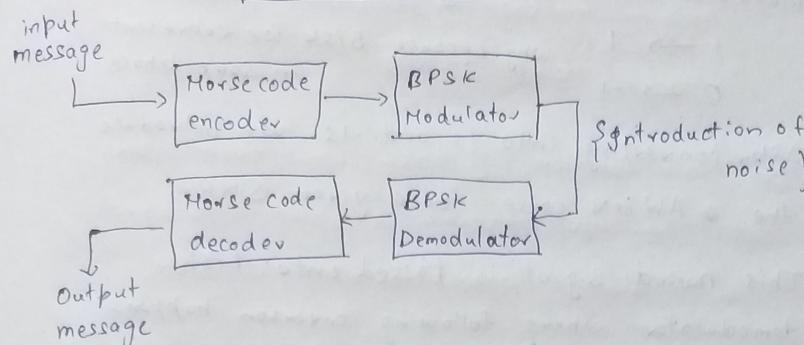


EE 309: Introduction to Communication Systems

- Amitabh Swain

- 180020002

Project component -1



Functions

- (i) Morsecode_encoder → Morse code encoder
- (ii) BPSK_modulation → BPSK Modulator + Introduction of noise
- (iii) BPSK_demodulation → BPSK Demodulator
- (iv) morsecode_decoder + Morse_search → Morse code decoder
- (v) plot_word_error → Word error rate vs SNR - Plotter
- (vi) plot_letter_error → Letter error rate vs SNR - Plotter
- (vii) Morse_main → Main function which contains all above functions

Approach

① Sender inputs his message which the person wants to send.

- ① The Morse encoder converts the characters of the message in following way.

• → |

— → |||

Space between parts of the same letter → 0

Space between letters → 0 0 0

Space between words → 0 0 0 0 0 0 0

④ Finally, the morse encoder converts the message in the form of 0s and 1s.

⑤ This encoded message is then passed onto BPSK modulator where following conversion happens:

$$\begin{aligned} 1 &\rightarrow 1 \quad \text{why because BPSK we represent} \\ 0 &\rightarrow -1 \quad \text{in same phase.} \end{aligned}$$

⑥ After converting into 1s and -1s, we introduce the AWGN noise for a given SNR.

⑦ This noised signal is passed onto BPSK demodulator where following conversion happens:

if signal bit < 0, assign 0 to it
else assign 1.

This motivation is based on the distance factor.

⑧ This demodulated signal is then passed onto Morse decoder, where will try to retrieve back our original ~~message~~ message.

We will use stack to find our decoded message comments in code are self explanatory.

Basically, I am trying to search for ~~multiple~~ spaces then according to that the message is getting decoded in a character wise manner.

I am using Morse Search function to match the code with corresponding character. ~~and~~ Then appending that character to decoded message

⑩ When Morse search is not able to find the character, g will assign '@' to the character. This is because due to distortion ~~of noise of~~ caused by noise. So bits get changed due to this and are not able to detect.

⑪ This everything above g need to do loop for 1000 times for a given SNR because due to the noise everytime we get a different signal. Whenever, the ~~original~~ loop, ~~original~~ message does not match with demodulated message, g increment the error by 1.

Finally dividing the total error for ~~the~~ each letter by 1000 to get the average error.

Now, this we need to do for every letter in our message and finally get final error for every characters. Dividing by ^{total no.} letters length will get the average final error for a given SNR.

⑫ Now, ~~this~~ ^{do} the above we need to, for SNR from 1 to 25 with a step size 0.1 And plot the letter error rate against SNR.

Similarly for words, we will plot words error rate against SNR.

⑥ Now why SNR plot is till 25 because

SNR = 25 dB is recommended for effective communication so that reconstruction of original message is possible in the range 15 dB - 25 dB

⑦ In Main function, g take input from user.

The input message is split into words for calculating word error rate vs SNR,

The words are split into letters for calculating Letter error rate vs SNR.

And finally the output will be the

plots: Letter error rate vs SNR,

word error rate vs SNR

and the decoded message.

Note: when getting input from the user, g

assume they don't enter any other characters

which is not in Morse code template.

As, the person ~~is~~ is highly trained, ~~the person~~ would be knowing this stuff like how to communicate the message by entering proper input required for transmission also no double spaces

Overall, a good project!

and
spaces in
Starting and
ending.