Signals & Systems Project 1

Sonar Digital Communication & Matched Filters

Joy Yeh, Ting Li

Contents

1 Part 1

We approach this part by first loading the active input and plotting the original ping and the echo. **fliplr** is used to flip the ping and a convolution between the ping and the echo is performed using **conv**.

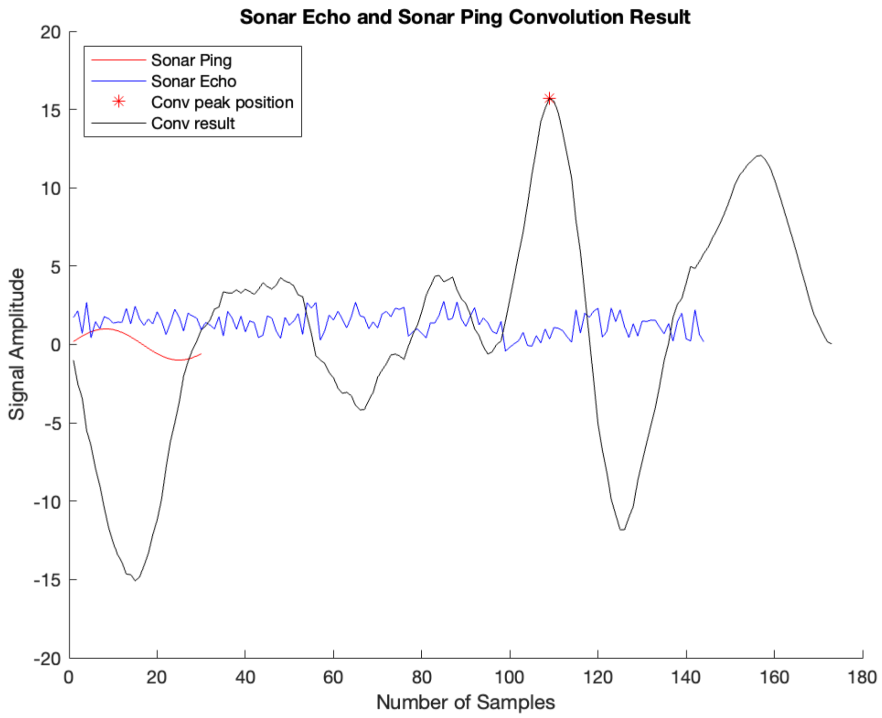

Figure 1: Sonar Convolution Result

The distance is determined by locating the maxima of convolution result Figure 1. The index of the peak we get is 109.

$$\begin{aligned} \text{distance} &= v_{sound} \cdot \frac{(\text{index} - 1)}{2 \cdot \text{samples\_per\_second}} \\ &= 5000 \times \frac{(109 - 1)}{2 \times 100} \\ &= 2700 \text{ ft} \end{aligned}$$

Therefore, the distance between this ship and the other ship is 2700 ft.

2 Part 2

For part 2, a **decode(sig_received, fs, pulse)** function is written. The optional **pulse** defaults to 0.3s unit step pulse. A filter is made by taking the pulse into **fliplr**, and the filter convolves with the **sig_received**. Subsequently, an empty bin string container of appropriate size is created. The decode the process is made possible by comparing the binary string with the ASCII table. The binary strings are:

- 010100110100111101010011
- 0100100001100101011011000111000000100001
- 010011100110010101110110011001010111001001101101011010010110111001100100

Compared with the ASCII file, we get that:

- Message 1: SOS
- Message 2: Help!
- Message 3: Nevermind

3 Part 3

Message: "Signals and Systems Sonar Project 01"

3.1 Clean and Noisy Signals

We used a ramp pulse signal with duration of 0.3 s and sample frequency of 100 samples/second. $s(t)$ generated with the triangular pulse wave is plotted in blue below. The noisy signal with a noise scale of 0.9.

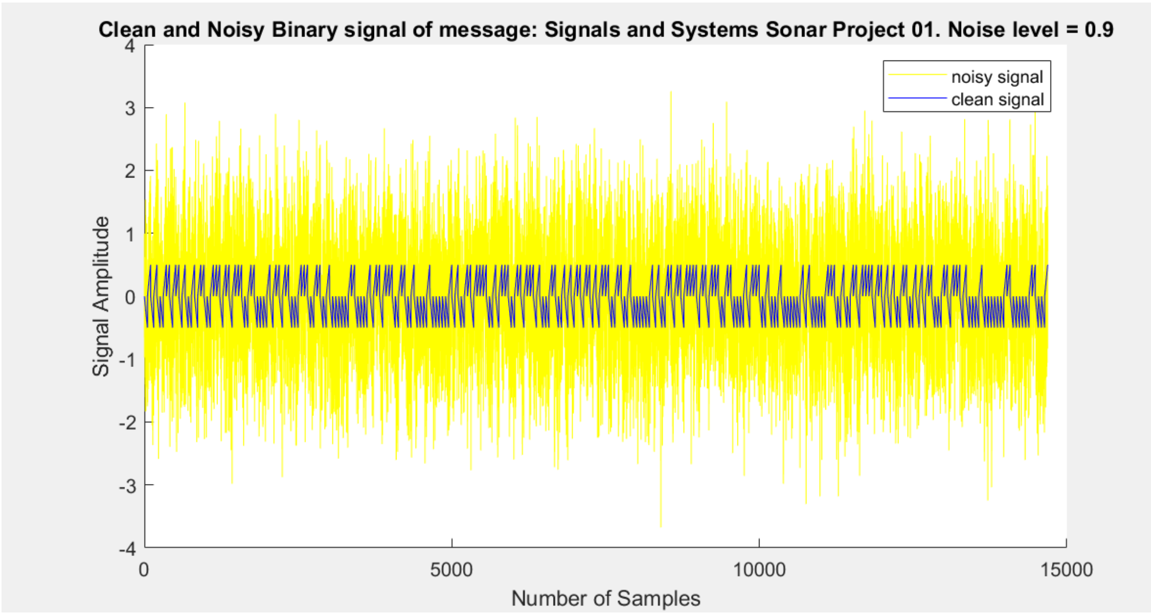

Figure 2: Noise Level = 0.9

3.2 Signal Recovery

After calling the **decode**() function, we are able to recover the message, as the following code snippet and command window prompt shows:

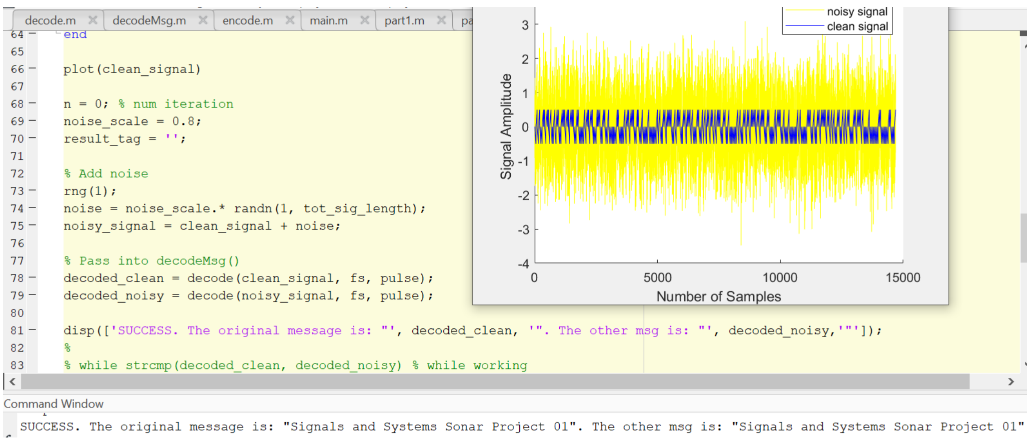

Figure 3: Recovery

The command prompt result shows that the input and output messages are the same.

Code to generate clean and noisy signals based on the binary string:

```
clean_signal = [];
for idx = 1 : bin_len
    if bin_str(idx) == '1'
      clean_signal = [clean_signal, pulse];
    else
      clean_signal = [clean_signal, -pulse];
    end
end

plot(clean_signal)

n = 0; % num iteration
noise_scale = 0.8;
result_tag = '';

```

Figure 4: Code Snippet

3.3 Comparisons between Different Filter Wave Pulses

In this section, we gradually increases the noise_scale parameter from 0.8 (where the actual additive noise amplitude will be $0.8 \times 1 = 0.8$, which is slightly below 1).

The iterations will run until the inputs and outputs are no longer the same. For example:

```
decode.m  decodeMsg.m  encode.m  main.m  part1.m  part2.m  part3.m  +
while strcmp(decoded_clean, decoded_noisy) % while working
    rng(1)
    noise = noise_scale.* randn(1, tot_sig_length);
    noisy_signal = clean_signal + noise;

    % Pass into decodeMsg()
    decoded_clean = decode(clean_signal, fs, pulse);
    decoded_noisy = decode(noisy_signal, fs, pulse);

    result_tag = ' SUCCESS';
    n = n + 1;
    noise_scale = noise_scale + 0.05;

  disp(['Iter: ', num2str(n), ': ', result_tag, '. Noise Scale: ', num2str(noise_scale)]);
```

```
Command Window
Iter: 13:  SUCCESS. Noise Scale: 0.75
Iter: 14:  SUCCESS. Noise Scale: 0.8
Iter: 15:  SUCCESS. Noise Scale: 0.85
Iter: 16:  SUCCESS. Noise Scale: 0.9
Test ended.
Pulse type: ramp.
FAILED at iteration #16. Noise Scale = 0.9.
The original msg is: "Signals and Systems Sonar Project 01" but the decoded msg is: "Signals and Systems Soner Pro*ect 01".
```

Figure 5: Demonstration of Iterations

While **strcmp(decoded_clean, decoded_noisy)** are the same, the command window keeps printing the iteration number and the current noise scale.

When they are no longer the same, we reach the decoding limit of this specific filter. In this case, the "ramp" filter can handle the maximum noise level of 0.9.

We ran the same tests with other pulse types:

1. Unit Step. Max noise level = 3.8

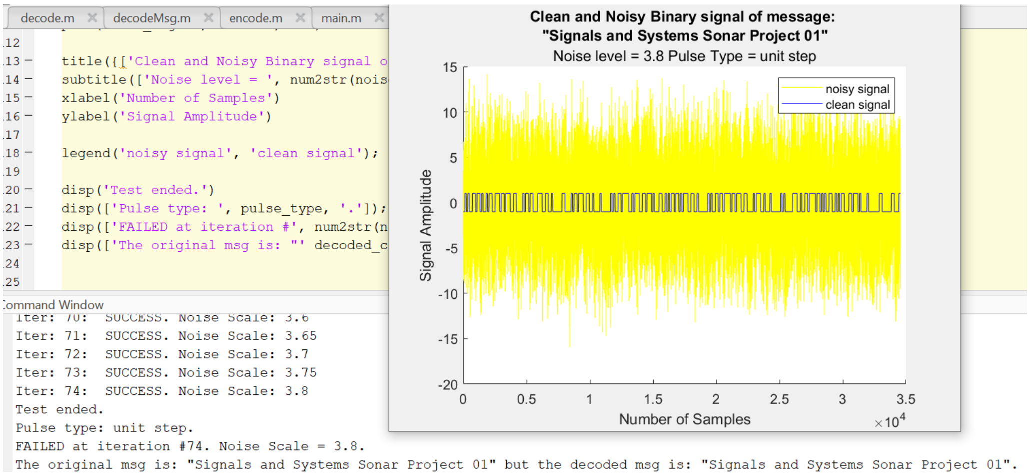

Figure 6: Unit Step

2. Unit Ramp. Max noise level = 0.9

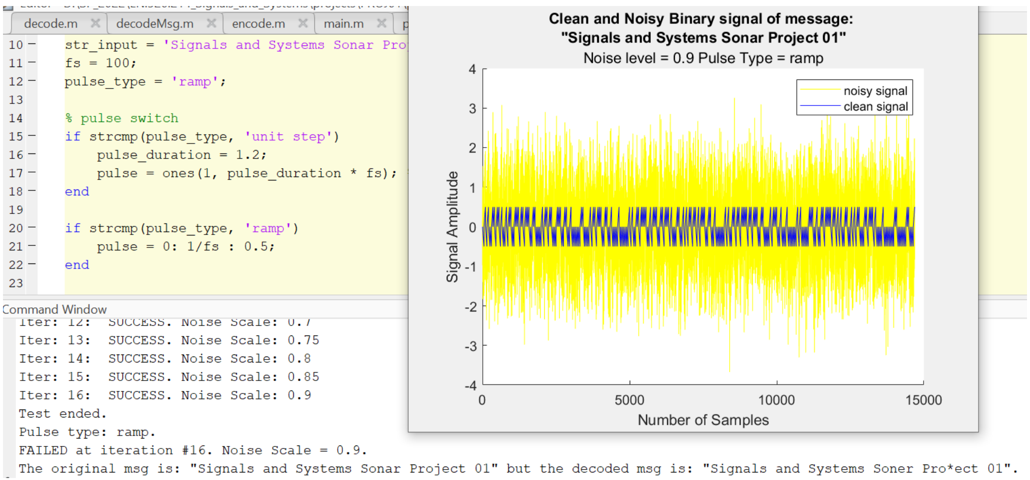

Figure 7: Unit Ramp

3. Sine Signal. Max noise level = 1.75.

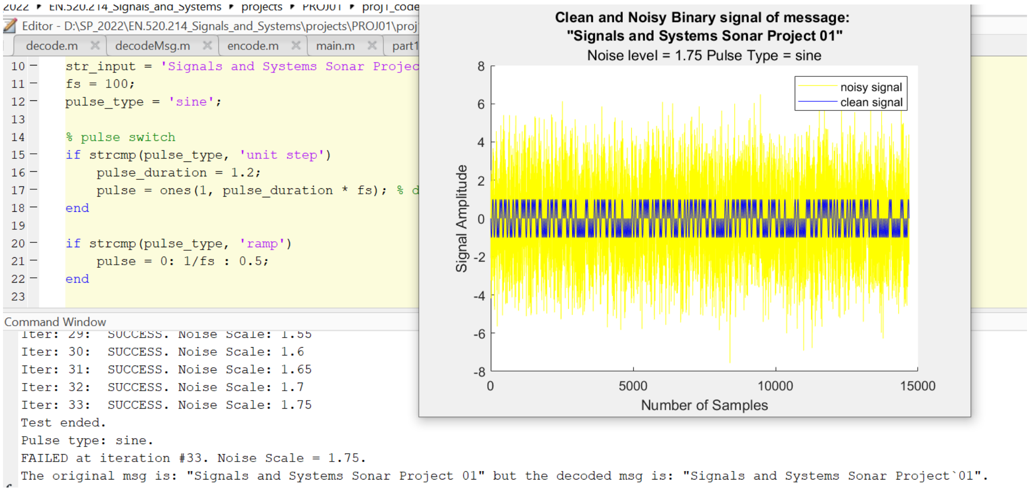

Figure 8: Sine Signal

4. Fence Signal (randomly generated pulses within the duration of one pulse). Max noise level: 0.1.

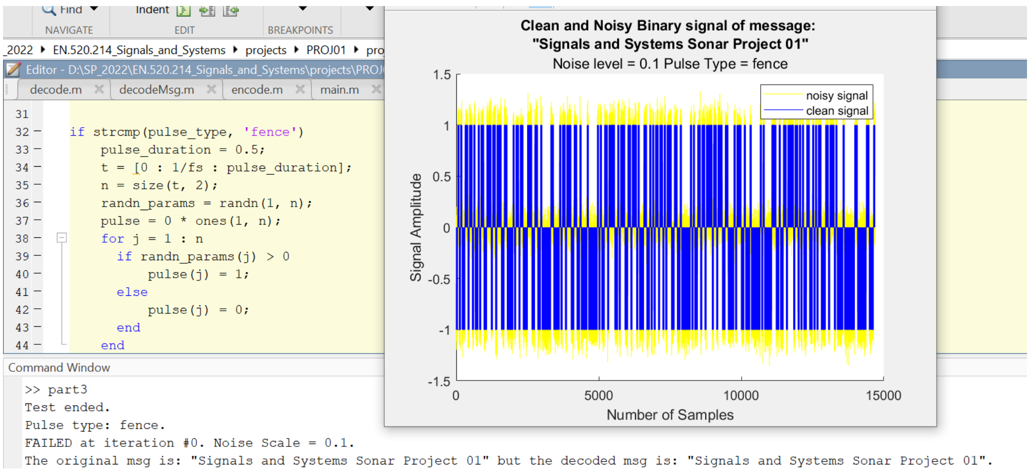

Figure 9: Fence Signal

Pulse Type	Max Noise Scale
Unit Step	3.8
Sine	1.75
Unit Ramp	0.9
Fence	0.1

In this sense, the unit step signal is the most robust.