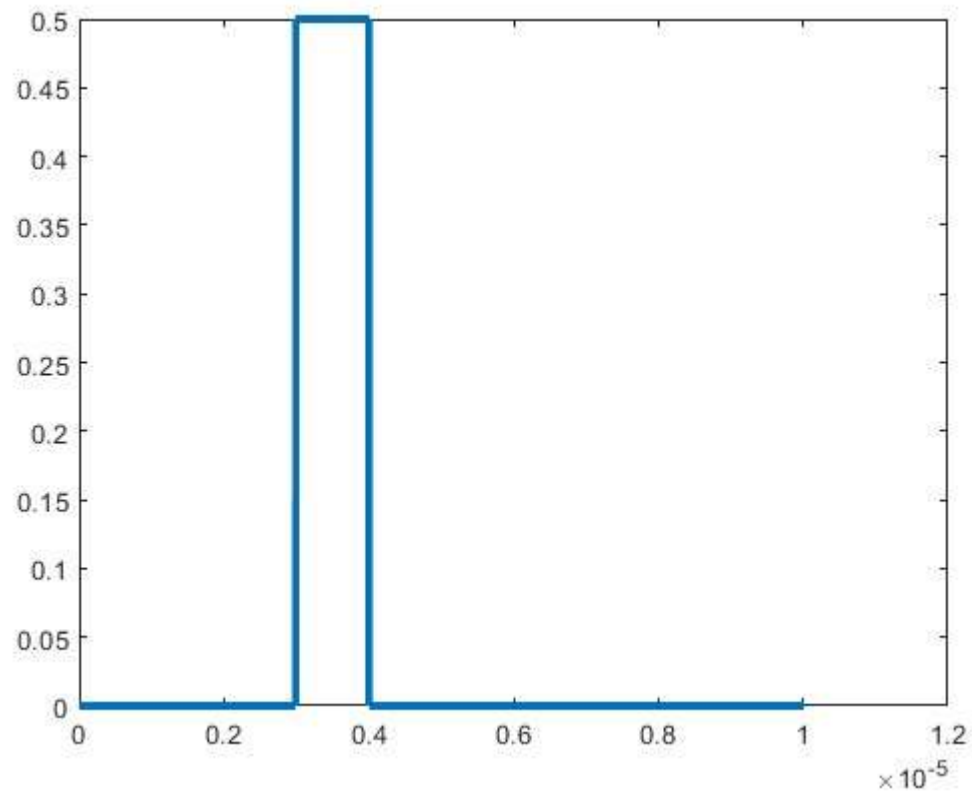
(3-1)



با توجه به صورت سورت سوال سه مقدار T , t_s , τ تعریف می کنیم. سپس یک بازه زمانی به اندازه T با فاصله زمانی t_s ایجاد می کنیم و یک سیگنال به طول اعضای این متغیر تعریف می کنیم که در ابتدا مقدار آن صفر است ، سپس با استفاده از یک حلقه for مقدار تعداد اعضای τ/t_s اولیه آن را به یک تغییر می دهیم تا برای ما یک پالس ایجاد شود. به این سیگنال اولیه ما می باشد .

(3-2)

فاصله داده شده برابر 450 متر است ، با توجه به رابطه $t_s = \frac{R \cdot 2}{C}$ و با در نظر گرفتن $C = 3e8$ می توان مقدار t_s را بدست آورد که برابر $3e-6$ می باشد. حال برای سیگنال دریافتی باید محدوده پالس را مشخص کنیم که برای این منظور نسبت t_s بدست آمده را به T حساب می کنیم و در طول ماتریس t ضرب می کنیم و با استفاده از تابع round نزدیک ترین عدد صحیح به آن را بدست می آوریم که این عدد نشان دهنده شماره درایه شروع سیگنال پاسخ می باشد و از آنجا که طول پالس تغییر نکرده است آن را تغییر نمی دهیم .



(3-3)

```
ts = 1e-9;
T = 1e-5;
tau = 1e-6;

t = 0:ts:T;
signal = zeros(1,length(t));

for a = 1:(tau/ts)
    signal(1,a) = 1;
end

ts = 1e-9;
T = 1e-5;
tau = 1e-6;
R = 450;
speed = 3e8;

t = 0:ts:T;
ansSignal = zeros(1,length(t));
```



```

td = (R*2)/speed;
td = round(td, 9);

t1=(td/T)*length(t);
t1 = round(t1,0);
for a = t1:(tau/ts)+t1
    ansSignal(1,a) = 0.5;
end

sumVector = zeros(1,length(t));
invrAnsSignal = ansSignal';

for s = 1:length(signal)
    y = circshift(signal,s);
    sumd = y*invrAnsSignal;
    sumVector(1,s) = sumd;
end

[Mr, Mc] = max(sumVector);

td1 = (Mc/length(signal))*T;
R1 = td1*speed/2;

```

همان طور که میبینیم برای تشخیص زمان سیگنالی برگشتی با استفاده از کورلیشن زمان را محاسبه کرده ایم. برای این منظور یکی از سیگنال ها را معکوس میکنیم و در سیگنال دیگر ضرب میکنیم ، و این کار با شیفت دادن یکی از سیگنال ها به اندازه یک واحد در هر مرحله تکرار میکنیم و در نهایت بزرگن ترین عدد بدست آمده نشان دهنده زمانی است که هر دو سیگنال بیشترین تطابق را داشته اند. و در نهایت با استفاده از رابطه $t_d * C = R * 2$ میتوان فاصله را بدست آورد. که فاصله نهایی 449.8050 می آید.

(3-4)

همان طور که مشخص است با افزایش نویز دقت این روش کم میشود و خطای فاصله اندازه گیری شده افزایش می یابد و این روش با دقت خوبی تا وقتی که ضریب نویز را 3 برار کنیم فاصله را تشخیص می دهد و بعد از این مقدار دیگر تضمینی نیست و خطا افزایش می یابد. در محاسبات انجام شده در هر ضریب ثابتی از خطا 100 بار انجام شده است و حاصل آنها را میانگین گرفته ایم.

