به نام خدا

سید حسن محقق بهشتی (۸۱۰۱۹۲۴۶۲)

دکتر صدف صالح کلیبر - سیگنال ها و سیستم ها - تمرین کامپیوتری شماره ی ۱

خلاصه

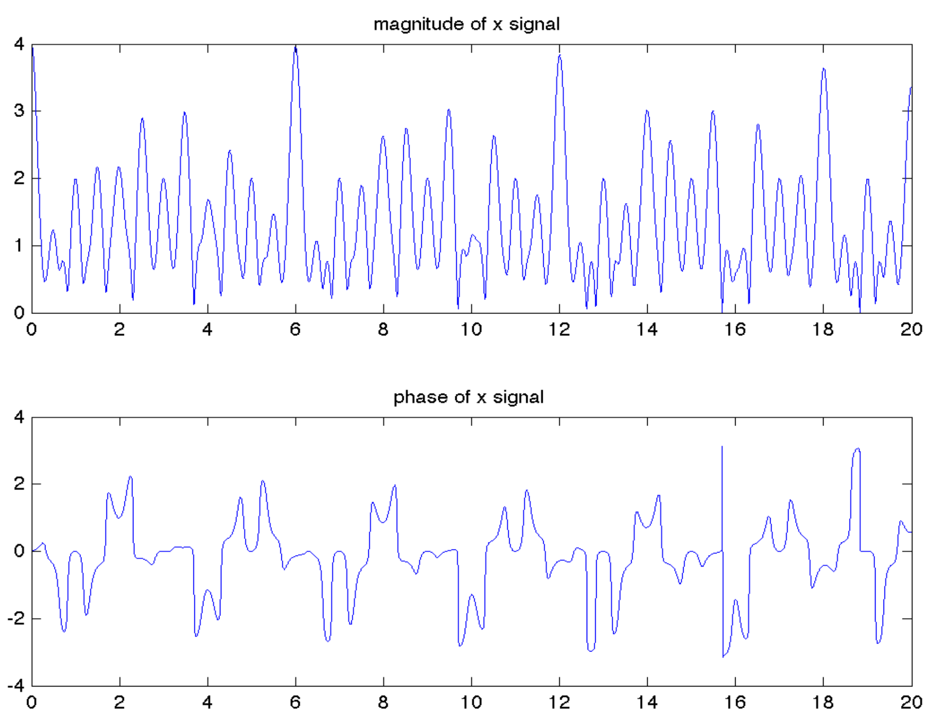
در تمارین این پروژه قصد داریم ابتدا کمی با بعضی توابع و زبان Matlab آشنا شده سپس با نوت های موسیقی و سیگنال های حاصل آن کار کنیم و فیلترهایی را بسازیم.

بخش اول

۱ - *رسم سیگنال* زیر

$$x_a(t) = 1 + \cos(4\pi t) + e^{jt}(\cos(2\pi t) + \cos(\pi t))$$ (فرمول شماره ی ۱)

نتیجه ی کد زده شده برای فرمول ۱ در شکل شماره ی ۱ بوده است .



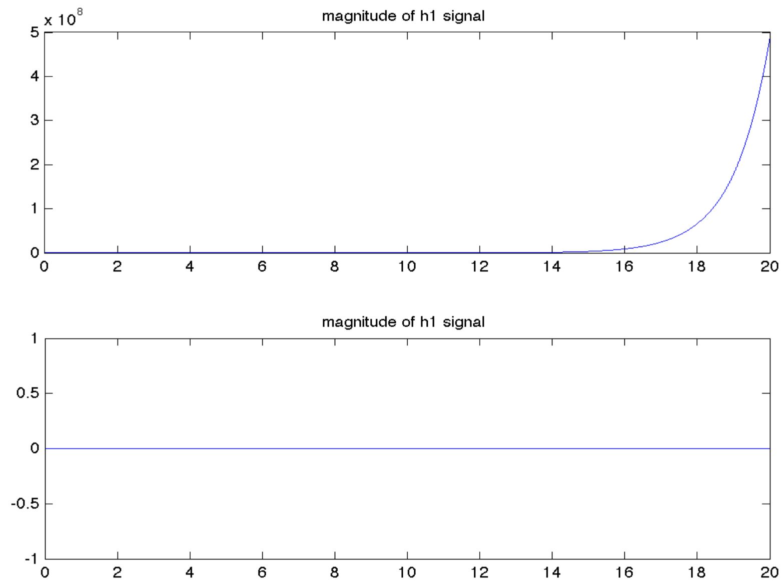
(شکل شماره ی ۱)

۲ - رسم سیگنال های زیر

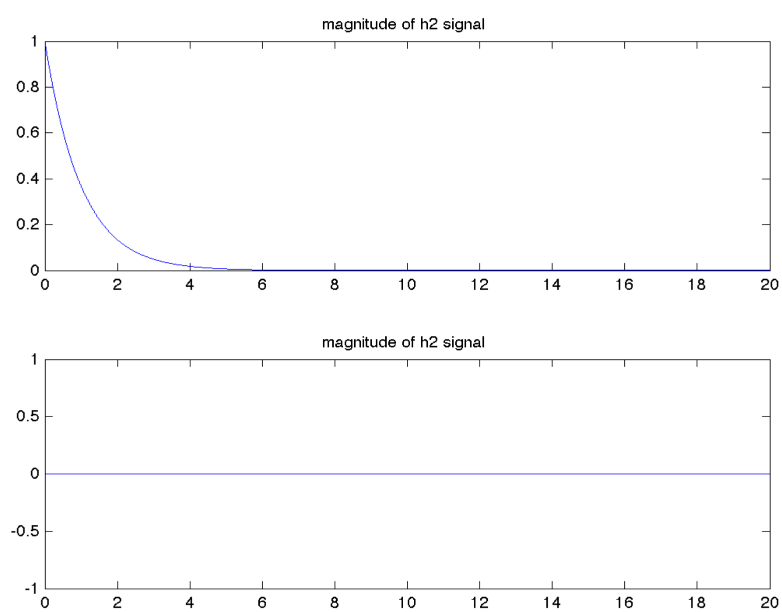
$h_1(t) = e^{t}$ (فرمول شماره ی ۲)

$h_2(t) = e^{-t}$ (فرمول شماره ی ۳)

به ترتیب نتیجه ی h_1 و h_2 به ترتیب در شکل های ۲ و ۳ آمده است.



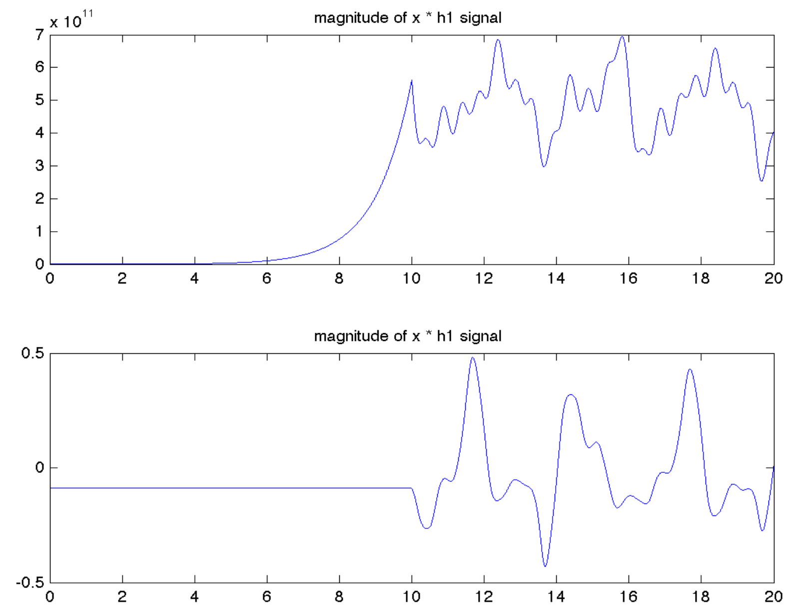
(شکل شماره ی ۲)



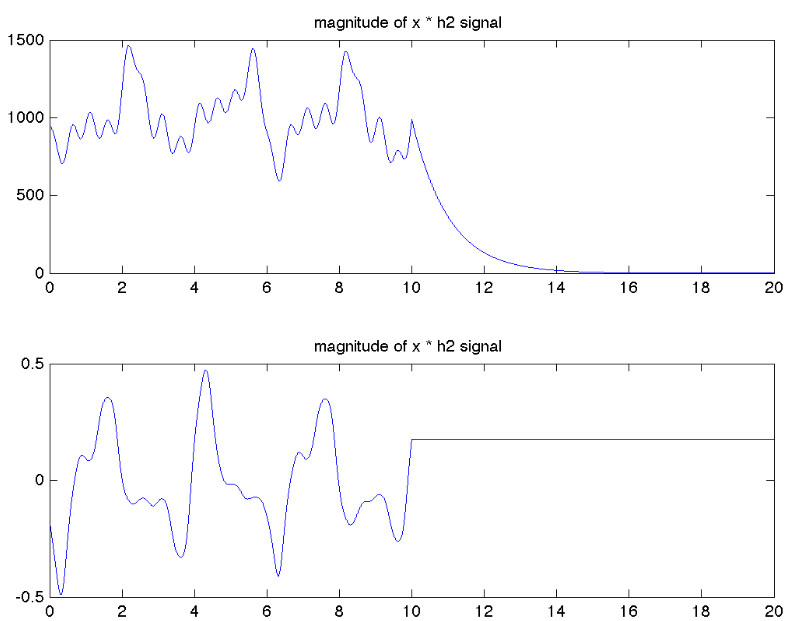
(شکل شماره ی ۳)

۳- کشیدن سیستم هایی با پاسخ ضربه ی h_1 و h_2 و ورودی x_a

برای کشیدن از تابع conv استفاده شده و در حالت 'Same' . در شکل ۴ و ۵ به ترتیب سیستم های حاصل از پاسخ ضربه ی h_1 و h_2 و ورودی x_a آمده است.



(شکل شماره ی ۴)



(شکل شماره ی ۵)

۴- مقایسه دو سیستم h_1 و h_2 با هم

برای مقایسه پاسخ فرکانسی این دو سیستم را با هم مقایسه می کنیم . برای اینکار از فرمول تبدیل فوریه استفاده می کنیم و این نکته هم در نظر میگیریم که محدوده ی t انتخاب شده ۰ تا ۲۰ است .

$$H_1(jw) = \int_{-\infty}^{+\infty} e^{t}\, e^{-jwt}\, dt = \int_{-\infty}^{+\infty} e^{(1-jw)t}\, dt = \frac{1}{1-jw}\, e^{(1-jw)t} \Big|_{-\infty}^{+\infty}$$ (فرمول شماره ی۴)

به طور مشابه برای $H_2(jw)$ هم داریم :

$$H_2(jw) = \int_{-\infty}^{+\infty} e^{-t}\, e^{-jwt} = \int_{-\infty}^{+\infty} e^{(-1-jw)t} = \frac{1}{-1-jw}\, e^{(-1-jw)t} \Big|_{-\infty}^{+\infty} = \frac{-1}{1+jw}\, e^{(-1-jw)t} \Big|_{-\infty}^{+\infty}$$ (فرمول شماره ی۵)

برای اینکه t ما محدود به ۰ تا ۲۰ است پس به جای بی نهایت های انتگرال از این حدود استفاده می کنیم و توابع h_1 و h_2 را در فواصل دیگر صفر فرض میکنیم. با این فرض داریم:

$$H_1(jw) = \frac{1}{1-jw}\, e^{(1-jw)t} \Big|_0^{20} = \frac{1}{1-jw}\, e^{(1-jw)20} - \frac{1}{1-jw}$$ (فرمول شماره ی۶)

$$H_2(jw) = \frac{-1}{1+jw}\, e^{(-1-jw)t} \Big|_0^{20} = \frac{-1}{1+jw}\, e^{(-1-jw)20} + \frac{1}{1+jw}$$ (فرمول شماره ی۷)

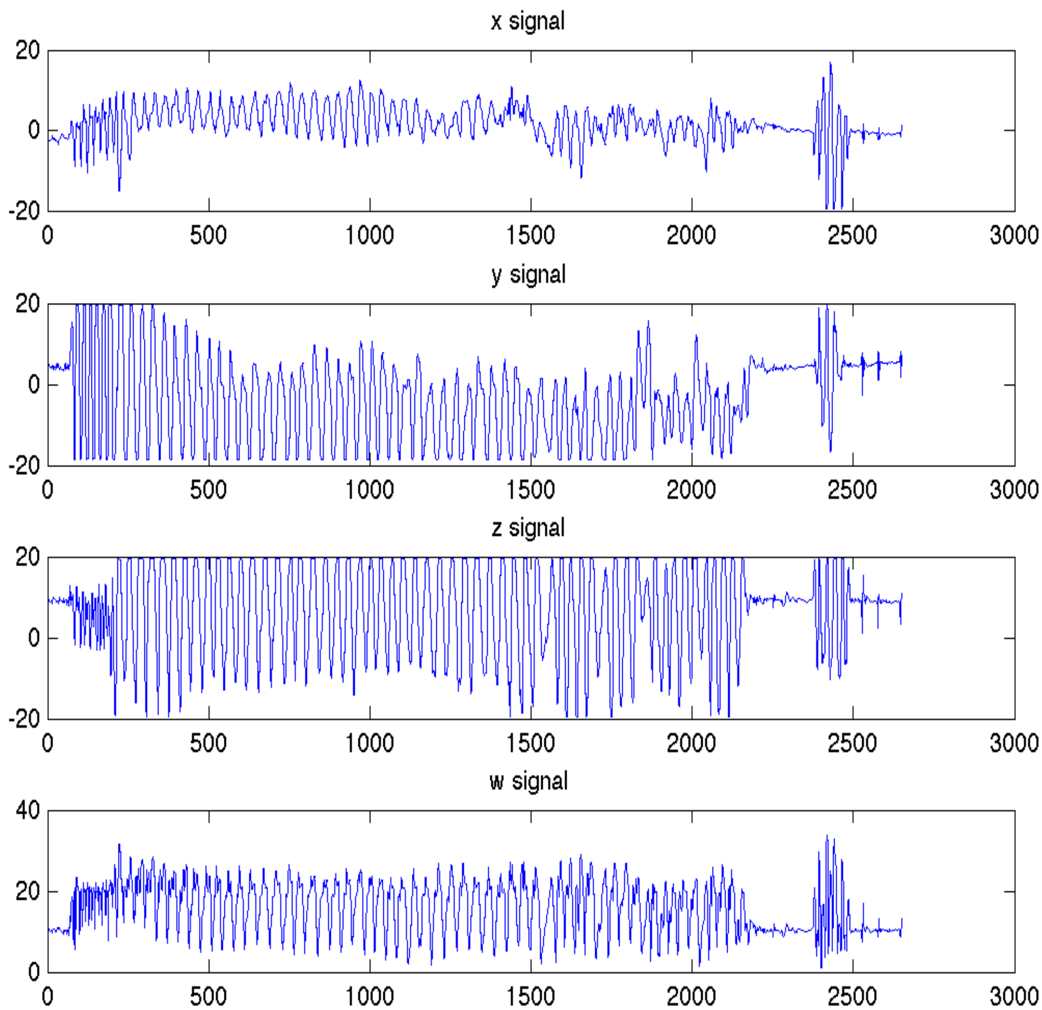
اگر ۰ و بی نهایت را به جای w در فرمول جای گذاری کنیم می بینیم که h_1 به نوعی فیلتر بال گذر است(غیر ایده آل) و h_2 به فیلتر پایین گذر است . پس می توان به این شکل تفاوت این دو جواب دو سیستم را با این استدلال توجیه کرد.

*فایل مربوط به بخش اول به اسم **part1.m** در فولدر پروژه پیوست شده*

بخش دوم

۵- لود کردن داده ها از فایل

فایل accel_data.txt در متغیر c به صورت یک آرایه ی 4 × n ذخیره شد و هر ستون آن در متغییرهای x y z و w ذخیره گردید و توسط plot رسم گردید که در شکل شماره ی ۶ می توانید مشاهده کنید .



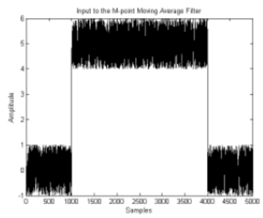
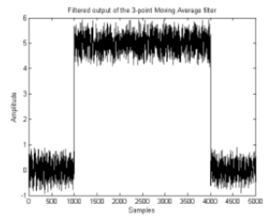
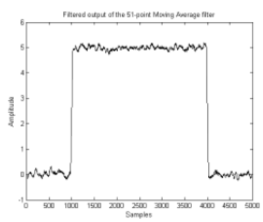
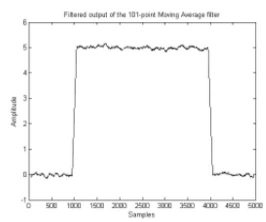
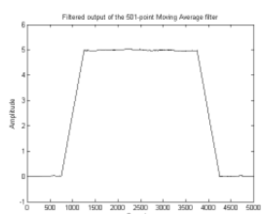
(شکل شماره ی۶)

همانطور که قابل مشاهده است این چهار سیگنال بسیار نویزی هستند.

۶- میانگین_متحرک *(moving average)*

فیلتر میانگین متحرک فیلتری است که برای smooth کردن signal های نویزی به کار برده می شود و در اصل یک فیلتر پایین گذر به صورت پاسخ های ضربه محدود است (FIR) که به این صورت عمل می کند که M دیتای آخر را میانگین می گیرد و به در خروجی قرار می دهد ، پس هرچه M بزرگتری داشته باشیم Signal خروجی Smooth تری خواهیم داشت .

همانطور که در شکل آمده قابل مشاهده است که با تعداد نقاط میانگین گیری بیشتر خواهیم signal خروجی کم نویز تری داریم .[1]

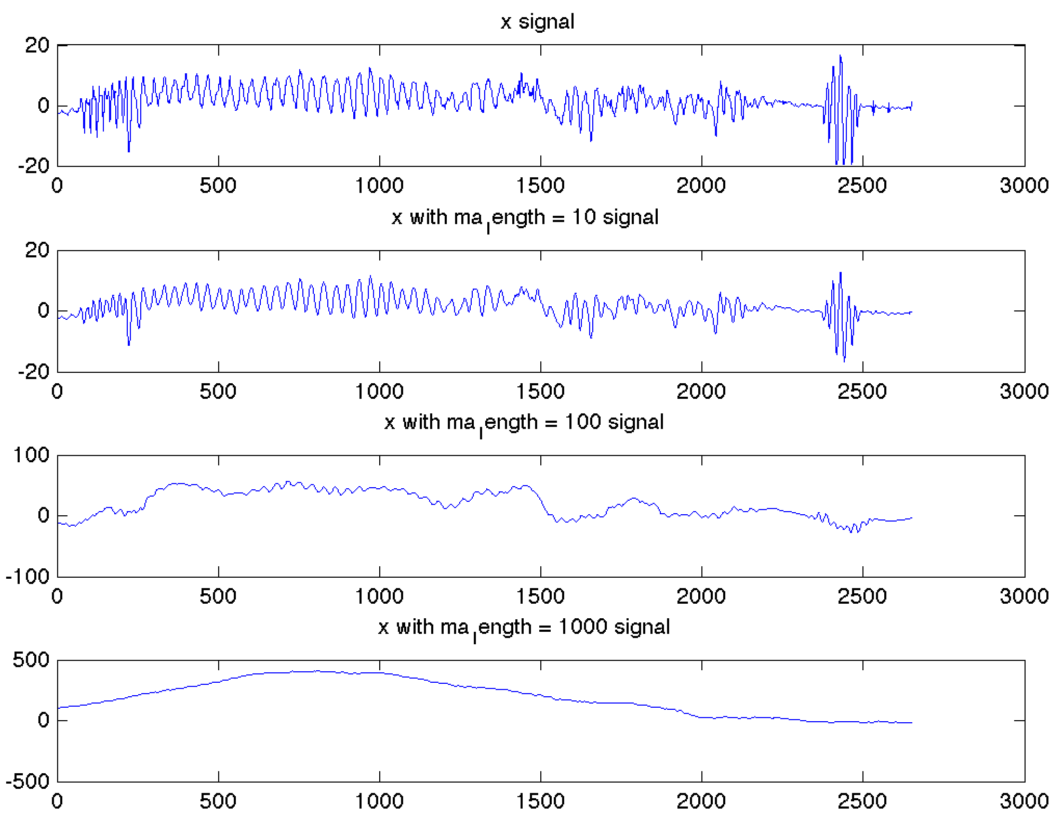
<table>
<tr><th>Input to MA filter</th><th>3-point MA filter output</th></tr>
<tr><td>
Input to Moving average filter</td><td>
Response of 3 point Moving average filter</td></tr>
<tr><th>51-point MA filter output</th><th>101-point MA filter output</th></tr>
<tr><td>
Response of 51-point Moving average filter</td><td>
Response of 101-point Moving average filter</td></tr>
<tr><th colspan="2">501-point MA filter output</th></tr>
<tr><td colspan="2">
Response of 501 point Moving average filter</td></tr>
</table>

(شکل شماره ی۷)

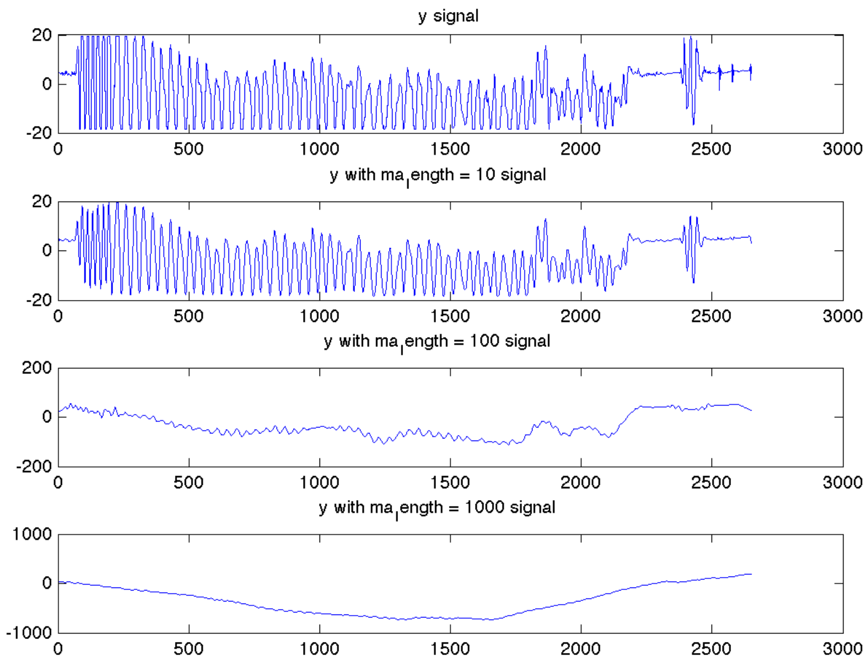
[1] http://www.gaussianwaves.com/2010/11/moving-average-filter-ma-filter-2/

۷- *میانگین_متحرک (moving average) و سیگنال های x,y,z*

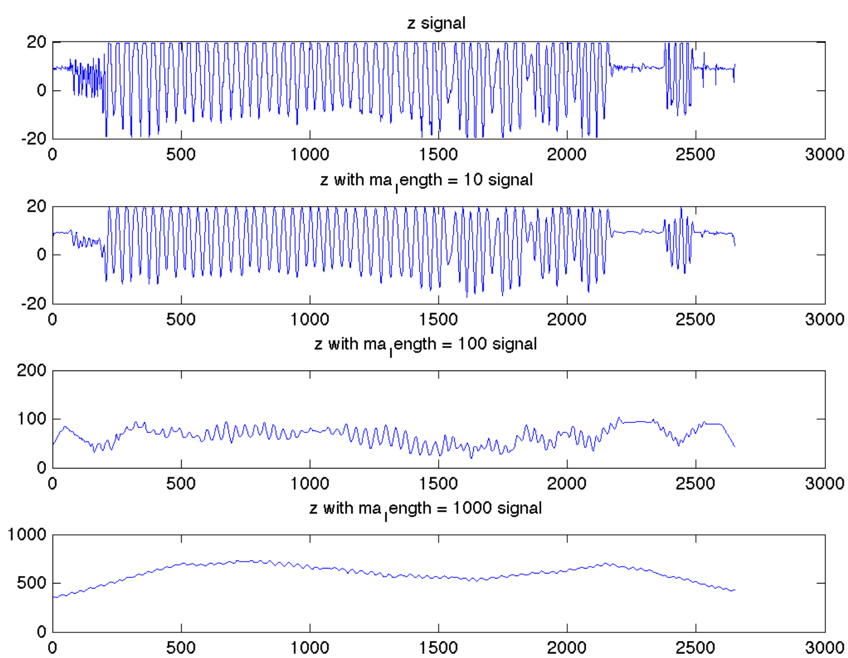
با ma_length های متفاوت فیلتر میانگین متحرک را بر روی سیگنال ها اعمال می کنیم و نتیجه به صورت زیر در می آید. اشکال به این صورت است که سیگنال بدون نویز ابتدا و سپس لول های مختلف حاصل از فیلترها با MA-Point های متفاوت در شکل های زیر آمده است . به ترتیب شکل های شماره ی ۸ ، ۹ و ۱۰ مربوط به حاصل کانولوشن سیگنال های x , y و z هستند .



(شکل شماره ی ۸)



(شکل شماره ی ۹)



(شکل شماره ی ۱۰)

۸- مقایسه میانگین_متحرک (moving average) با MA-Point های متفاوت و سیگنال های x,y,z

همانطور که قابل مشاهده است با افزایش این عدد ما نویزها را بیشتر کاهش می دهیم ، ولی اگر خیلی این عدد هم بزرگ شود با این مشکل روبرو خواهیم بود که دقت نمونه ها را کاهش می دهیم و با بزرگ کردن این عدد و میل دادن آن به بی بینهایت به میانگین تابع نزدیک می شویم

*فایل مربوط به بخش دوم به اسم **part2.m** در فولدر پروژه پیوست شده.*

بخش سوم

در این بخش قصد داریم با کمک MATLAB آهنگ بسازیم.

۹ - خلاصه ای از توضیحات گفته شده در ابتدای بخش سوم

آهنگ از note ها تشکیل شده . برای تولید آهنگ باید note های مختلف در طول زمان پخش می شوند. برای تولید این نوت ها ما موج سینوسی آن را می توانیم شبیه سازی کنیم که برای شبیه سازی بهتر ما نه تنها آن موج سینوسی بلکه موج های سینوسی با فرکانس هایی با ضریب صحیح فرکانس note ولی با دامنه ی کمتر را در نظر می گیریم. برای شبیه سازی میرایی note ها هم آن تابع را در تابع میرایی ضرب می کنیم. به همین ترتیب برای تابع سینوسی داریم :

$$x_i(tt) = na(i) * sin(2 * \pi * i * nf(i) * tt)/i$$ (فرمول شماره ی۸)

و این کار را به ازای i ها در محدوده ی ۱ تا ۱۰ انجام می دهیم . به ازای تمام note های نوشته شده هم این کار را تکرار می کنیم.

فرمول tt هم از این جا در آمده که ، فرکانس کشیدن که FS است پس طول واحد زمانی ما $1/FS$ خواهد بود و بعضی note ها از استاندارد طول TD که برای note ها است کمتر است پس داریم :

$tt = 0 : 1/FS : TD/nd(i)$ (فرمول شماره ی۹)

این که چرا طول tt به اندازه ی $FS * (TD/nd(i))$ است ، اگر هر واحد $1/FS$ باشد و طول بازه ۰ تا $TD/nd(i)$ پس این بازه از $FS * (TD/nd(i))$ قسمت تشکیل شده و به همین دلیل طول بردار آن $FS * (TD/nd(i))$ است.

(۱۰ ، ۱۱ ، ۱۲ ، ۱۳ ،۱۵ ،۱۴)- تولید آهنگ

در این بخش طبق دستورالعمل داده شده عمل شد و بخش امتیازی هم انجام گردید ، فایل های موسیقیه ساخته شده هم ضمیمه شد.

*فایل مربوط به بخش سوم به اسم **part3.m** و فایل موسیقی **music.wav** در فولدر پروژه پیوست شده.*

پایان