Signal & Systems DTFM

Arshia Afzal _

First we design 8 filters for 8 frequencies 697Hz-1633Hz like the figure below:

And there are 7 filters like this one for other frequencies.

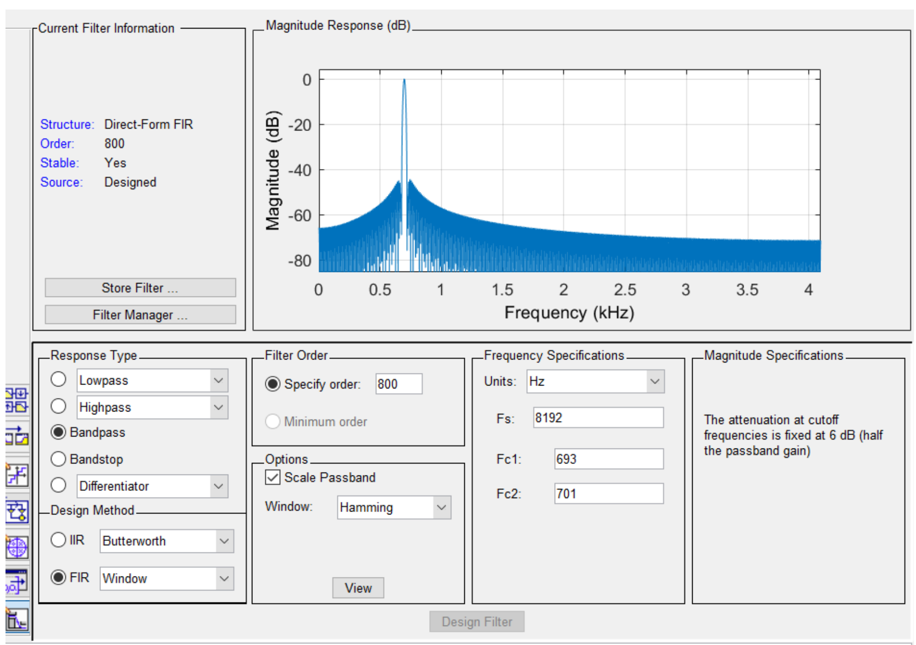

Figure 5: designing filter (697Hz)

And now we build another filter named Nr (noise removal) which is the sum of these 8 filters and we use it to remove the noise.

Here we have a figure which contains the frequency responses of these 8 filters:

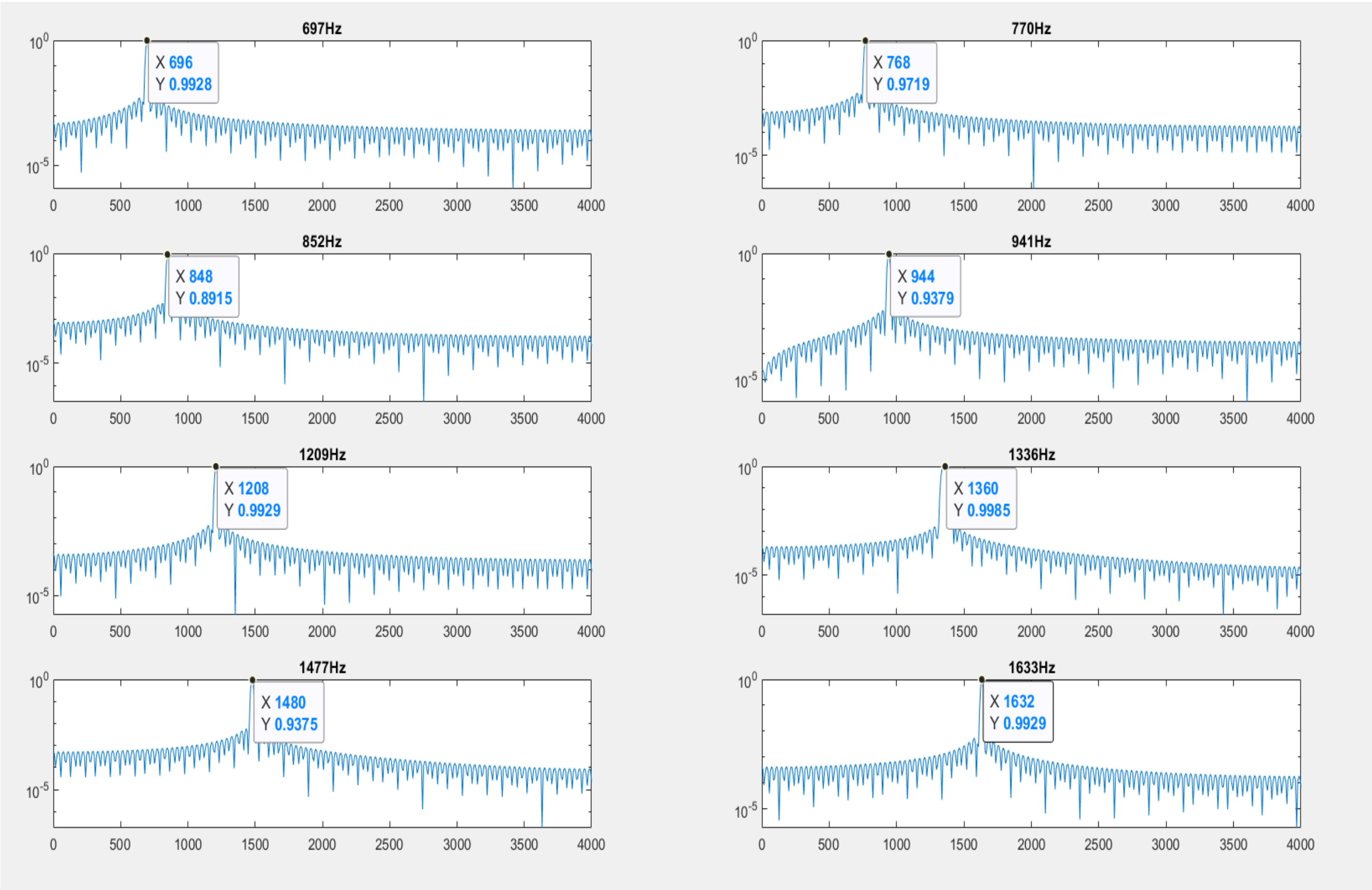

Figure 5: frequency responses of the 8 filters

After removing the noise (using the function Nr):

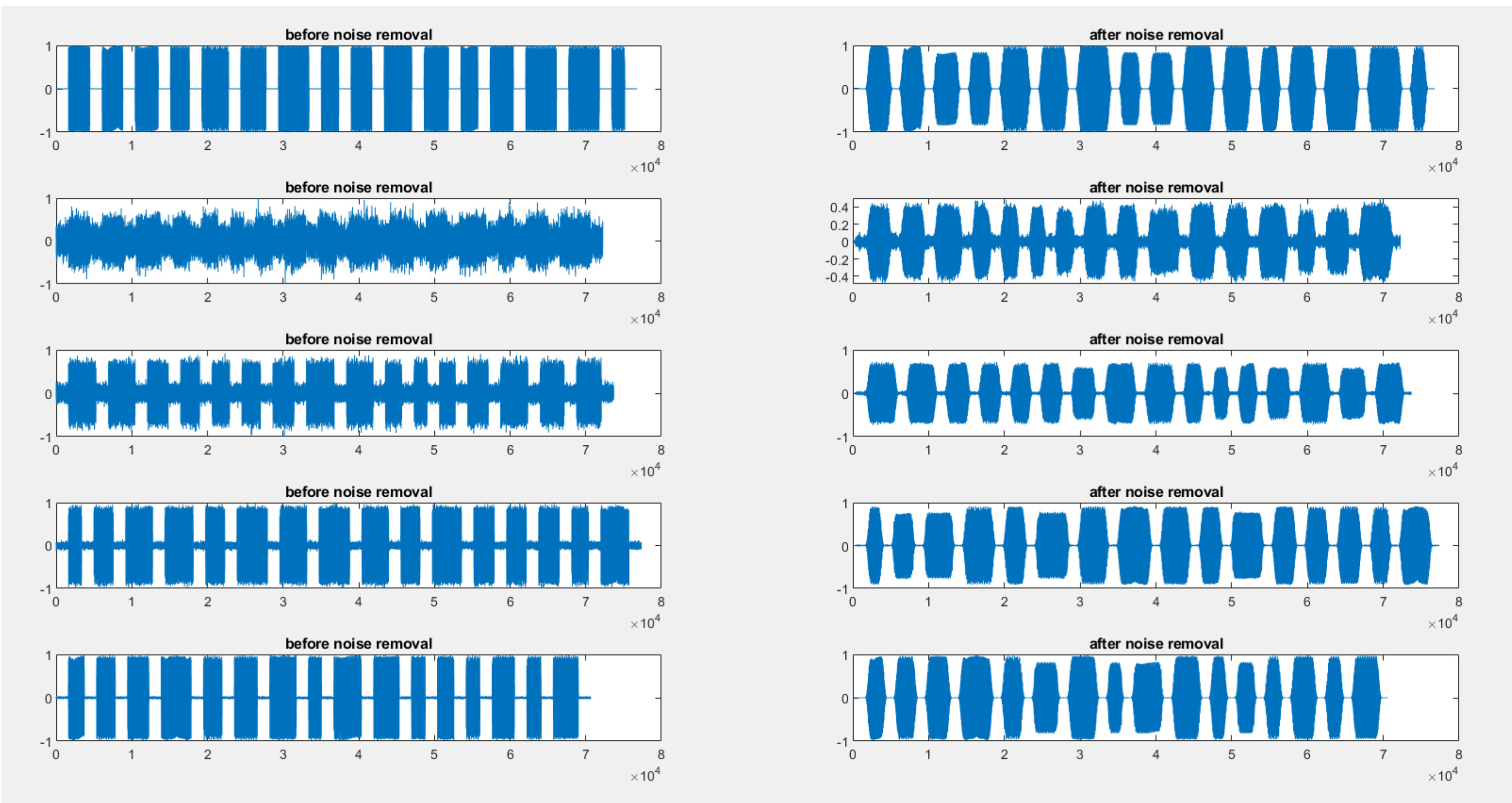

Figure 6: the signals before and after noise removal

By using the self-defined function *find digit* we find out which part of our signal (from the 16 parts) is equal to which digit by dividing that part into all possible paired frequencies (which can be from digit 1 to letter D), and by using this method we finally have the right number (letter or symbol) for each part of our signal.

And also by using the function named *number* we divide our signals into 16 parts by using the energy of our signal (sum(y^2)) and finding out which parts of our signal contains actual data and which parts are just noise since the energy of noise is much smaller than the main data and after we divide our signal into 16 parts we use the previous function to find the right digit for each part. And finally representing the answer for all 16 parts.

The results are:

```
nonoise =

    "number is : 4#206A78513*9DBC"


snr00dB =

    "number is : A13D475BC296#80*"


snr10dB =

    "number is : D176A354C90#8B2*"


snr20dB =

    "number is : 325640B*9187ADC#"


snr30dB =

    "number is : #A7*60452918C3BD"
```

Nonoise : 4#206A78513 * 9DBC

Snr00dB : A13D475BC296#80 *

Snr10dB : D176A354C90#8B2 *

Snr20dB : 325640B * 9187ADC#

Snr30dB : #A7 * 60452918C3BD