Chapter 6
Double-Sideband Suppressed-Carrier Amplitude Modulation

Contents

Chapter 6
Double-Sideband Suppressed-Carrier Amplitude Modulation and Coherent Detection

Standard AM contains a sinusoidal component at the carrier frequency which does not convey any message information. It is included to create a positive envelope which allows demodulation by a simple inexpensive envelope detector. From an information theory point of view, the power in the carrier component is wasted.

Definition of the DSBSC-AM Signal

Let $m(t)$ be a bandlimited baseband message signal with cutoff frequency W. The DSBSC-AM signal corresponding to $m(t)$ is

$$s(t) = A_c m(t) \cos \omega_c t$$

This is the same as AM except with the sinusoidal carrier component eliminated.

A message $m(t)$ typically has positive and negative values so it can not be recovered from $s(t)$ by an envelope detector. A demodulation method called *coherent demodulation* will be explored in this experiment.

Spectrum of a DSBSC-AM Signal

The Fourier transform of $s(t)$ is

$$S(\omega) = 0.5A_cM(\omega - \omega_c) + 0.5A_cM(\omega + \omega_c)$$

This is the same as the AM spectrum but with the discrete line at the carrier frequency removed. An example is shown in the figure on Slide 6-6.

- The carrier frequency must satisfy the bound, $\omega_c > W$ so that the two shifted baseband transforms do not overlap.
- When they overlap *foldover* is said to have occurred and perfect demodulation cannot be achieved.

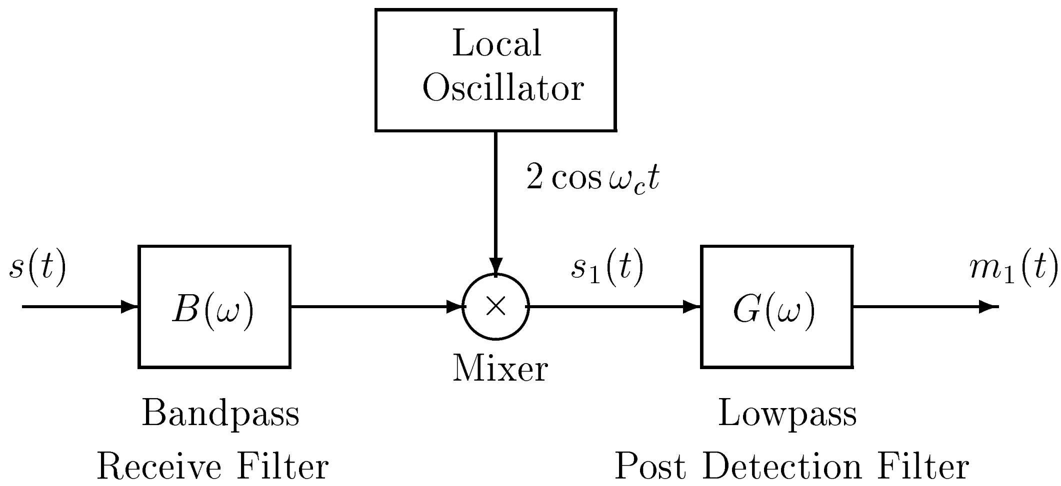
Why the Term Double-Sideband is Used

When $m(t)$ is a real signal, $M(-\omega) = \overline{M(\omega)}$ and

$$S(\omega_c - \omega) = \overline{S(\omega_c + \omega)} \quad \text{for} \quad 0 \leq \omega \leq \omega_c$$

- This equation shows that the component at frequency $\omega_c + \omega$ contains exactly the same information as the component at $\omega_c - \omega$ since one can be uniquely determined from the other by taking the complex conjugate.
- The portion of the spectrum for $|\omega| > \omega_c$ is called the *upper sideband* and the portion for $|\omega| < \omega_c$ is called the *lower sideband*.
- The fact that the modulated signal contains both portions of the spectrum explains why the term, double-sideband, is used.

The Ideal Coherent Receiver for DSBSC-AM



First, the received signal is passed through a bandpass filter $B(\omega)$ centered at the carrier frequency that passes the DSBSC signal and eliminates out-of-band noise.

The output of $B(\omega)$ is then multiplied by a replica of the carrier wave. This replica is generated by a device called the *local oscillator* (LO) in the receiver. The device that performs the product is often called a *product modulator* or *balanced mixer*.

Ideal Coherent Receiver Analysis

Assuming no noise, the product is

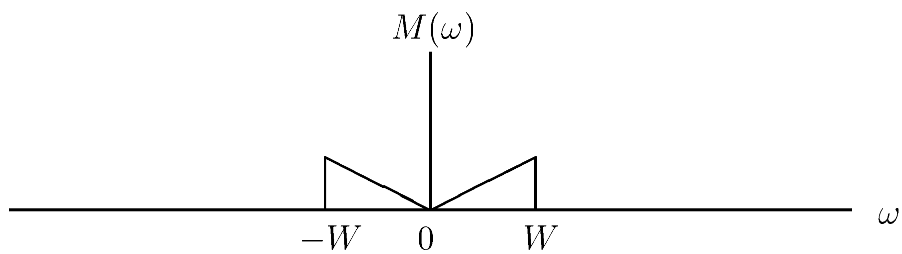
$$\begin{aligned} s_1(t) &= 2s(t)\cos\omega_c t = 2A_c m(t)\cos^2\omega_c t \\ &= A_c m(t) + A_c m(t)\cos 2\omega_c t \end{aligned}$$

The Fourier transform of the product modulator output is

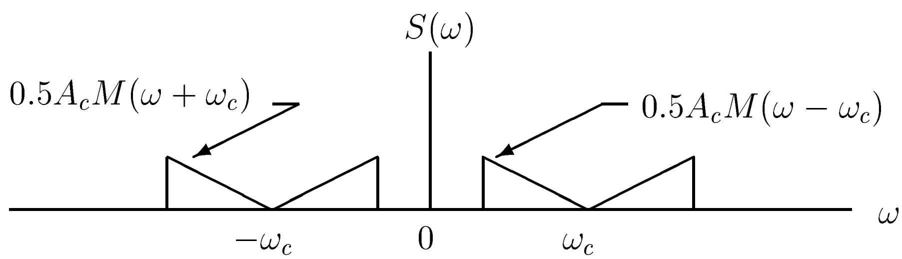
$$\begin{aligned} S_1(\omega) = \;& A_c M(\omega) + 0.5A_c M(\omega + 2\omega_c) \\ & + 0.5A_c M(\omega - 2\omega_c) \end{aligned}$$

and is illustrated in the figure on Slide 6-6. The first term on the right-hand side is proportional to the desired message. The second term has spectral components centered around $-2\omega_c$ and $2\omega_c$. The corresponding terms can be seen in $S_1(\omega)$. The undesired high frequency terms are eliminated by the final lowpass filter which has cutoff frequency W. This is often called a *post detection* filter.

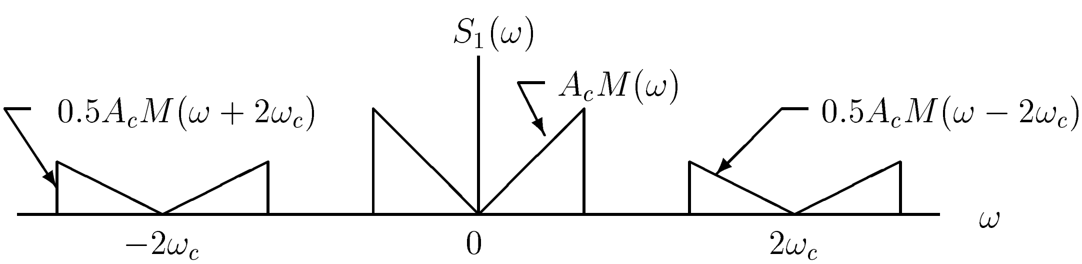
Spectra in DSBSC-AM Communication System



(a) Fourier Transform of Baseband Message



(b) Fourier Transform of DSBSC-AM Signal



(c) Fourier Transform of Mixer Output

A Demodulator Using the Pre-Envelope

An alternative method of demodulation is to first form the pre-envelope of the received signal. With no additive noise, this is

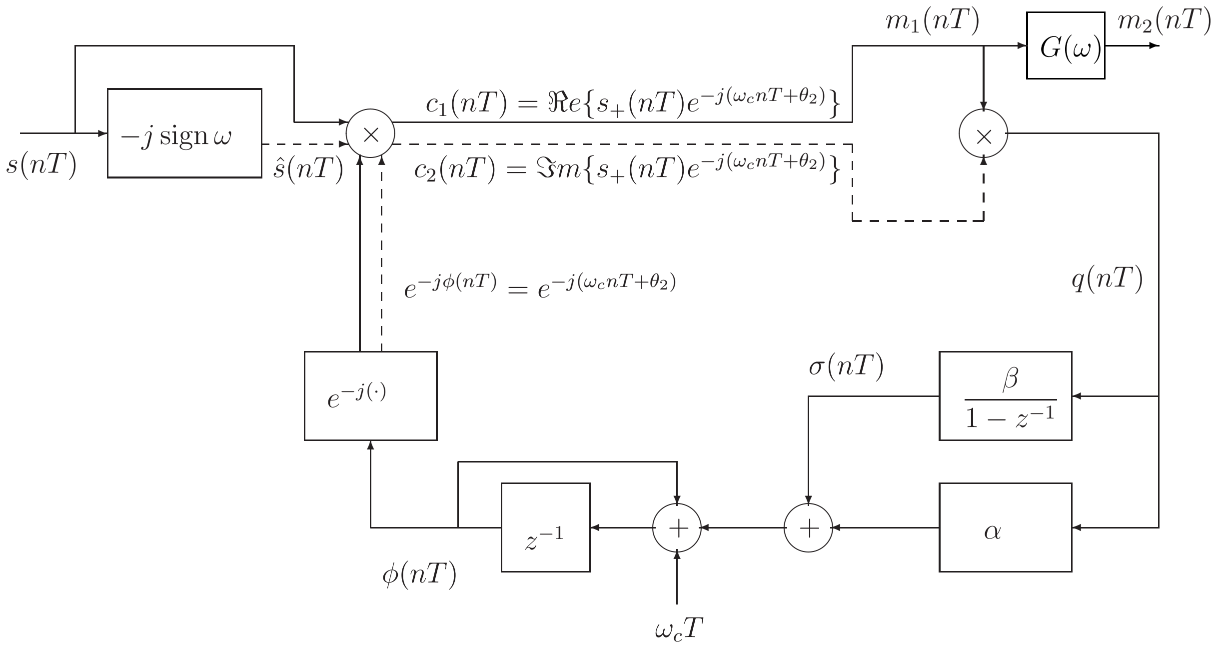
$$\begin{aligned} s_+(t) &= s(t) + j\hat{s}(t) \\ &= A_c m(t) \cos \omega_c t + j A_c m(t) \sin \omega_c t \\ &= A_c m(t) e^{j\omega_c t} \end{aligned}$$

The baseband message is then recovered to within a scale factor by forming the complex product

$$s_+(t) e^{-j\omega_c t} = A_c m(t)$$

The Costas Loop Demodulator

The Costas loop locks to the carrier frequency and phase and performs nearly ideal coherent demodulation.



Second-Order Costas Loop Demodulator

Let the received signal after it has passed through a bandpass receive filter be

$$s(nT) = A_c m(nT)\cos(\omega_c nT + \theta_1)$$

where ω_c is the nominal carrier frequency and θ_1 is a constant or slowly changing phase angle.

Costas Loop (cont. 1)

- The first step is to form the complex envelope

$$s_+(nT) = s(nT) + j\hat{s}(nT) = A_c m(nT) e^{j(\omega_c nT + \theta_1)}$$

The parallel solid and dotted lines in the figure represent complex signals with the solid line corresponding to the real part and dotted line to the imaginary part.

- The system generates an estimate $\phi(nT)$ of the angle of the received signal that can expressed as

$$\phi(nT) = \omega_c nT + \theta_2(nT)$$

It is passed through the complex exponential box to give the local oscillator signal $e^{-j\phi(nT)}$. The method for generating this angle will be explained shortly.

Costas Loop (cont. 2)

• The local oscillator signal is multiplied by the complex envelope resulting in the signal

$$c(nT) = s_+(nT)e^{-j\phi(nT)} = A_c m(nT)e^{j[\theta_1 - \theta_2(nT)]}$$

which is separated into its real part

$$\begin{aligned} c_1(nT) &= s(nT)\cos\phi(nT) + \hat{s}(nT)\sin\phi(nT) \\ &= A_c m(nT)\cos[\theta_1 - \theta_2(nT)] \end{aligned}$$

and imaginary part

$$\begin{aligned} c_2(nT) &= \hat{s}(nT)\cos\phi(nT) - s(nT)\sin\phi(nT) \\ &= A_c m(nT)\sin[\theta_1 - \theta_2(nT)] \end{aligned}$$

• The loop is *in lock* when the phase error $\theta_1 - \theta_2$ remains small. When $\theta_1 - \theta_2 \equiv 0$, the demodulated message appears at the point labeled $c_1(nT) = m_1(nT)$ and $c_2(nT) = 0$. The filter $G(\omega)$ is a lowpass filter that passes the message and eliminates out-of-band noise.

Costas Loop (cont. 3)

A lock detection strategy is to lowpass filter $c_2^2(nT)$ and declare that the loop is in lock when this signal falls below a threshold.

- The real and imaginary parts are multiplied, resulting in the signal

$$\begin{aligned} q(nT) &= c_1(nT)c_2(nT) \\ &= A_c^2 m^2(nT)\cos[\theta_1 - \theta_2(nT)]\sin[\theta_1 \\ &\qquad - \theta_2(nT)] \\ &= 0.5A_c^2 m^2(nT)\sin\{2[\theta_1 - \theta_2(nT)]\} \end{aligned}$$

Notice that when θ_1 and θ_2 differ by less than 90 degrees, $q(nT)$ has the same sign as the phase error $\theta_1 - \theta_2$, so it indicates in which direction the local phase estimate θ_2 should be changed to reduce the phase error to zero. When the loop is in lock, the small angle approximation $\sin x \simeq x$ can be used to accurately approximate $q(nT)$ by

$$q(nT) \simeq A_c^2 m^2(nT)[\theta_1 - \theta_2(nT)] \quad \text{for } |\theta_1 - \theta_2(nT)| \ll 1$$

Costas Loop (cont. 4)
Generating the Phase Estimate

The lower half of the block diagram generates the loop's estimate of the phase of the received signal by computing

$$\phi((n+1)T) = \phi(nT) + \omega_c T + \alpha q(nT) + \sigma(nT)$$

where

$$\sigma(nT) = \beta q(nT) + \sigma((n-1)T)$$

and α and β are small positive constants with $\beta < \alpha/50$, typically.

- The basic philosophy behind these equations is that at each new sampling instant the loop's phase estimate is incremented by the nominal change in carrier phase between samples $\omega_c T$ plus a small correction term $\alpha q(nT)$ roughly proportional to the phase error.

Generating the Phase Estimate (cont.)

- Notice that when $q(nT) = 0$ for all n, $\phi(nT)$ is the linear ramp

$$\phi(nT) = \omega_c nT + \phi(0)$$

which has a slope equal to the nominal carrier frequency.

- The accumulator block $\beta/(1 - z^{-1})$ is included to allow the loop to track a carrier input phase $\theta_1(nT)$ that is a linear ramp with zero steady-state error. The input phase has this form when there is a frequency offset between the received and local carrier frequencies. The output $\sigma(nT)$ of the accumulator reaches the steady-state value $\Delta\omega T$ which is the phase change between samples caused by the frequency offset $\Delta\omega$.

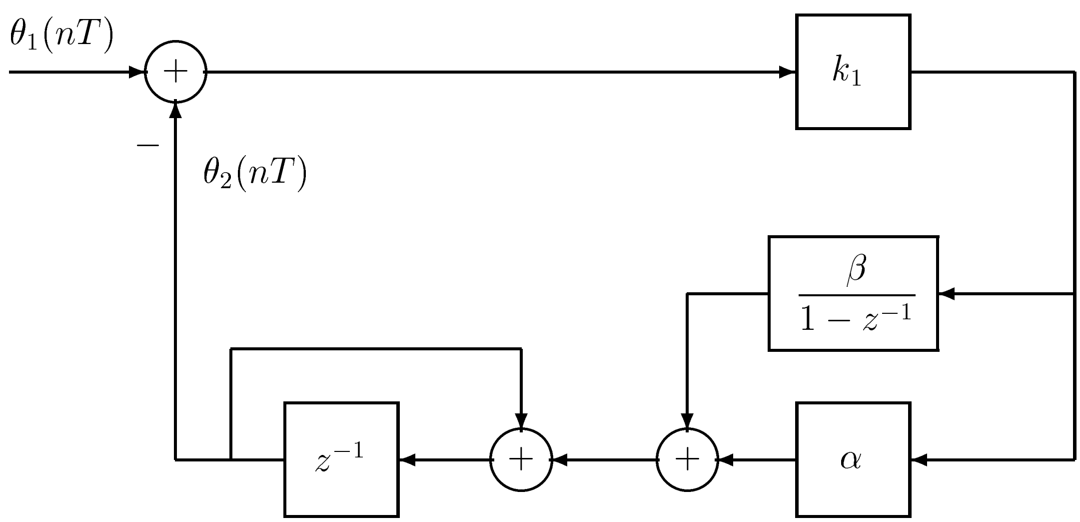
Linearized Loop Model

The Costas loop is a nonlinear and time-varying system. However, when it is in lock, it can be accurately approximated by a linear, time-invariant system by using the small angle approximations and replacing $m^2(nT)$ by its expected value. Let

$$k_1 = A_c^2\, E\{m^2(nT)\}$$

and further approximate $q(nT)$ by

$$q(nT) \simeq k_1[\theta_1 - \theta_2(nT)]$$



Linearized Loop Block Diagram

Linearized Loop Model (cont.)

The transfer function for the linearized loop is

$$
\begin{aligned}
H(z) &= \frac{\Theta_2(z)}{\Theta_1(z)} \\
&= \frac{k_1(\alpha+\beta)\left(1-\frac{\alpha}{\alpha+\beta}z^{-1}\right)z^{-1}}{1-[2-k_1(\alpha+\beta)]z^{-1}+(1-k_1\alpha)z^{-2}}
\end{aligned}
$$

The frequency response is obtained by letting $z = e^{j\omega T}$ and has the shape of a narrowband lowpass filter for small α and β. The closed loop gain at zero frequency is $H(1) = 1$.

Performance with Additive Noise

Suppose the received signal without noise is

$$s(t) = a(t)\cos(\omega_c t)$$

and corrupting band limited additive noise is

$$n(t) = n_I(t)\cos(\omega_c t) - n_Q(t)\sin(\omega_c t)$$

where $n_I(t)$ and $n_Q(t)$ are the independent lowpass inphase and quadrature noise components.

Performance with Additive Noise (cont. 1)

Then, the received signal is

$$\begin{aligned} r(t) &= s(t) + n(t) \\ &= [a(t) + n_I(t)]\cos(\omega_c t) - n_Q(t)\sin(\omega_c t) \end{aligned}$$

The Hilbert transform of the received signal is

$$\hat{r}(t) = [a(t) + n_I(t)]\sin(\omega_c t) + n_Q(t)\cos(\omega_c t)$$

and the pre-envelope is

$$\begin{aligned} r_+(t) &= r(t) + j\hat{r}(t) \\ &= [a(t) + n_I(t)][\cos(\omega_c t) + j\sin(\omega_c t)] \\ &\quad + jn_Q(t)[\cos(\omega_c t) + j\sin(\omega_c t)] \\ &= [a(t) + n_I(t) + jn_Q(t)]e^{j\omega_c t} \end{aligned}$$

The demodulator output is

$$c_1(t) = \mathcal{R}e\left\{r_+(t)e^{-j\omega_c t}\right\} = a(t) + n_I(t)$$

Notice that only the inphase noise component is present in the output. The square-law demodulator output contains both $n_I(t)$ and $n_Q(t)$.

Laboratory Exercises and Experiments for the Costas Loop

Use the signal generator to generate the AM wave

$$s(t) = A_c m(t) \cos 2\pi f_c t$$

where

$$A_c = 1$$
$$m(t) = 1 + 0.4 \cos(2\pi f_m t + \Phi)$$
$$f_c = 4000 \text{ Hz}$$
$$f_m = 400 \text{ Hz}$$

where Φ is a random variable uniform over $[0, 2\pi)$.

Actually, $s(t)$ is an AM signal with modulation index $\mu = 0.4$. However, it can also be considered to be a DSBSC-AM signal with $m(t)$ containing a dc value and all the theory for the Costas loop still holds.

Theoretical Design Exercises

In these exercises you will do theoretical computations to select the Costas loop parameters for a reasonable design. Do the following:

1. Compute k_1 by the equation on Slide 6-14.
2. Choose some small values for the loop filter constants, for example, $\alpha = 0.01$ and $\beta = 0.0002$. You will find that β should be small relative to α, perhaps, less than $\alpha/50$, to get a transient response without excess ripple. Recursively compute the response of the linearized loop to a unit step in $\theta_1(nT)$ using the following realization of $H(z)$:

$$\begin{aligned}\theta_2(nT) &= k_1[(\alpha+\beta)\theta_1((n-1)T) - \alpha\theta_1((n-2)T)] \\ &\quad + [2 - k_1(\alpha+\beta)]\theta_2((n-1)T) \\ &\quad - (1 - k_1\alpha)\theta_2((n-2)T)\end{aligned}$$

   Continue the computations until $\theta_2(nT)$ gets close to its final value and plot the result. Repeat this step for different values of α and β until you find a pair for which the step response settles to its final value in about 0.5 seconds.

3. Compute and plot the closed loop amplitude response $A(f) = 20\log_{10}|H(e^{j2\pi f/f_s})|$ for the values of α and β finally selected in the previous step.

Hardware Experiments

Write a C/assembly language program for the TMS320C6713 to do the following:

1. Initialize the DSP and codec as in Chapter 2.
2. Read samples from the ADC at a 16 kHz sampling rate.
3. Demodulate the input signal with a Costas loop. Make sure to keep the LO angle confined to $0 \leq \phi(nT) \leq 2\pi$.
4. Send $c_1(nT) = m_1(nT)$ to the left AIC output channel and $c_2(nT)$ to the right channel. What should $c_2(nT)$ be when the loop is in lock?

Hardware Experiments (cont. 1)

Perform the following exercises:

1. Connect the signal generator to the DSK line input and set it to generate the AM signal $s(t)$ defined above. Connect the DSK line output to the oscilloscope and debug your DSP program if the output is not $m(t)$.

2. Investigate your Costas loop performance in the presence of a frequency offset.
   - First set the signal generator carrier frequency to the nominal 4 kHz value and let your loop achieve lock.
   - Then slowly change the carrier frequency to a slightly different value and see if the loop remains locked.
   - Use Code Composer to watch $\sigma(nT)$ in your DSP program and check that it has the correct steady-state value for the frequency offset you are using.

Hardware Experiments (cont. 2)

Creating a Watch Window

To watch a variable, right click on the variable and then left click on "Add Watch Expression..." Enter the variable name in the box of the "Add Watch Expression" window. You can move and resize the watch window as you prefer.

3. The linearized equations describe the loop behavior accurately when it is in lock. When there is a large initial frequency offset, the behavior is quite different.

   • Experimentally investigate this behavior by setting the signal generator carrier frequency to a value that differs by 30 Hz or more from the nominal 4 kHz value and starting the Costas loop. The loop may take a much longer time than you expected to achieve lock and may not even lock. This is called its *pull in* behavior.

Hardware Experiments (cont. 3)

4. Design and implement the lowpass filter $G(\omega)$ as an IIR filter. It should have a cutoff frequency such that it passes the baseband message band and eliminates noise above the message band.

5. Experiment demodulating signals corrupted by additive, zero mean, Gaussian noise. Observe the behavior as the SNR is decreased and make a rough estimate of the SNR at which the demodulator no longer works. It should work at lower SNR's than the envelope detectors. You should "gear shift" to much smaller values of α and β once the loop is in lock to reduce its bandwidth and response to noise.

6. Set up an array and write the first few thousand samples of $q(nT)$ to the array. Do not add noise to the input samples. Then send the array to the PC and plot the

resulting signal to get a clearer picture of the loop transient response. You can use the C function `fprintf()` to write the samples to a file on the PC. If time permits, do this with other signals in the loop.