

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



آزمایشگاه پردازش بی درنگ سیگنال های دیجیتال

تمرین شماره ۳

نام و نام خانوادگی : سهند خوشدل

شماره دانشجویی : ۸۱۰۱۹۶۶۰۷

پاییز ۹۹

سوال اول: پیاده سازی سیستم تولید echo در MATLAB و تحلیل ناپایداری

قسمت اول (

بدست آوردن تابع تبدیل سیستم و مشخص کردن ضرایب فیلتر :

در صورتی که تابع تبدیل مربوط به سیستم نشان داده شده در دستور کار را بدست آوریم، به عبارت زیر می رسیم:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + (1-g)z^{-N}}{1 - (gz^{-N})}$$

سپس با توجه به تابع تبدیل به دست آمده یک فیلتر ساده از مرتبه N طراحی می کنیم، که ضرایب آن به صورت زیر

باشد :

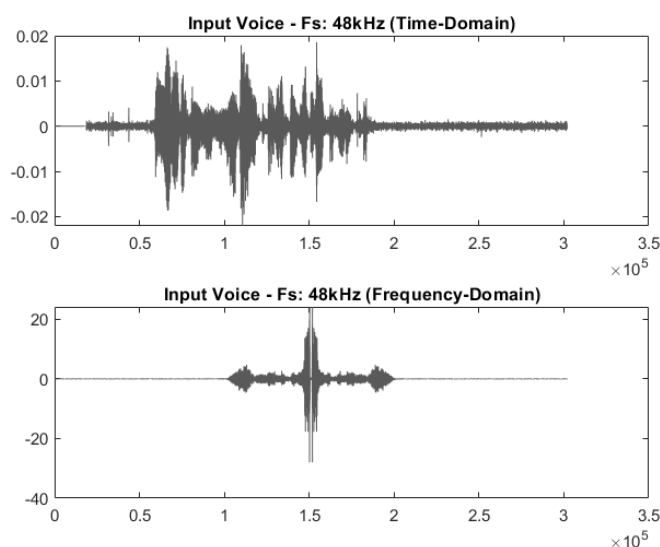
$$\text{Num} = [1, 0, 0, \dots, 0, (1-g)]$$

$$\text{Den} = [1, 0, 0, \dots, 0, (-g)]$$

قسمت دوم (

انتخاب یک صدای نمونه و نمونه برداری از صدای ورودی :

صدای خود را به عنوان یک صدای نمونه برای تست سیستم اکو ضبط کردم و این فایل صوتی را در فولدر question1 با نام Echo_System_Test_Input ذخیره کردم. سپس به کمک دستور audioread فایل صوتی را خوانده و با فرکانس گفته شده یعنی ۱۶ کیلو هرتز از آن نمونه برداری کردم و سیگنال را در حوزه زمان و فرکانس رسم کردم. موقع گوش دادن صدا متوجه شدم صدا به طرز قابل توجهی بم و کشیده شده و درست نمونه برداری نشده است که این نشان می دهد. به همین دلیل نمونه برداری را با فرکانس ۴۸ کیلو هرتز مجددا انجام دادم و سایر آزمایش را بر پایه این نرخ نمونه برداری ادامه دادم.



شکل ۱-۱: شکل سیگنال ورودی در حوزه زمان و فرکانس

قسمت سوم (

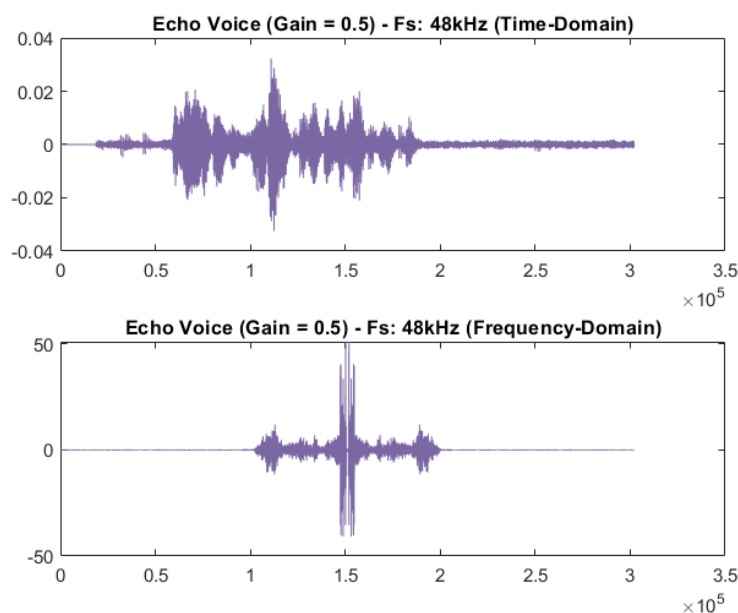
طراحی تابع `apply_echo` و دادن ورودی به فیلتر `echo`:

با توجه به اینکه در وطل این آزمایش به چنرین بار اعمال فیلتر بر روی ورودی احتیاج داریم. اعمال فیلتر را به صورت تابع جداگانه در MATLAB پیاده سازی کردم. و هر دفعه برای اعمال فیلتر به ورودی با `gain` دلخواه سرفا تابع را فراخوانی کرده و `gain` مد نظر را به عنوان آرگومان ورودی به آن داده ام.

ساختار این تابع بسیار ساده ست یک ورودی میگیرد و فیلتر را که در قسمت اول پیاده سازی کرده ایم در بدنه تابع قرار دادهام تا به ورودی اعمال کند و خروجی را برگرداند.

برای `gain=0.5` این کار را انجام دادم و صدای تولیدی را گوش کردم و کمی اکو (به صورت لرزش تشخیص داده شد) دستور `play` را تمامی مراحل `comment` کرده ام تا خروجی کد شلوغ نشود و در عوض خروجی تمام مراحل را به کمک دستور `audiowrite` در فایل مربوط به سوال ۱ ذخیره کرده ام.

خروجی سیستم اکو برای `gain = 0.5` در تصویر زیر در حوره زمان و فرکانس مشخص شده است :



شکل ۲-۱: خروجی سیستم اکو با `gain=0.5`

در فیلتری که سیستم اکو معادل آن عمل میکند از طرفی یکبار تمام ضرایب از ورودی در خروجی به صورت مستقیم ظاهر می شود (مسیر `feed forward` از بالا) و از طرفی هم در به دلیل وجود فیدبک مثبت در سیستم یک ورودی پس از تاخیر یافتن (z^{-N}) با `gain < 1` نسبت مشخصی تضعیف می شود و دوباره در خروجی ظاهر می شود تا به مرور زمان به قدری تضعیف شود تا صدای آن برای گوش انسان قابل ملاحظه نباشد

از آنجا که تنها محور اندازه در پاسخ فرکانسی تا 50 اسکیل شده و شکل تقریباً ثابت مانده در می یابیم که سیستم اکو اعوجاج اندازه ندارد و تمام ی فرکانس ها در آن تقریباً به یک اندازه تقویت می شوند هر چند نمونه هایی که زودتر وارد

سیستم شده اند یعنی در محور زمان عتب تر بوده اند گردش های بیشتری در سیستم خواهند داشت و اکوی آن ها قابل ملاحظه تر خواهد بود . (با توجه به اینکه زمان ورودی و خروجی محدود است)

قسمت چهارم)

مقایسه اثر gain های بالاتر از 0.5 بر خروجی سیستم اکو و تحلیل ناپایداری آن:

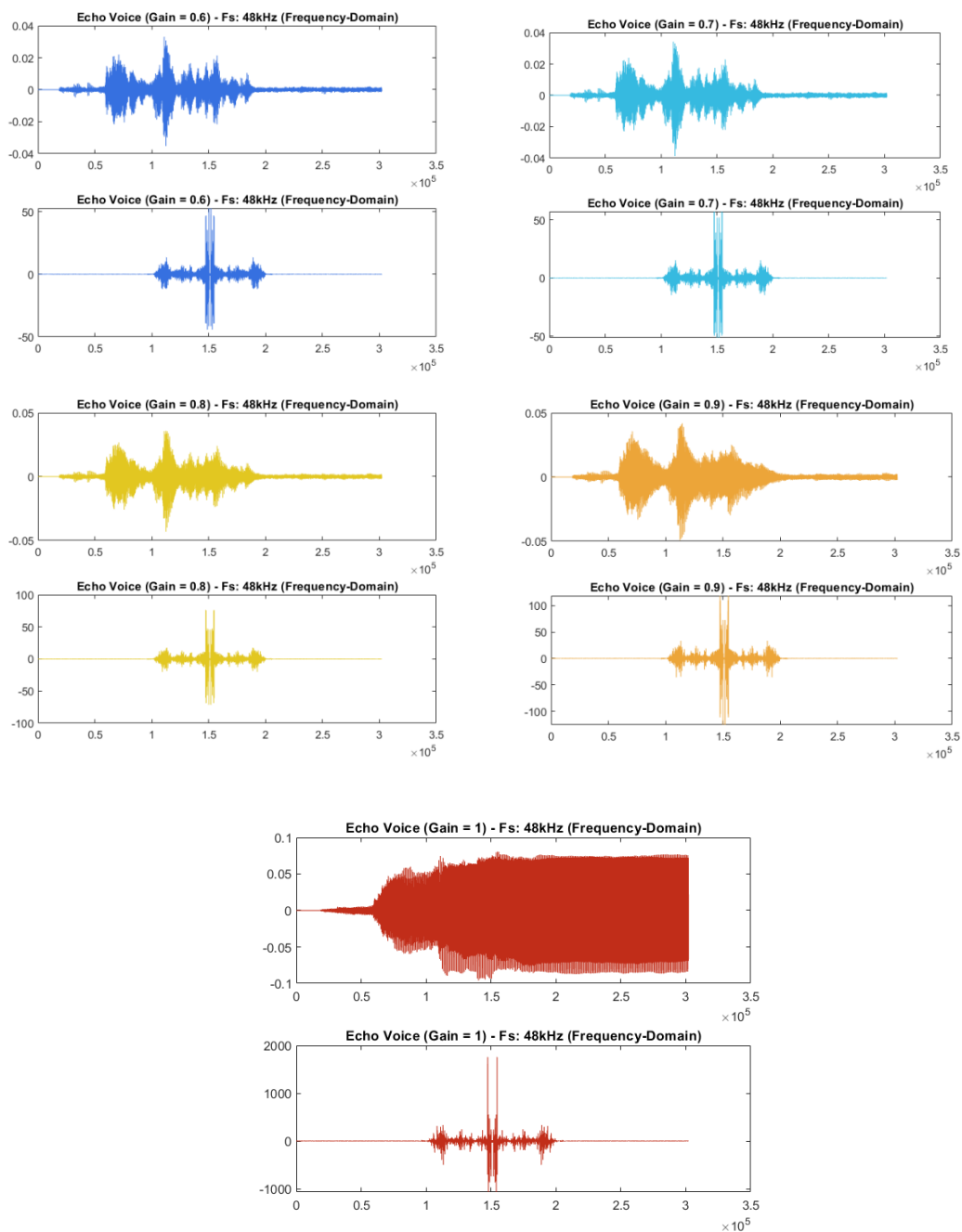
در صورتی که gain سیستم را به صورت پلکانی از ۰.۵ تا ۱ زیاد کنیم اثر ناپایداری سیستم اکو در $gain = 1$ دیده می شود . ناپایداری به این دلیل اتفاق می افتد که ورودی پس از قرار گرفتن در سیستم فیدبک بعد از ۱۰۰۰ نمونه تاخیر کاملاً کپی شده و دوباره به forward-path وارد می شوند. بدین صورت از زمان t اگر نمونه NO در حال طی مسیر feed forward است نمونه NO-1000 پس از یک تاخیر ۱۰۰۰ تایی دوباره به سیستم وارد شده و دامنه این دو نمونه جمع می شود (بدون ضریب کاهشی برای نمونه تاخیر یافته) و همچنین نمونه ای که ۲۰۰۰ واحد عقب تر بوده هم با $gain = g^2$ هم در جال ورود به سیستم است و به همین صورت الی آخر نمونه هایی که اختلاف های ۱۰۰۰ تایی با هم دارند به صورت بر هم نهی شده در خروجی ظاهر می شوند.

در واقع خروجی بر هم نهی تعدادی سری توانی است که به شرط $gain < 1$ همگرا خواهند شد و در غیر این صورت به بی نهایت میل خواهند کرد و خروجی به مرور زمان بزرگ و بزرگ تر می شود و سیستم ناپایدار می شود.

اما در صورتی که مقدار gain زیر عدد ۱ باقی بماند، این باعث می شود محتوای نمونه های خروجی به هم نزدیک تر شوند و نوعی فیلتر lowpass اما با gain اندازه داشته باشیم که در تصویر آخر هم نسبت اندازه های فرکانس های پایین تر که چند صد برابر فرکانس های بالاتر است دیده می شود .

ناپاداری برای $g > 1$ را می توان به صورت تحلیلی نیز اثبات کرد. می دانیم سیستم وقتی ناپادار است که قطبی خارج دایره واحد در نمودار صفر و قطب داشته باشد. برای اینکه چنین اتفاقی نیفتد باید قطب سیستم ما که برابر با $z_{pole} = \sqrt[N]{g}$ است. نهایتاً روی دایره واحد قرار بگیرد که این یعنی $g = 1$ بشود. برای g های بزرگتر قطب خارج دایره واحد خواهیم داشت و سیستم ناپایدار می شود

در صفحه بعد خروجی سیستم اکو برای gain های 0.5,0.6,0.7,0.8,0.9,1 مشاهده می شود.



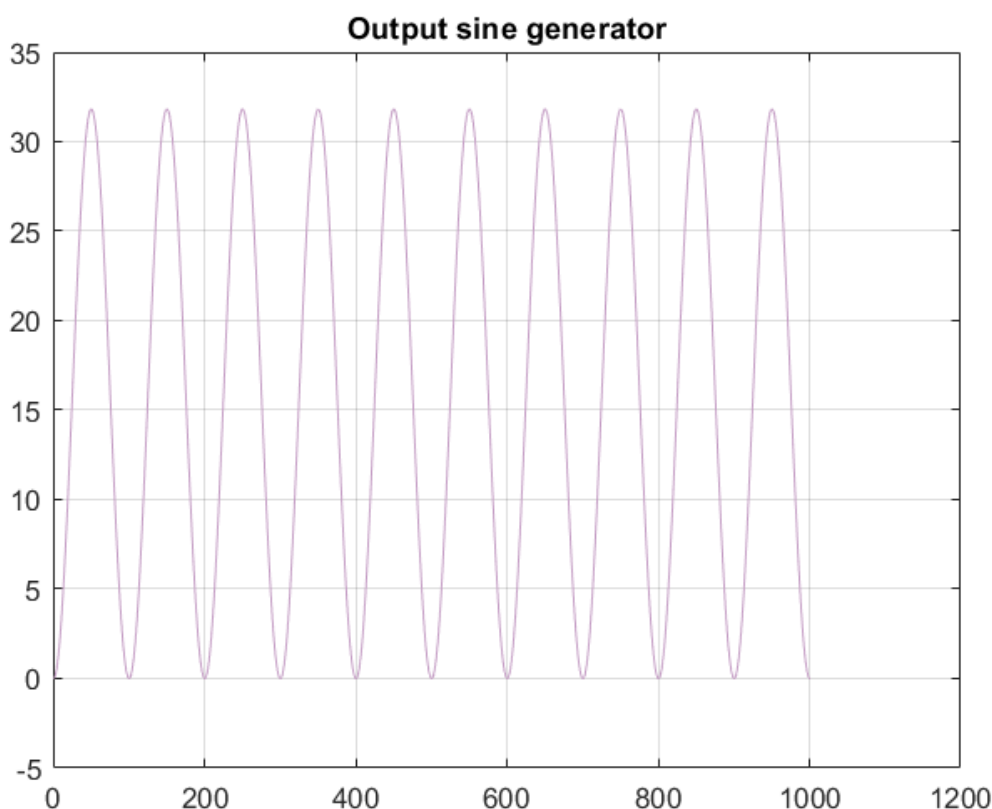
شکل ۳-۱: خروجی سیستم اکو با gain های بزرگتر از 0.5

سوال دوم: تولید سیگنال سینوسی به کمک عبور تابع پالس از فیلتری با پاسخ ضربه سینوسی

در این سوال هدف تولید یک سینوسی با فرکانس دلخواه به وسیله عبور یک تابع دلتای دیراک (پالس) از یک فیلتر با پاسخ ضربه سینوسی در حوزه زمان است. تابع تبدیل فیلتر گفته شده در گزارش کار آمده است. تنها کاری که نیاز است دادن ضرایب صورت و مخرج به یک فیلتر و اعمال کردن آن به تابع ضربه به صورتی که گفته شده است، می باشد. البته سینوسی تولید شده در این روش سنوسی خالص نیست. بلکه یک سینوسی تقویت شده و آفست داده شده است که در خروجی زیر که به ازای:

$$T = 1/50000 \text{ f} = 5000, N = 1000$$

رسم شده مشخص می باشد :



ظاهرا T شده هم دقیقا $1/f$ نیست بلکه scale زمانی رسم سینوسی را تعیین می کند به همین دلیل نتوانستیم آن را $1/f$ قرار دهیم و سینوسی ناقص رسم میشد.

همچنین در قسمت زیر هم تبدیل Z یک سیگنال سینوسی آورده شده است

$$\sin(\omega T n) = \frac{e^{j\omega T n} - e^{-j\omega T n}}{2j}$$

$$z \text{ transform} \Rightarrow \frac{1}{2j} \left(\frac{1}{1 - z^{-1} \times e^{j\omega T}} - \frac{1}{1 - z^{-1} \times e^{-j\omega T}} \right) = \frac{\sin(\omega T) \times z^{-1}}{1 - 2 \cos(\omega T) \times z^{-1} + z^{-2}}$$