

## فاز دوم شبیه سازی:

در فاز دوم چند بار کد های متفاوت را پیاده سازی کردم بهینه ترین حالتی که کد پیاده سازی شد در DTMF2 است ورژن ۱ و ۲: ابتدا سیگنال را به چند زیر آرایه تبدیل میکنیم که هر کدام 0.11 ثانیه است و سپس در هر آرایه بررسی کردم. سپس index 8 مربوط به فرکانس ها را پیدا میکنم و در لیست Xf میریزم این قسمت از کد به بعد مرحله ی قبل شبیه سازی فرق دارد. سپس آرایه ی Xf را که در هر خانه ی آن مقدار فرکانس های مذکور است به تابع find\_key میدهم و علاوه بر آن مقدار میانگین آن قسمت از سیگنال را هم به تابع میدهم تا ضریبی از آن یک threshold برای بررسی فشرده شدن دکمه باشد. در تابع find key از فرکانس های پایین ماکزیمم و از فرکانس های بالا هم ماکزیمم را پیدا میکنم و میانگین این دو مقدار را با lower bound مقایسه میکنم در صورتی که شرایط صدق کند آن را به عنوان کلید فشرده شده در نظر میگیریم. یک مورد دیگر هم بررسی میکنم برای اطمینان از درستی جواب آن هم این که در صورتی که جواب الان با دوتا قبلی یکی بود و بین شان کلیدی تشخیص داده نشده بود آن کلید را فشرده شده در نظر نمیگیرید زیرا احتمالاً آن قسمتی که کلیدی فشرده نشده اشتباه محاسباتی بوده است و اگر اشتباه نباشد در بازه ی بعدی کلید فشرده شده را تشخیص میدهد.

در DTMF2\_2 کد مشابه ورژن قبلی است که بازده ای آ» پایین تر است.

ورژن ۳- کد این ورژن که ورژن نهایی کد است و خطایی که روی دیتای predict گرفته است ۱.۳۷ است. در این ورژن به جای این که از میانگین قطعه های سیگنال برای ارزیابی فشرده شدن یا نشدن سیگنال استفاده کنم از دامنه ی فرکانس برای ارزیابی این کار استفاده میکنم به این صورت که اگر دامنه ی هر کدام از فرکانس ها از عددی کوچکتر باشد قدرت آن فرکانس به اندازه ی کافی نبوده است و وجود آن را در نظر نمیگیرم. که کد این ورژن در DTMF2\_v3 است.