

## تکلیف درس DSP

### طراحی یک آشکارساز سیگنال DTMF

در سیستم شماره‌گیری تلفن بر مبنای tone ، برای هر کلیدی که روی صفحه کلید فشار می‌دهید، دو فرکانس ارسال می‌شود (و به همین دلیل آن را DTMF=Dual Tone Multi Frequency می‌نامند). طبق شکل زیر:

Hz	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

یعنی مثلاً اگر روی صفحه تلفن کلید شماره ۵ را فشار دهید، سیگنال صوتی که تولید می‌شود، جمع یک سینوسی با فرکانس ۷۷۰ هرتز و یک سینوسی با فرکانس ۱۳۳۶ هرتز است:

$$x = \sin(2\pi \cdot 770 \cdot t) + \sin(2\pi \cdot 1336 \cdot t)$$

البته در شکل بالا کلیدهای A و B و C و D در بیشتر تلفنهای معمولی وجود ندارند، ولی در استاندارد DTMF هست و کاربردهای خاص خودش را دارد.

در این تکلیف، هدف نوشتن برنامه‌ای است که از روی یک سیگنال صوتی دیجیتال تشخیص دهد که چه شماره‌ای گرفته شده است. این تکلیف دو قسمت دارد:

الف) نوشتن برنامه برای کار کردن به صورت آفلاین: در این حالت برنامه یک فایل صوتی را، که حاوی صدای شماره‌های گرفته شده است، دریافت می‌کند و در خروجی اعلام می‌کند که چه شماره‌ای گرفته شده است. به این منظور، تعدادی فایل صوتی (به فرمت wav) هم به شما داده شده است، که صدای شماره‌گیری هستند، و شما باید برنامه‌ای بنویسید که از روی آن بگوید چه شماره‌ای گرفته شده است.

ب) نوشتن برنامه برای کار کردن به صورت آنلاین (با pyAudio) که در تکلیف قبلی با آن آشنا شدید: در این حالت برنامه ورودی صوتی خودش را از میکروفون می‌گیرد. مثلاً با موبایل یا تلفن ثابت خود

شماره می‌گیرید و در حین شماره‌گیری آن را نزدیک میکروفون کامپیوتر نگه می‌دارید، و برنامه شما از روی صدای دریافتی از روی میکروفون و به صورت آنلاین شماره‌هایی را که می‌گیرید روی صفحه کامپیوتر می‌نویسد. برای آشکارسازی سیگنال DTMF روشهای گوناگونی وجود دارد. روشی که شما در این تکلیف انجام می‌دهید، عبور دادن شکل موج دریافتی از تعدادی (۸ عدد) فیلتر bandpass است که هر کدام فقط یکی از فرکانسهای بالا را عبور می‌دهند. سپس از روی خروجی این فیلترها، می‌توانید تشخیص دهید که شماره‌ای گرفته شده است یا نه، و اگر بله چه شماره‌ای.

فرکانس نمونه‌برداری در فایل‌های داده شده، 8KHz یا به عبارت دقیقتر ۸۱۹۲ هرتز است. همچنین طول مدت فشار داده شدن هر کلید، بطور تصادفی عددی بین 0.2 تا 0.5 ثانیه است. در نتیجه برنامه شما نباید به این عدد حساس باشد (یعنی مثلاً اگر یک کلید به مدت ۱۰ ثانیه هم به طور پیوسته نگه داشته شود، شما فقط یک رقم حساب می‌کنید). بین هر دو رقم، 0.2 ثانیه سکوت وجود دارد. ۵ فایل صوتی به شما داده شده است: در اولی هیچ نویزی با سیگنال جمع نشده است. در ۴ تای دیگر، سیگنال صوتی با نویز سفید جمع شده، به طوری که سیگنال به نویز در آنها 10dB، 20dB، 30dB و 0dB است.

مشاهده می‌شود که شما برای حل این مسئله ابتدا به ۸ فیلتر Bandpass نیاز دارید. که آنها را می‌توانید با ابزارهای مختلف (مثلاً FDatool در متلب یا pyDFA در پایتون، یا وبسایتهایی که برای اینکار هستند (و یکی از آنها در درس هم معرفی شد) استفاده کنید.

همچنین برای اینکه تشخیص دهید خروجی فیلترها انرژی دارند یا نه، می‌توانید خروجی فیلترها را به توان ۲ رسانده، و از یک LPF با فرکانس عبور ۲۰۰ هرتز و فرکانس قطع ۵۰۰ هرتز عبور دهید (تا کوچکترین فرکانس DTMF که ۶۹۷ هرتز است از آن عبور نکند). این فیلتر را نیز با همان ابزارها طراحی کنید.

در گزارش نهایی، شماره گرفته شده را برای هر کدام از این ۵ فایل صوتی ذکر کنید. توضیح دیگری در گزارش نهایی لازم نیست، فقط این ۵ شماره را ذکر کنید، و به سوال زیر هم جواب بدهید. همچنین فایل پایتون برنامه، و نیز فیلترهای طراحی شده (مثلاً با فرمت mat که در متلب هم قابل خواندن است) را نیز باید تحویل دهید. همه فایلها را با هم به صورت یک فایل زیپ تحویل دهید.

**سوال:** اگر فیلترهای FIR را خیلی با شرایط سنگین طراحی کنیم (تا با دقت خیلی خوبی فیلترهایی bandpass بشوند)، و در نتیجه طول آنها مثلاً ۵۰۰۰ نمونه بشود، چه مشکلی پیش می‌آید؟ بخصوص دقت کنید که اگرچه در فایل‌های صوتی ما، حداقل دوره یک شماره 0.2 ثانیه است، در استاندارد DTMF دوره آن می‌تواند به کمی ۵۰ میلی‌ثانیه (و دوره سکوت بین شماره‌ها به کمی ۵۰ میلی‌ثانیه) هم باشد.

توجه ۱: در حالت آنلاین، دقت کنید که حداقل باید یک چانک از ورودی قبلی را در حافظه نگه دارید (بسته به طول فیلتری که طراحی کرده‌اید). وگرنه مثلاً نمی‌توانید فیلترشده اولین نمونه چانک جدید را حساب کنید.

توجه ۲: همه فیلترهای مورد نیاز این تکلیف را با روش Parks-McClellan طراحی کنید.

موفق باشید

مسعود بابایی‌زاده