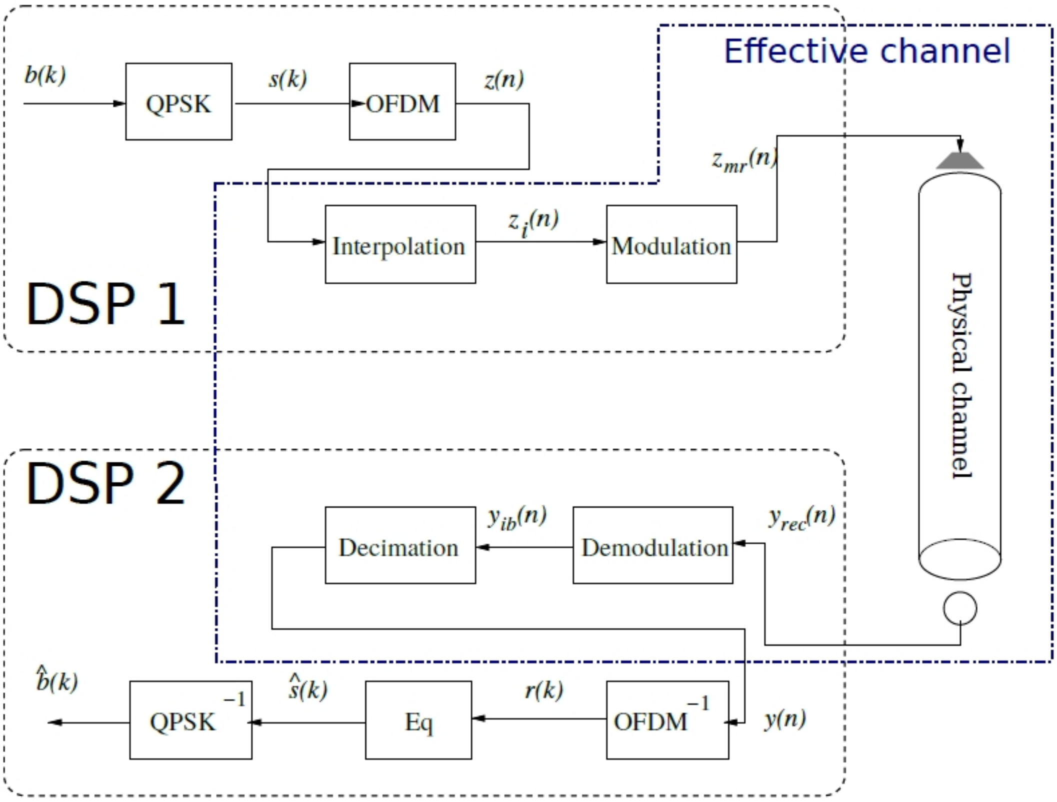
SSY130 - Project 1B
Interpolation, Modulation, Demodulation and Decimation

Diego Jauregui   Lucas Rath   Sondre Wiersdalen   Mohammad Shaikh

December 4, 2018

student id: 19930425
secret key: Bouffalant



Matlab Implementation

(1) If the desired up-sampling frequency is chosen to be 16 kHz and a up-sampling factor 8 is used, then the signal bandwidth is 16/8=2 kHz. Clearly, the up-sampled data will be periodic with period 16 kHz and applying a low pass filter with a pass-band-width of 2 kHz will result in a signal with approximately 2 kHz bandwidth and centered in zero, i.e. the signal will present relevant magnitudes between -1 and 1 kHz. The described procedure can be seen in figure 1 and 2.

Further, if we modulate the signal in 4 kHz, the transmitted audio signal band will lie between [-1 1]+4, i.e. between 3 and 5 kHz. Unfortunately, we can not use a ideal filter because it would require infinite memory and processing time and therefore the transition- and stop-band will originate some considerable magnitudes outside the range w=[3 5] kHz.

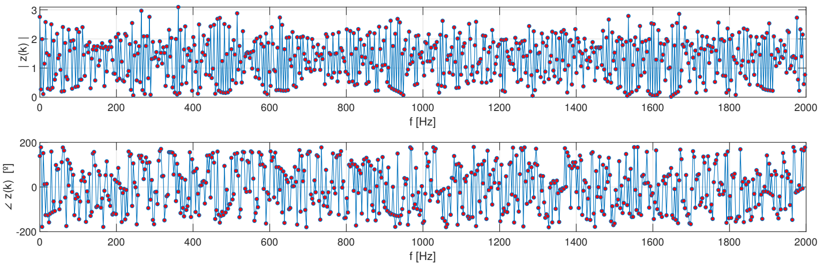


Figure 1: *FFT of the original signal to be transmitted, containing the pilot, data and correspondent cyclic prefixes*

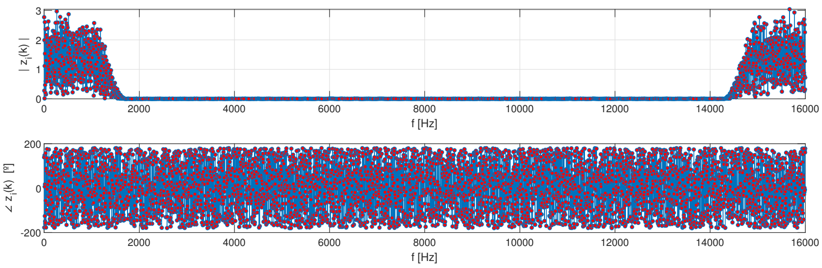


Figure 2: *FFT of the signal after up-sampling using factor 8*

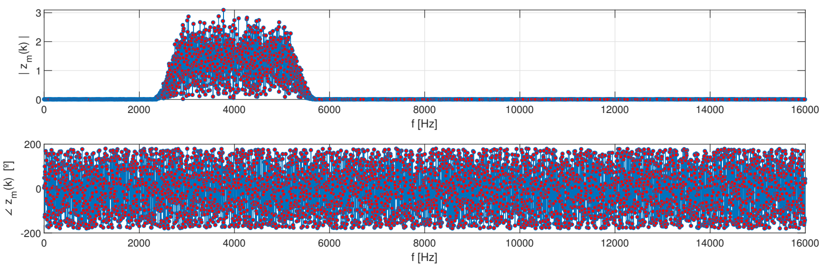


Figure 3: *FFT of the signal after interpolation and modulation of 4 kHz*

---

(2) The effect of nonzero EVM, even when using ideal channels, happens because we are interpolating our signal using a non-ideal low-pass filter. Analyzing figures 1, 2 and 3, one can easily see that all the original information in frequency domain we have is contained in a bandwidth of 2 kHz. However, since we are applying a non-ideal filter in the interpolation procedure, we will still have some high magnitudes beyond our pass-band of 2 kHz, located mainly in the transition band. This extra signals with higher frequencies will then cause aliasing and distort our signal when down-sampling the signal back to the "original" sampling rate, that is equal to the band-width.

In figure 4 we show the FIR filter designed to work with a up-sampling ratio $L = 8$. Clearly, it will start to attenuate all the frequencies beyond $1/L * f_s/2 = 0.125 * 16000/2 = 1kHz$. However, we still have considerable energy outside the pass-band region.

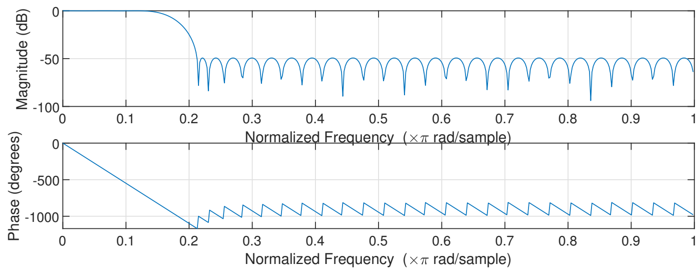


Figure 4: *Designed equiripple FIR filter used for interpolating the signal. In this case, the filter was designed for up-sampling factor 8*

---

(3) After making some test, we first verified that the modulation and demodulation blocks do not contribute at all to the channel H, simply because the effect of modulation is completely removed when demodulating the signal.

Taking the real part will however contribute to the channel H. When taking the real part, we are actually averaging the spectrum with its conjugate $Z_r(k) = (Z(k) + \bar{Z}(k))/2$. After removing the conjugate in the decimation stage, we lose 50% of the magnitude, something that is compensated later with the channel H.

Other distortions that occur are mainly attributed to the all convolutions of the signal with filters. As verified in the last items, the low-pass filters applied during interpolation and decimation are not ideal and are the most source of interference in the received symbols. As shown in figure 5, H will be able to capture the effect of pass-ripple distortions, which are more intense close to the cut-off frequency (in this case 1 kHz). In addition, it also captured the almost linear (would be for an ideal filter) phase distortion generated by the filters in the pass-band region.

It is also important to observe that the down-sampling will reduce the magnitude of the signal, dividing it by the up-sampling factor L. As a result, the magnitude will be reduced by $1/(L*2) = 1/(8*2) = 0.0625$ due to the combination of taking the real part and down-sampling the signal, exactly what is seen in figure 5 as the base value.

Of course, any additional linear convolution when propagating the signal over the physical channel will be also captured by H.

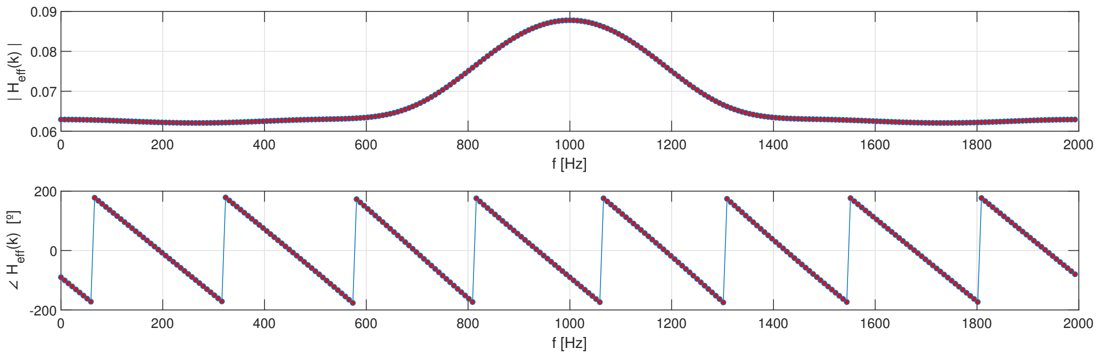


Figure 5: *Channel estimated using system identification in the equalization phase, which occurs after the decimation stage. Data generated for a ideal channel scenario*

---

(4) After interpolating the OFDM block, we got a signal $z_i(n)$ with up-sampled sampling frequency w_s and then we modulate the signal in f_m Hz:

$$z_m(n) = z_i(n)e^{j2\pi n \frac{f_m}{f_{sup}}} \tag{1}$$

Thereafter, we take the real part of the signal, and analyze the information in frequency domain:

$$Re\{z_m\} = z_{mr} = \frac{1}{2}(z_m + \bar{z}_m) \tag{2}$$

$$Z_{mr}(\omega) = \frac{1}{2}\left[Z_i\left(\omega - \omega_s\frac{f_m}{f_s}\right) + \bar{Z}_i\left(\omega - \omega_s\frac{f_m}{f_s}\right)\right] \tag{3}$$

According to the relation $\bar{Z}(\omega) = Z(-\omega)$, one might conclude that:

$$Z_{mr}(\omega) = \frac{1}{2}\left[Z_i\left(\omega - \omega_s\frac{f_m}{f_s}\right) + Z_i\left(-\omega + \omega_s\frac{f_m}{f_s}\right)\right] \tag{4}$$

According to (4), it is now clear that taking the real part of the signal does not result in loss of information. Instead, we added a mirrored copy of the spectrum around the up-sampled nyquist frequency w_s. Of course, this spectrum is w_s periodic. The resultant spectrum can be seen in figure 6. It follows then, that after channel transmission, demodulation and decimation (figures 7 and 8), the copy generated by the conjugate is removed by the low pass filter and therefore the information is perfectly recovered (except for the effect of reducing the magnitude by half).

It follows that we could also have kept the imaginary part and the information would still be kept. The mathematical expressions that describe this effect are similar to the defined for the real case, but now using these relations $Im\{z_m\} = z_{mi} = \frac{1}{2j}(z_m - \bar{z}_m)$; $Z(\omega) = -\bar{Z}(-\omega))$ instead.

The only condition that must be followed in order to avoid loss or distortion of information is that we must keep the modulated symbols sub-carriers between 0 and the up-sampled nyquist frequency. As can be verified in equation 4, it is fairly easy to show that we must choose a modulation frequency f_m in the following range:

$$\frac{f_s}{2L} \leq f_m \leq \frac{fs}{2} - \frac{f_s}{2L} \tag{5}$$

in order to avoid overlapping between the copies, which would cause overlap between sub-carriers and therefore distort the information.

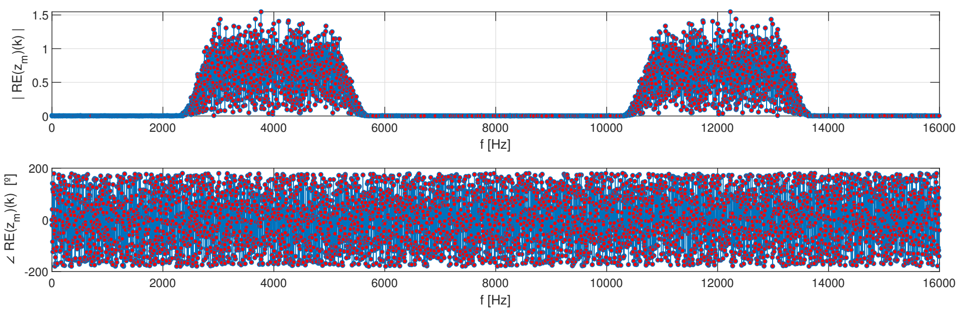


Figure 6: *FFT of the real part of the signal to be transmitted, containing pilot, data and correspondent cyclic prefixes*

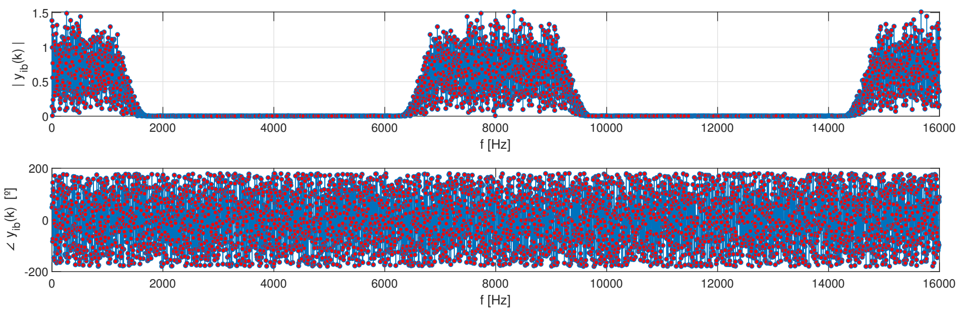


Figure 7: *FFT of the received signal after demodulation of 4 kHz*

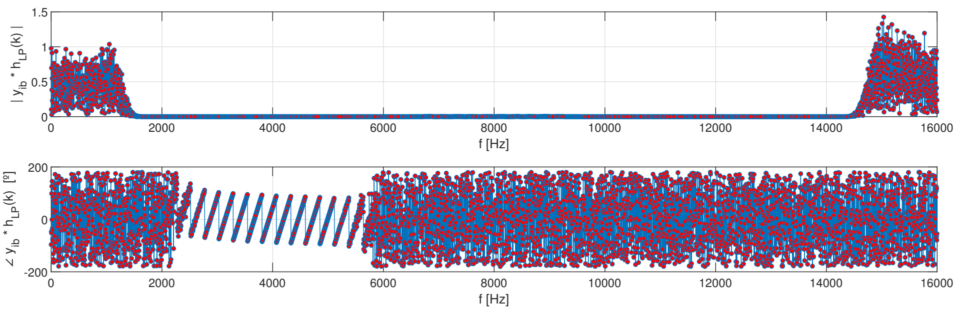


Figure 8: *FFT of the demodulated signal after the LP filter in the decimation block*

---

(5) When designing the low pass filter for both interpolation and decimation stages, the pass-band ripple and the phase linearity properties are not so important because the channel equalization will be able to equalize its effects using the pilot message, as explained in item (3).

The large EVM deviation happens when we have a small stop-band attenuation or a high transition band width, because they will cause aliasing when down-sampling and corrupt the spectrum when we decide to take only the real part of the signal.

Basically, for the **interpolation** stage, we need to care more about the transition band width than about the stop band attenuation. The main reason is because the decimation low-pass filter will remove the ripples in the stop-band region. However, if we discard the imaginary part when transmitting the signal, ripples in the stop-band width may cause interference between sub-channels.

For the **decimation** stage, both stop-band attenuation and transition band width are equally important because, as said before, they will cause aliasing and distortion of the symbols when down-sampling.

DSP Implementation

(7) First, we show below the magnitude and the phase for each formula for estimating the channel.

$$H = \frac{R}{T} \quad \begin{cases} |H| = |\bar{T}| \ / \ |R| \\ \angle H = \angle R - \angle T \end{cases} \qquad \hat{H} = \bar{T} \cdot R \quad \begin{cases} |\hat{H}| = |\bar{\bar{T}}| \cdot |R| \\ \angle \hat{H} = \angle R + \angle \bar{T} \end{cases} \tag{6}$$

Using the equivalence for complex numbers $\angle \bar{A} = -\angle A$, one can easily see that $\angle H = \angle \hat{H}$. We can use now another important equivalence $|\bar{T}| = |T|$ on equation 6 to get the following relations:

$$|\hat{H}| = |H| \cdot |T|^2 \qquad , \qquad \angle \hat{H} = \angle H \tag{7}$$

As proven the alternative formula for estimating the channel will not change the phase but will scale the magnitude by $|T|^2$. However, as we are transmitting symbols in the QPSK format, the magnitude of the transmitted symbols are always equal 1, so $|T| = 1$ and therefore $|\hat{H}| = |H|$.

---

(8) As given in the description of the exercise:

$$R_{eq} = R \cdot \bar{\hat{H}} \tag{8}$$

Using now the relations obtained in the last item, we have:

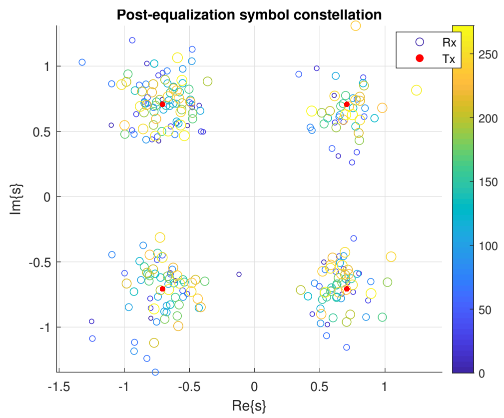
$$\begin{aligned} \angle R_{eq} &= \angle R + \angle \bar{\hat{H}}, \qquad \angle \bar{\hat{H}} = -\left(\angle R - \angle T\right) \\ \Rightarrow \angle R_{eq} &= \angle T \end{aligned} \tag{9}$$

Which proves that the angle of the equalized symbols are the same angle of the transmitted signal.

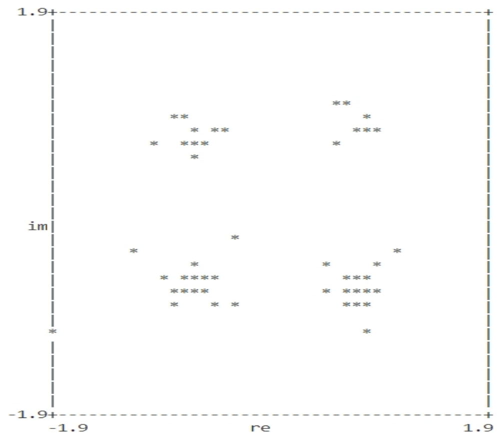
(9) If we keep the speakers close enough to the microphones and use the highest volume amplitude, then we get a very small EVM around 0.05 and most of the time no bit error. As we keep decreasing the amplitude of the transmitted signal, the EVM error increases and the symbols start to be more spread or less precise in relation to the right symbol position in the constellation graph.

What happens is that reducing the amplitude of the signal and keeping (as best as possible) an equal background noise level reduces drastically the *Noise to Signal Ratio*. The problem is that the received signal will contain a noise with an large energy in comparison to the energy of the transmitted message.

This effect is similar of what we get in Matlab simulations using a small SNR parameter of 10 dB, as can be seen in figure 9a.



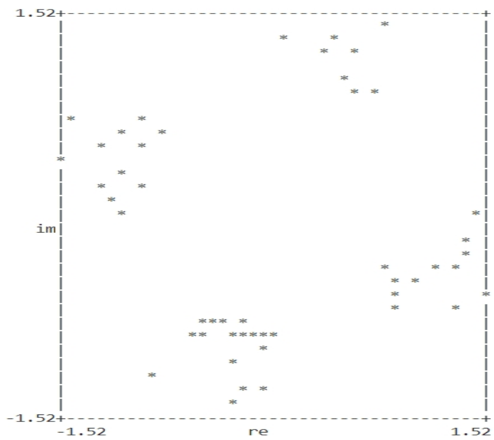
(a) *Post-equalization symbol constellation for SNR=10dB using Matlab*



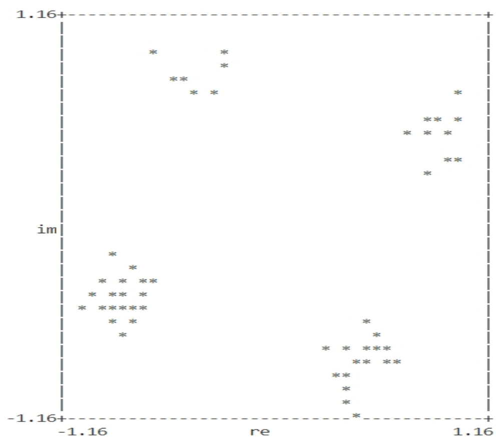
(b) *Post-equalization symbol constellation using DSP*

---

(10) Now, we tried to transmit the data moving the speakers with speed between 2-5 cm/s. We noticed that the symbols are shifted by a positive angle if the motion is directed towards the DSP kit and by a negative value if the motion is directed further away. Both effect can be seen in figures 10a and 10b.



(a) *Speaker movement towards DSP kit*



(b) *Speaker movement away DSP kit*

After few investigations, we concluded that this effect occurs because the pilot and the data are being transmitted by "different" channels. Since the speaker is moving, it is not at the same distance to the DSP when transmitting the pilot and the message. This observed phase shift θ of the symbols in frequency domain can be related to a time delay in time domain:

$$e^{j\,\theta}\;X(w) = e^{j\,w\,t_0}\;X(w) \quad \rightarrow \quad x(t-t_0) \tag{10}$$

This effect can be then attributed to the time difference between each matching sub-carrier of the pilot and the message reaching the DSP. Clearly, the shift $\theta = w \cdot t_0$ will depend on the sub-carrier frequency. However, since it is known that the symbols were transmitted in a band between 3 and 5 kHz, we will use w_{avg}=4000*(2π) rad/s as an average value to then obtain an average time delay $t_0 = \theta_{avg}/w_{avg}$.

We can then calculate t_0 by estimating the phase shift in figures 10a and 10b, in comparison to the true symbol positions (as in figure 9b) that are 45° for the QPSK. In average, we estimated that $\theta_{avg} \approx 20°$=0,35 rad shift for "towards" and $\theta_{avg} \approx -15°$=-0.26rad "away".

Next, as we know that the message is being transmitted at the speed of sound, we can calculate the distance related to this time delay t_0.

$$\Delta d = t_0 \cdot (\text{Speed of sound}) = t_0 \cdot 343m/s \tag{11}$$

We also now that the difference between the time that the first sample of the pilot and the first sample of the message took to reach the receiver is:

$$\Delta t = (\#\ \text{transmitted pilot samples})/f_s = 768/16000 = 0.0479\ s \tag{12}$$

The pilot is composed by 16 char plus 8 char cyclic prefix, and each char corresponds to 4 symbols, resulting in 24*4=96 symbols or alternatively 96 samples in time domain. But, since we up-sampled by factor 8, the pilot was effectively transmitted in 96*8=768 samples.

Finally, the relative velocity between the DSP and the speaker can be estimated by:

$$V_{dsp} = \frac{\Delta d}{\Delta t} \tag{13}$$

The whole procedure can then be summarized in the following algorithm, which resulted in the relative velocities for the two described scenarios:

Listing 1: *Calculating relative velocity between transmitter and receiver*

```
f_s = 16000;             % Hz
w_avg = 2*pi*(4000);     % rad/s
v_sound = 343;           % m/s
pilot_chars = 16+8;      % 16 pilot + 8 cp
char2symbols = 8/2;      % 1 char = 8 bits;  2 bits = 1 symbol
L=8;                     % up-sampling factor

pilot_samples = pilot_chars * char2symbols *L;  % number of transmitted samples

t0 = @(theta) theta/w_avg;
delta_d = @(theta) t0(theta) * v_sound;
delta_t = pilot_samples / f_s;

v = @(theta) delta_d(theta)/delta_t;    % m/s

% Moving to towards DSP (theta=20deg)
v(20/180*pi) *100   % cm/s
%
%    ans =
%         9.9248

% Moving further away from DSP (theta=-15deg)
v(-15/180*pi) *100  % cm/s
%
%    ans =
%        -7.4436
```

Moreover, we can also determine what is the highest possible relative velocity we can achieve in an ideal scenario that will not lead to any symbol misclassification due to this phase shifting effect. Clearly, the highest shift to avoid misclassification is 45°for a QPSK communication scheme. Further, as stated in (10), the sub-carriers with higher frequencies will be the most affected by this phase shifting. The worst case is then when the last sub-carrier at 5 kHz shifts $\pi/4$ rad in phase resulting $t_0 = (\pi/4)/(2\pi\ 5000)$. Following the same procedure as outlined in algorithm 1, we get the maximum relative velocity $v_{MAX} \approx 17.9cm/s$