Introduction to Communication Engineering
SSY121, Lecture # 7

Fredrik Brännström
Fredrik.Brannstrom@chalmers.se

Communication Systems Group
Department of Electrical Engineering
Chalmers University of Technology
Göteborg, Sweden

September 15, 2021

Outline

1. Constant-Envelope Modulation
   - Constant-Envelope Modulation
   - Nonlinear Amplifier
   - FSK and CPFSK

2. Carrier, Phase, Symbol, and Frame Synchronization
   - 2D Passband Tx and Rx
   - Things that can destroy the design
   - How do synchronizers work?

Part I

Constant-Envelope Modulation

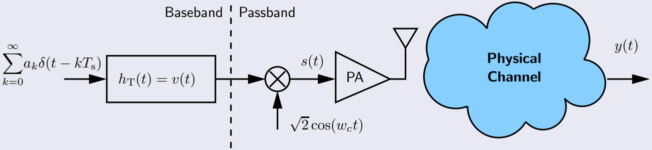
Constant-Envelope Modulation

- Consider the class of signals alternatives of the form

$$s_i(t) = \cos\left(w_c t + \psi_i(t)\right)$$

- The messages are sent in the phase of the signal $\psi_i(t)$, $i = 1, 2, \ldots, M$ with $0 \le t \le T_\mathrm{s}$.
- The envelope of the signal is constant
- Why are constant-envelope signals good?

1D Tx

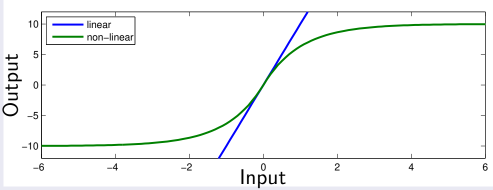


Nonlinear Amplifier

- Class A amplifiers are linear, but have an efficiency 20–30%
- Class C amplifiers have an efficiency of 90%, but are strongly nonlinear
- Nonlinearity leads to spectral spreading
- More details? See Sec. 3.8.2 in [Anderson]

Nonlinearity

$$g(x) = \left(1 + e^{-|x|}\right) \mathsf{sign}(x)$$



Constant-envelope modulations

- OOK, PAM, and QAM are never constant-envelope modulations
- QPSK (or M-PSK) is constant-envelope if square pulses are used, but it is not if for example RRC pulses are used

QPSK and offset QPSK (from [Anderson])

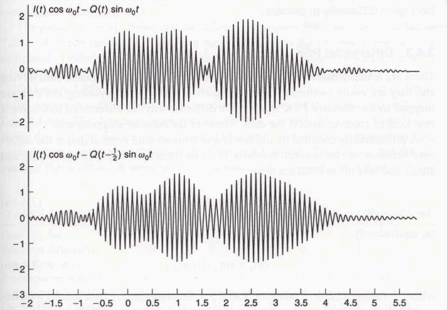


Figure 3.12 Comparison of QPSK **(top)** and offset QPSK **(bottom)** with same data and 30% root RC pulses. Data as in Figs. 3.1, 3.2, and 3.11.

How to visualize this?

It is possible to make a "continuous constellation plot" (I/Q plot in [Anderson]) which shows the amplitude of the envelope.

I/Q plots (from [Anderson])

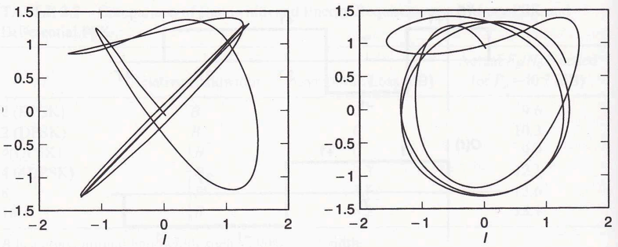


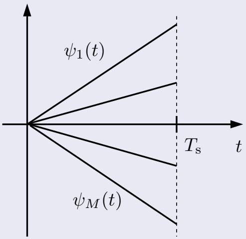
Figure 3.13 The *I*/*Q* plots for QPSK **(left)** and offset QPSK **(right)** for *I* data (+−+−+−−++−) and *Q* data (+−+−+++−−+). There are five 180° phase changes in the left plot.

Frequency Shift Keying

- In M-FSK, $\psi_i(t) = h\pi \frac{t}{T_\mathrm{s}} i$ for $i = \pm 1, \pm 3, \ldots, \pm(M-1)$, and therefore, the signals alternatives are

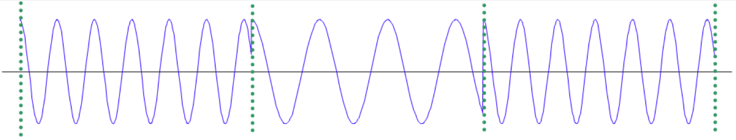
$$s_i(t) = \cos\left(2\pi \left[f_c + \frac{h}{2T_\mathrm{s}} i\right] t\right),$$

  where h is a constant (the *modulation index*).
- h determines the separation between the signal frequencies



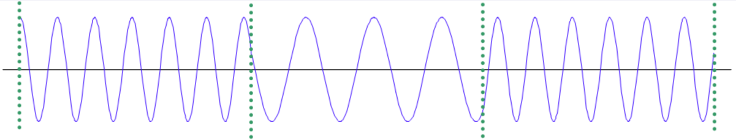
Frequency Shift Keying

- An FSK signal in time...



  - Stepwise signal changes (phase is not continuous)
  - This will produce a wide spectrum

- Solution? Make sure that the phase is continuous $\Rightarrow$ "Continuous phase FSK" (CPFSK)



  - Lower BW than FSK
  - More complex receiver

Part II

Carrier, Phase, Symbol, and Frame Synchronization

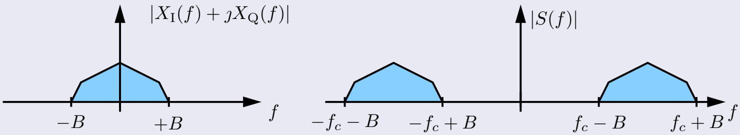
2D Passband Tx

- Transmitted signal (time domain):

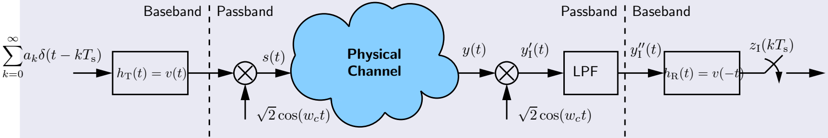
$$s(t) = x_{\mathrm{I}}(t)\cos(2\pi f_c t) + x_{\mathrm{Q}}(t)\sin(2\pi f_c t)$$

- Transmitted signal (frequency domain):

$$\begin{aligned}
S(f) &= \mathcal{F}\{x_{\mathrm{I}}(t)\cos(2\pi f_c t)\} + \mathcal{F}\{x_{\mathrm{Q}}(t)\sin(2\pi f_c t)\}\\
&= \mathcal{F}\{x_{\mathrm{I}}(t)\} * \mathcal{F}\{\cos(2\pi f_c t)\} + \mathcal{F}\{x_{\mathrm{Q}}(t)\} * \mathcal{F}\{\sin(2\pi f_c t)\}\\
&= X_{\mathrm{I}}(f) * \frac{1}{2}\Big[\delta(f+f_c)+\delta(f-f_c)\Big] + X_{\mathrm{Q}}(f) * \frac{\jmath}{2}\Big[\delta(f+f_c)-\delta(f-f_c)\Big]\\
&= \frac{1}{2}\Big[X_{\mathrm{I}}(f+f_c)+X_{\mathrm{I}}(f-f_c) + \jmath X_{\mathrm{Q}}(f+f_c) - \jmath X_{\mathrm{Q}}(f-f_c)\Big]
\end{aligned}$$



2D Passband Rx



If the channel is good, we can assume $y(t) \approx s(t)$, and thus

$$y_I'(t) \approx \left[\sqrt{2}\sum_{k=0}^{\infty} a_k v(t - kT_s)\cos(w_c t) + b_k v(t - kT_s)\sin(w_c t)\right]\sqrt{2}\cos(w_c t)$$

$$= \sum_{k=0}^{\infty} a_k v(t - kT_s) + \sum_{k=0}^{\infty} v(t - kT_s)(a_k \cos(2w_c t) + b_k \sin(2w_c t))$$

$$y_Q'(t) \approx \left[\sqrt{2}\sum_{k=0}^{\infty} a_k v(t - kT_s)\cos(w_c t) + b_k v(t - kT_s)\sin(w_c t)\right]\sqrt{2}\sin(w_c t)$$

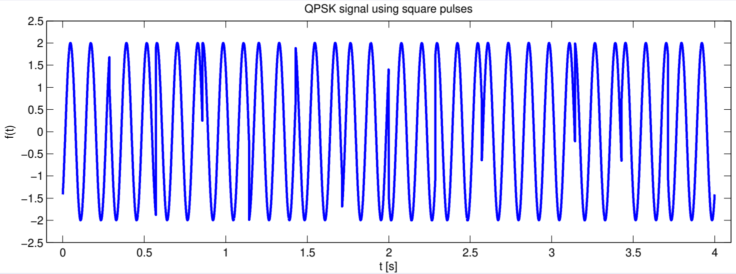
$$= \sum_{k=0}^{\infty} b_k v(t - kT_s) + \sum_{k=0}^{\infty} v(t - kT_s)(a_k \sin(2w_c t) - b_k \sin(2w_c t))$$

The LPF removes the red terms at frequency $2w_c$!

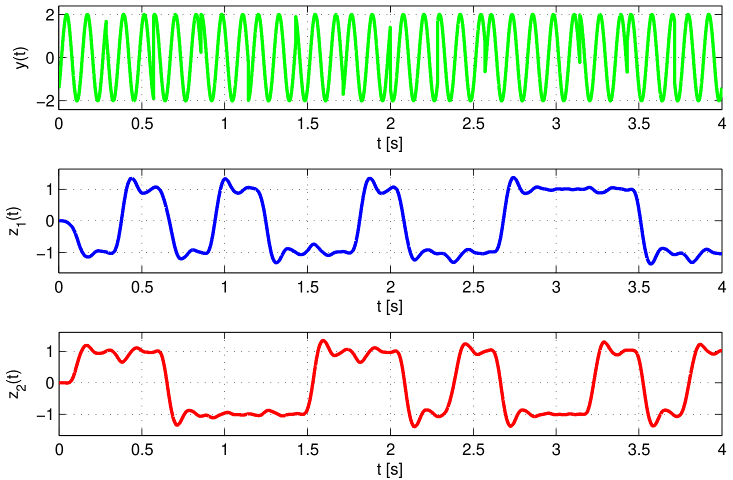
It seems that everything is quite easy...

- The MFR works nicely even for bad channels
- The implementation of the MFR can be done with relatively low complexity
- If everything is so easy...What is the problem?

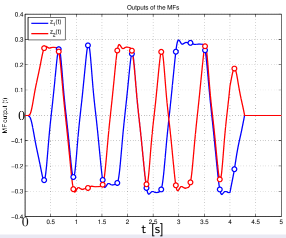
QPSK signal at the Tx

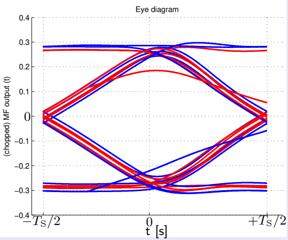


Signals at the Rx

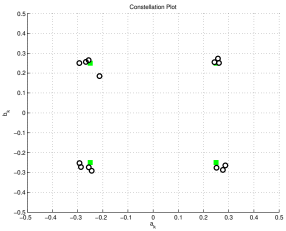


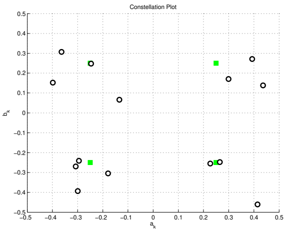
MF's outputs and Eye Diagram





Constellation Plot (Good and Bad Channel)





Things that can destroy the design

- The local oscillators in Rx is not exactly f_c but $f_c + \Delta$
- The phase of the reference signals in Rx is different than in Tx, i.e., $\cos(w_c t)$ and $\sin(w_c t)$ are $\cos(w_c t + \theta)$ and $\sin(w_c t + \theta)$
- The output of the MFs are taken at the wrong instant:

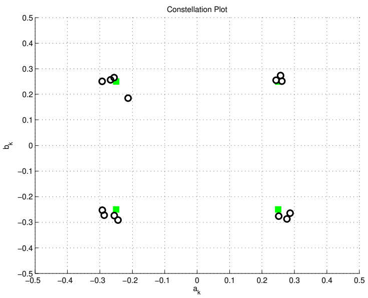
$$\begin{aligned} t &= 0.0T_\mathrm{s}, 1.1T_\mathrm{s}, 2.2T_\mathrm{s}, \ldots && \text{wrong sampling frequency} \\ t &= 0.1T_\mathrm{s}, 1.1T_\mathrm{s}, 2.1T_\mathrm{s}, \ldots && \text{wrong timing} \end{aligned}$$

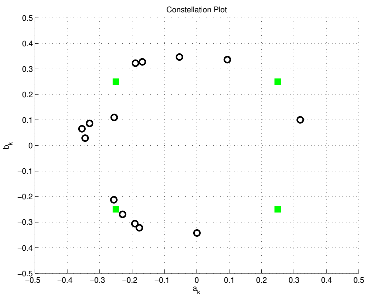
- It is not clear when the block starts: even if we have the correct bits, we need to know the beginning of the "sentence"

Each of the previous "problems" have a "solution"

- Carrier synchronization: Find the carrier frequency
- Phase synchronization: Find the phase of the carrier
- Symbol synchronization: Find the correct sampling instants
- Frame synchronization: Find the start of the block

Constellation Plot for QPSK, good channel, and $\Delta = 0.1 f_c$



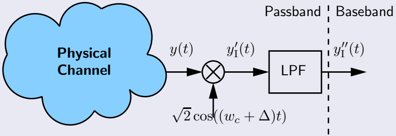


Wrong frequency $\Rightarrow$ a “rotating” constellation

Frequency mismatch

- The local oscillators are defined by its f_c plus some error
- It can be corrected by changing the frequency of the local oscillator
- A rotating constellation will cause many errors

Rx with frequency offset is a rotating constellation



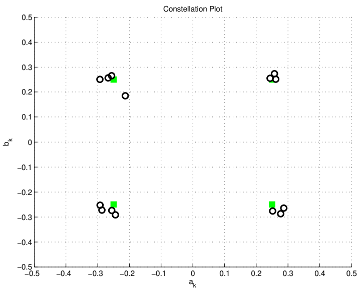
Why?

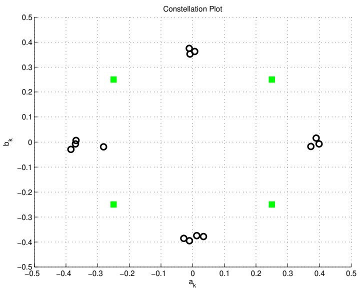
Using some trigonometry manipulation:

$$
\begin{aligned}
y(t) &= \sqrt{2}[a_0 v(t)\cos(w_c t) + b_0 v(t)\sin(w_c t)] \\
y_{\mathrm{I}}'(t) &= 2[a_0 v(t)\cos(w_c t) + b_0 v(t)\sin(w_c t)]\cos((w_c+\Delta)t) \\
&= a_0 v(t)[\cos((2w_c+\Delta)t) + \cos(\Delta t)] \\
&\quad + b_0 v(t)[\sin((2w_c+\Delta)t) - \sin(\Delta t)] \\
y_{\mathrm{I}}''(t) &= v(t)[a_0\cos\Delta t - b_0\sin\Delta t] \\
y_{\mathrm{Q}}'(t) &= a_0 v(t)[\sin((2w_c+\Delta)t) + \sin(\Delta t)] \\
&\quad + b_0 v(t)[\cos(\Delta t) - \cos((2w_c+\Delta)t)] \\
y_{\mathrm{Q}}''(t) &= v(t)[a_0\sin\Delta t + b_0\cos\Delta t]
\end{aligned}
$$

MATLAB demo!

Constellation Plot for QPSK, good channel, and $\theta = \pi/4$



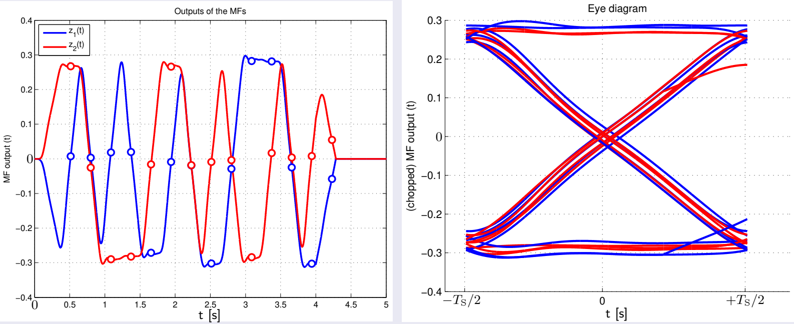


Wrong phase $\Rightarrow$ a "rotated" constellation

Coherent vs. Non-Coherent

- An Rx that needs the phase θ for detection is called a *coherent* Rx
- If θ is needed, a phase synchronizer must be implemented
- *Noncoherent* receivers can be implemented (e.g., FSK/OOK)

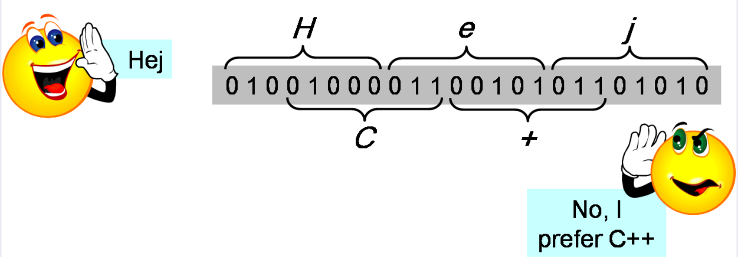
When the sampling instant is incorrect... $t_k = kT_s + 0.5T_s$



Wrong symbol synchronization

- The samples are very close to each other, i.e., it is hard to distinguish the signal alternatives
- It can be corrected by adding an appropriate delay in the sampling instant

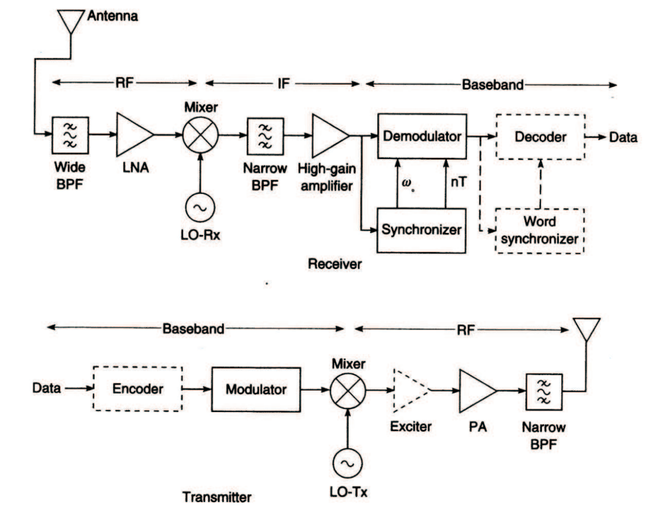
When the frame synchronization is incorrect...



In the project?

Have you notice any of the previous problems in the project yet?

A Simplified Communication System [Anderson p. 10]



How do synchronizers work?

- The synchronizer must estimate some parameter (frequency, phase, symbol timing, frame timing, etc.)
- The only information available is the received signal
- The synchronizer may look at
  - Maximum correlation to the squared (or Mth power) of the received signal (See Sec. 4.2.1 of [Anderson])
  - Early-late timing synchronizer (See Sec. 4.7.1 of [Anderson])
  - Maximum eye opening in the eye diagram
  - A *pilot tone*: carrier or frequency reference
  - A *marker*: known sequence inserted before the data
  - What is bad about the last two options?

What to do first?

1. Frequency and phase recovery must be done first
2. Symbol synchronization is the next step (after MF)
3. The last step is to find frame synchronization

What about a different order?

Sure, it is possible, but it is more complex!

Block-based vs. sequential

- A **block-based synchronization algorithm** registers a block of the signal, then optimizes parameters for the whole block at the same time
  - Numerical optimization
  - Delayed detection
- A **sequential synchronization algorithm** adapts the parameters continuously, allowing a small correction at each step
  - Real-time detection
  - Feedback loop
  - Read about the phase-locked loop (PLL) in the book Sec. 4.2

Need more information about synchronization for the project?

Check Chapter 4 in the course book, even though not all pages are listed in the Course Memo, Appendix B: Lecture Plan!