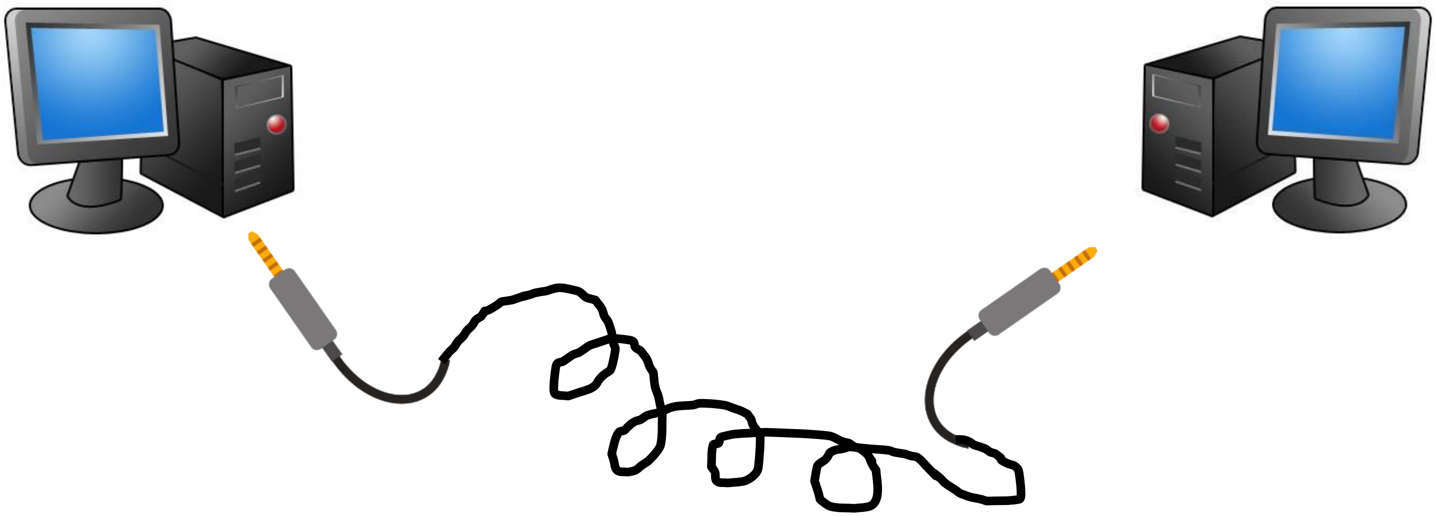
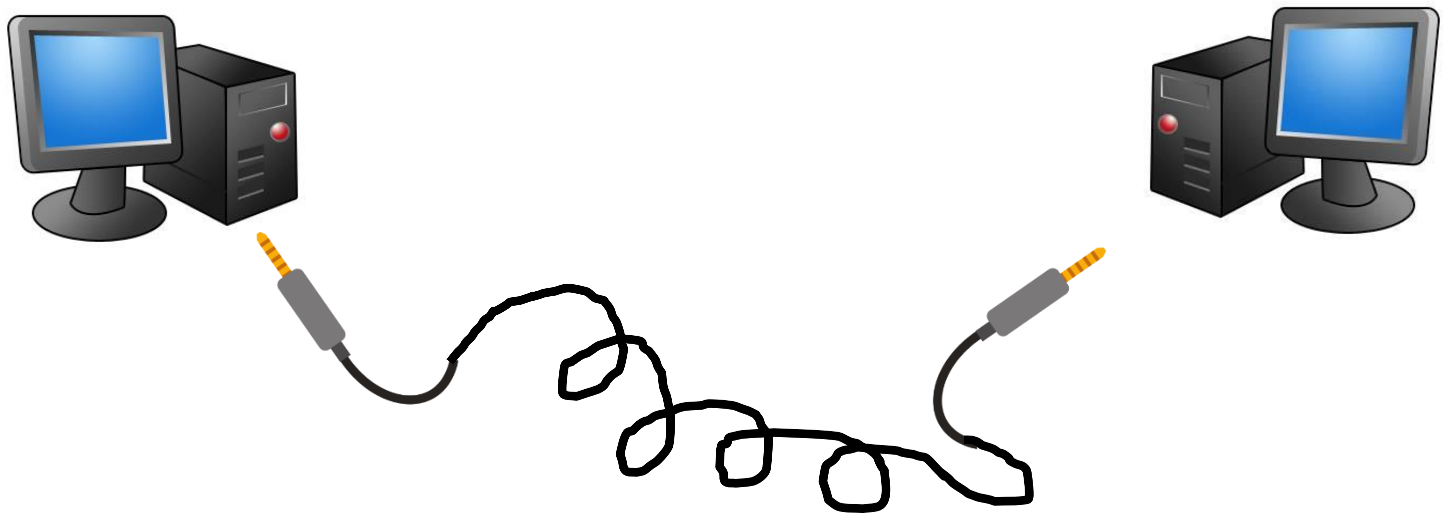
LAB #3.1 – DIGITAL TRANSMISSION

A WIRED CHANNEL

The cable between PC-audio cards



REAL-TIME OR OFF-LINE PROCESSING?



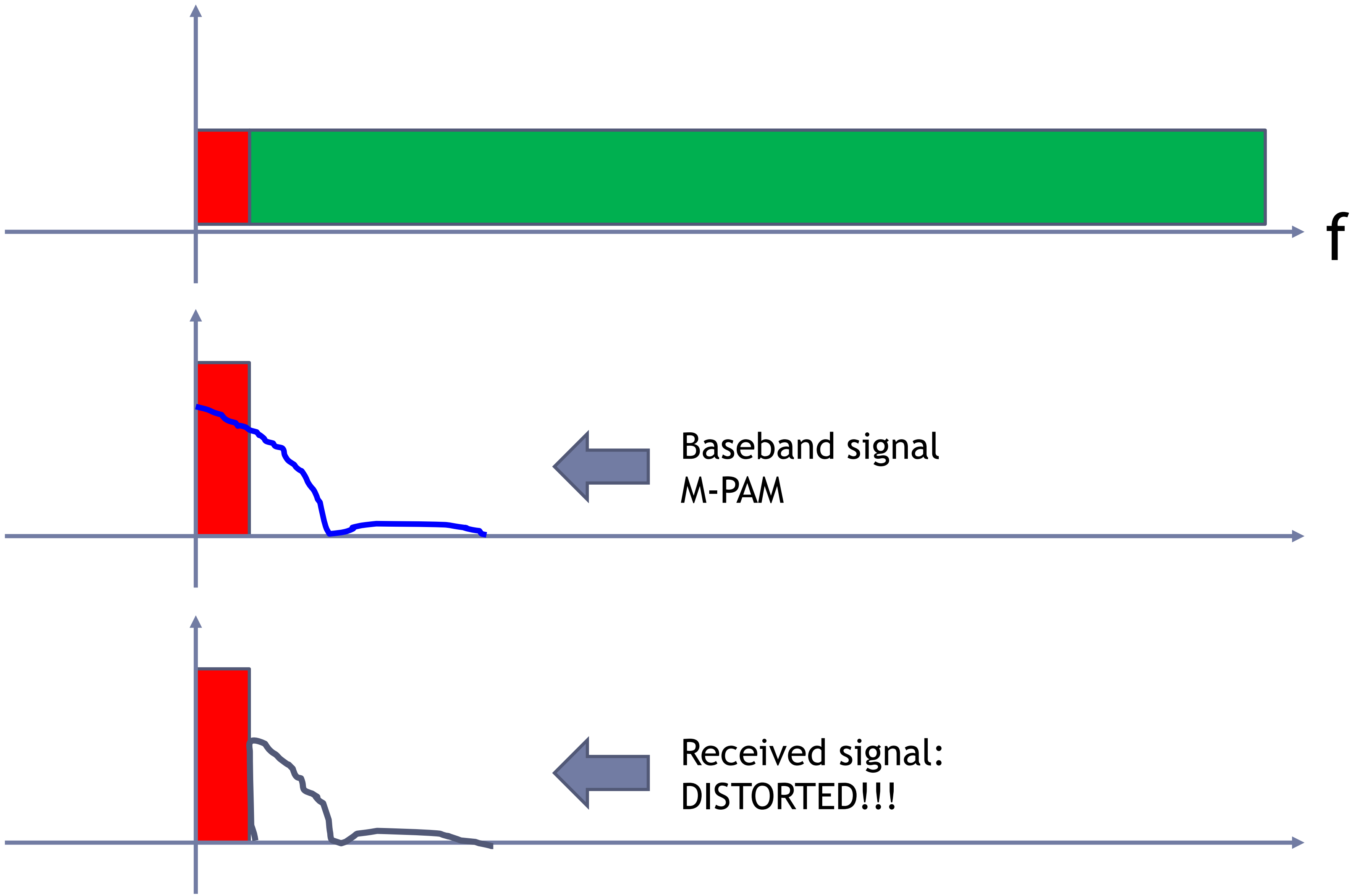
- Achieving a real-time digital transmission between two PCs using Matlab and transmitting over an audio cable is not an easy task
- We use off-line processing
- One PC generate and transmit a signal
- A second PC receive the signal and store it
- Then, it will recover the information in a deferred time

AVAILABLE BANDWIDTH

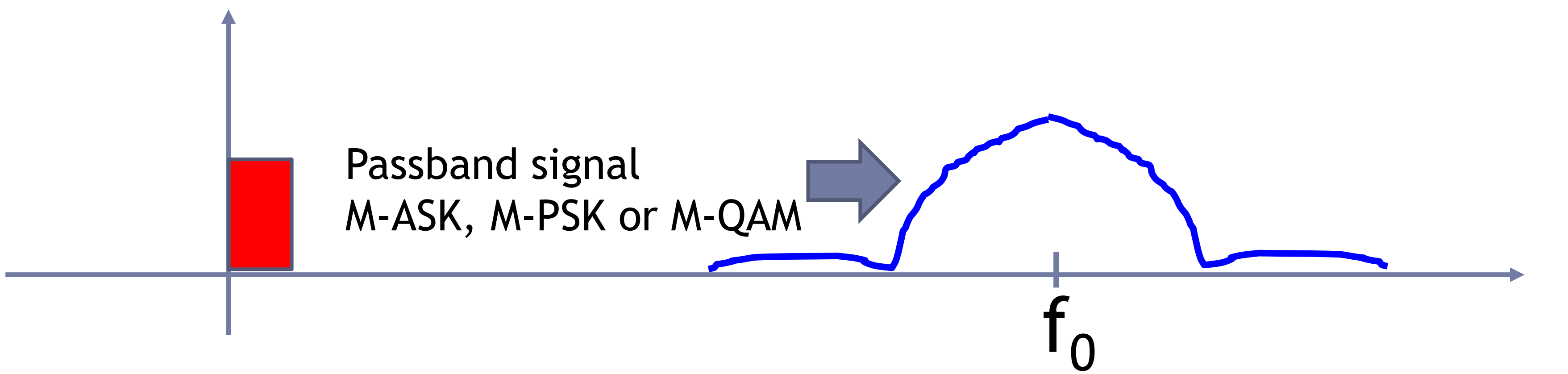
The cable itself has a pass-band from DC (0 Hz) to 20000 Hz

But TX (and RX), i.e. audio cards can generate signal from about 20 Hz to 20000 Hz

Baseband or Passband signal?



Baseband or Passband signal?



BLOCK DIAGRAM - TX

Data source → **Baseband Digital TX** → **Modulation** →

Data source

- Analog signal sampled, quantized and encoded (PCM)
- Image
- Text
- Your data

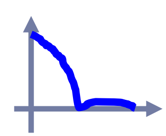
VECTOR OF BITS

BIT STREAM

Baseband Digital TX

- Select modulation
- Select pulse shape

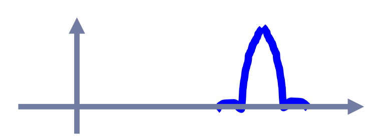
BASEBAND SIGNAL (SAMPLED)



Modulation

- Select carrier frequency

PASSBAND SIGNAL (SAMPLED)



DATA SOURCE

Select at least TWO of the following data sources to test your software:

- PCM: use the code from LAB1 to generate a data vector from an audio signal
- Text file
  - Reading a file

```
FID=fopen('filename.txt');
while (~feof(FID))
    riga = fgetl(FID);
end
fclose(FID);
```

  - Convert to ASCII

```
ASCII_codes=double(riga)   →   riga=char(ASCII_codes)
```

  - Convert to bits
- Image
  - Reading a file

```
[data,map]=imread('polito.png');   →   imshow(data,map);
```

  - Convert to bits

BASEBAND DIGITAL TRANSMITTER

- Use the code from LAB2 to generate a baseband signal carrying digital data
- System parameters
  - Bit-rate:
    - 100 bit/s … 1000 bit/s … 2000 bit/s
    - Start with a low bit-rate and the increase as much as you can
  - Modulation format
    - Start with a 2-PAM as in LAB2, it is the simplest way transmit
    - You can then try higher order M-PAM: first test it in baseband!!!
    - Optionally you can also try bidimensional M-QAM modulations
  - Pulse shape: NRZ or RZ (select the Duty Cycle)

COMMENTS

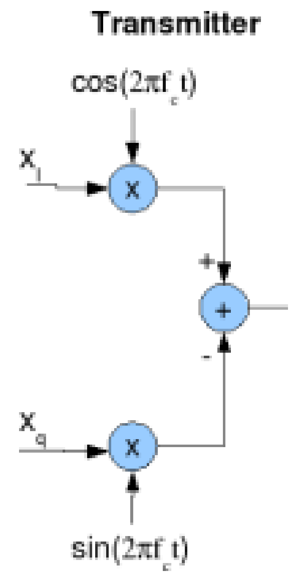
- Consider the dimension of your data vector to be transmitted
- Given the selected bit-rate, how long it will last the audio file?
- Be careful!

MODULATION

- Here is intended as frequency shift to f_0
  - Properly select you carrier frequency to fit in the available bandwidth

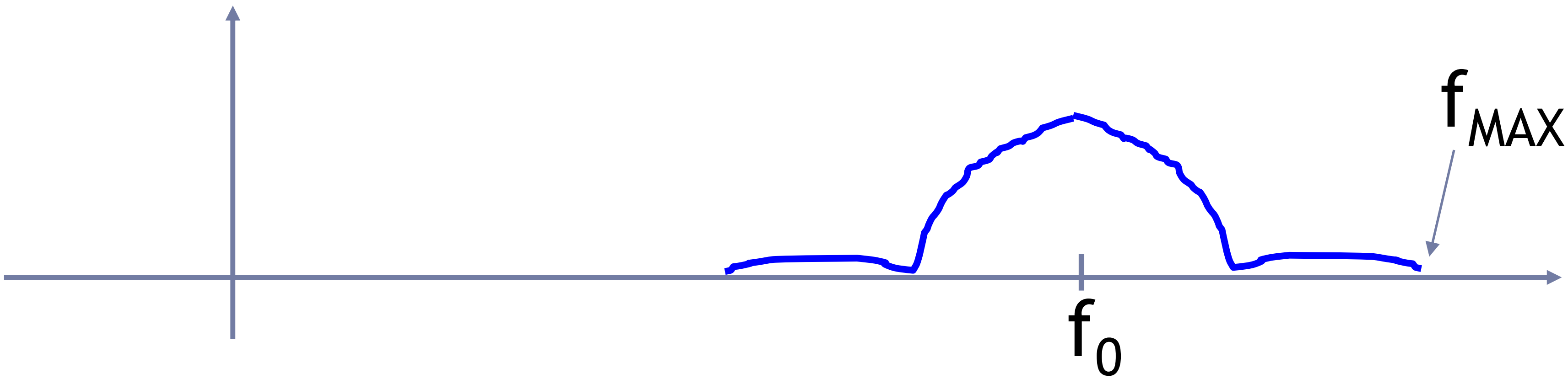
$$x(t) = x_I(t)\cos(2\pi f_0 t) - x_Q(t)\sin(2\pi f_0 t)$$

- For mono-dimensional modulations, only $x_I(t)$ is used: $x_Q(t)=0$
- $x_I(t)$ is the M-PAM baseband signal



MODULATION

- Properly select the sampling frequency satisfying the sampling theorem
  - Consider also that sampling frequency must be compatible with your audio card



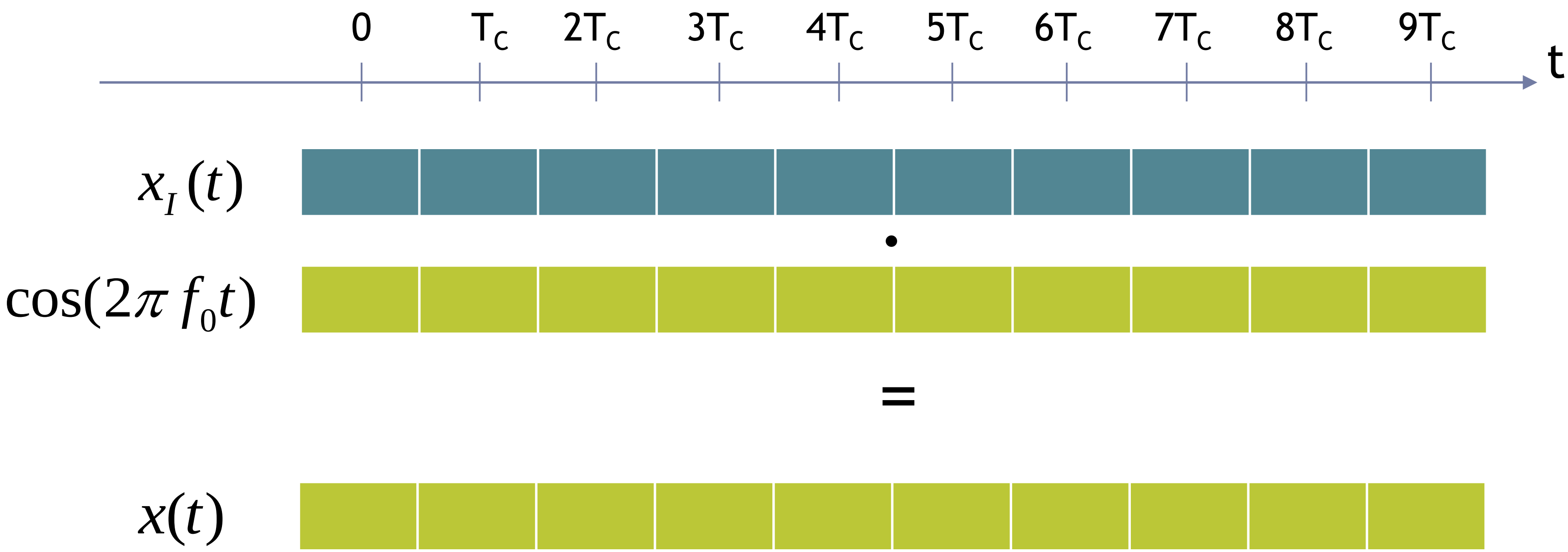
- This impact on the baseband digital transmitter
  - In LAB2 it has been shown that the number of samples (SpS) per symbol depends on sampling frequency, that is equivalent to the simulation bandwidth (B_{SIM})
  - As SpS MUST be an integer, you have some limitation in the choice of the bit-rate and /or the sampling frequency

Here $B_{SIM} = f_C$

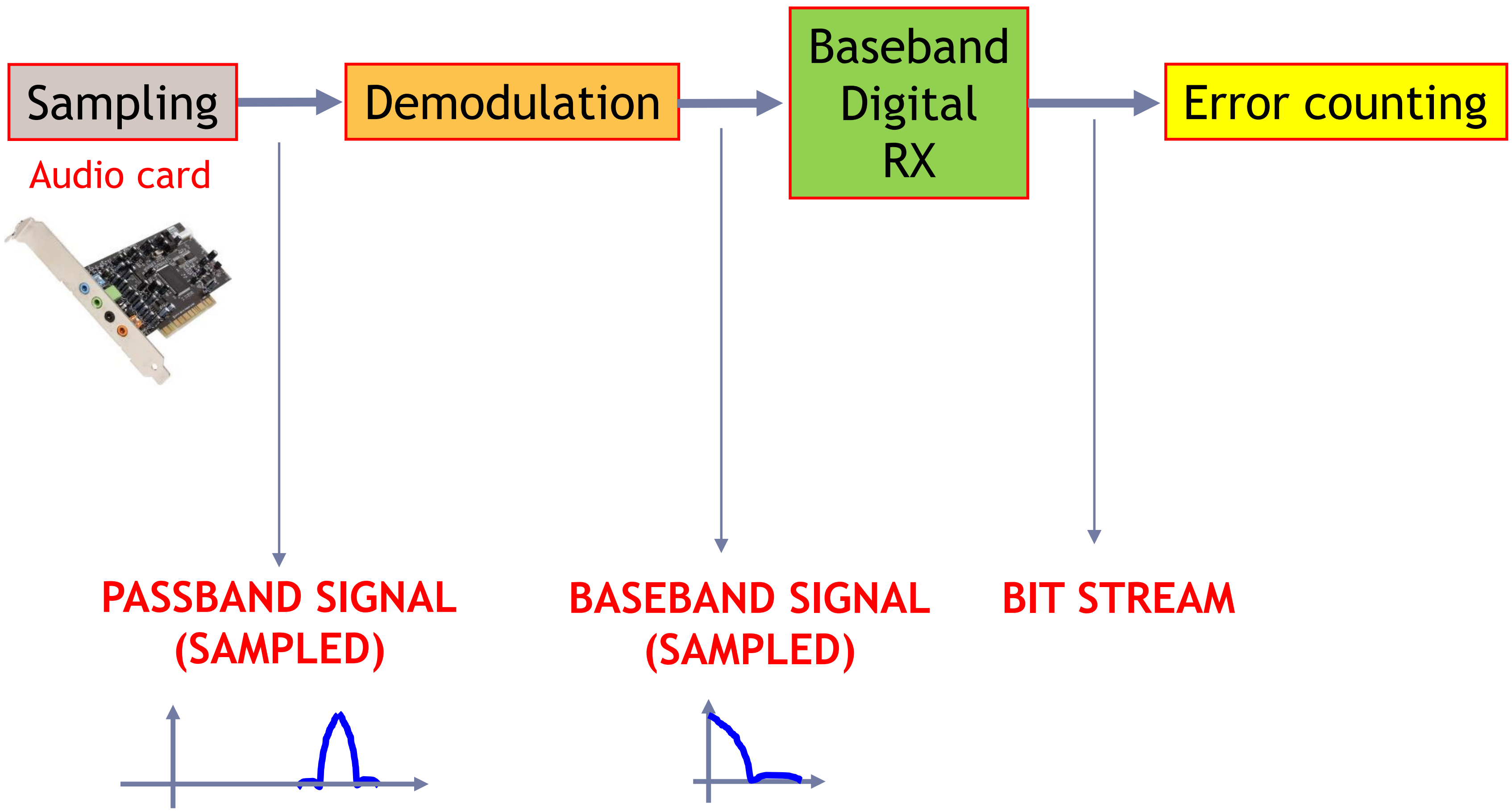
$$B_{SIM} = SpS \cdot R_S$$

MODULATION

- Applying the modulation to the baseband signal $x(t) = x_I(t)\cos(2\pi f_0 t)$



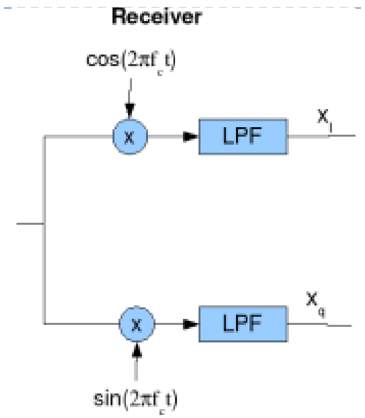
BLOCK DIAGRAM - RX



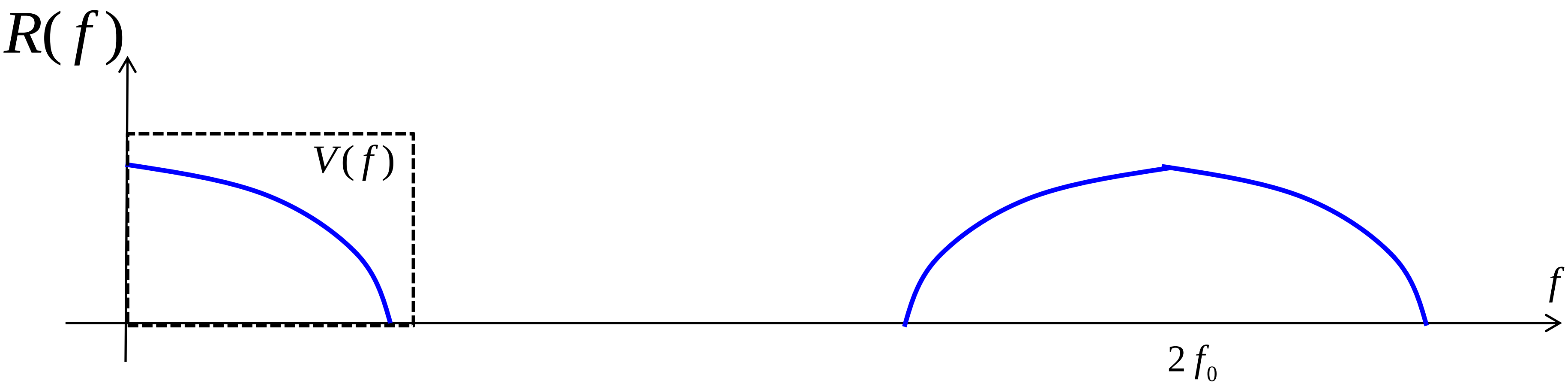
SAMPLING

- The audio card in the receiver PC acts as a sampler
  - Theoretically you can have this sampling frequency different from the one used at transmitter side: it does not modify the signal if the sampling theorem is still verified
  - To keep it simple, used the same sampling frequency imposed at transmitter side

DEMODULATION



- The demodulator consist in a multiplication with a carrier having the same f_0
  - f_0 is known and stable: we do not have to recover it from the signal



- After beating with a “local oscillator” a second harmonic signals is generated
  - It must be filtered out by a low pass filter: it is already present in the Baseband Digital Receiver
    - Matched Filter or other choice

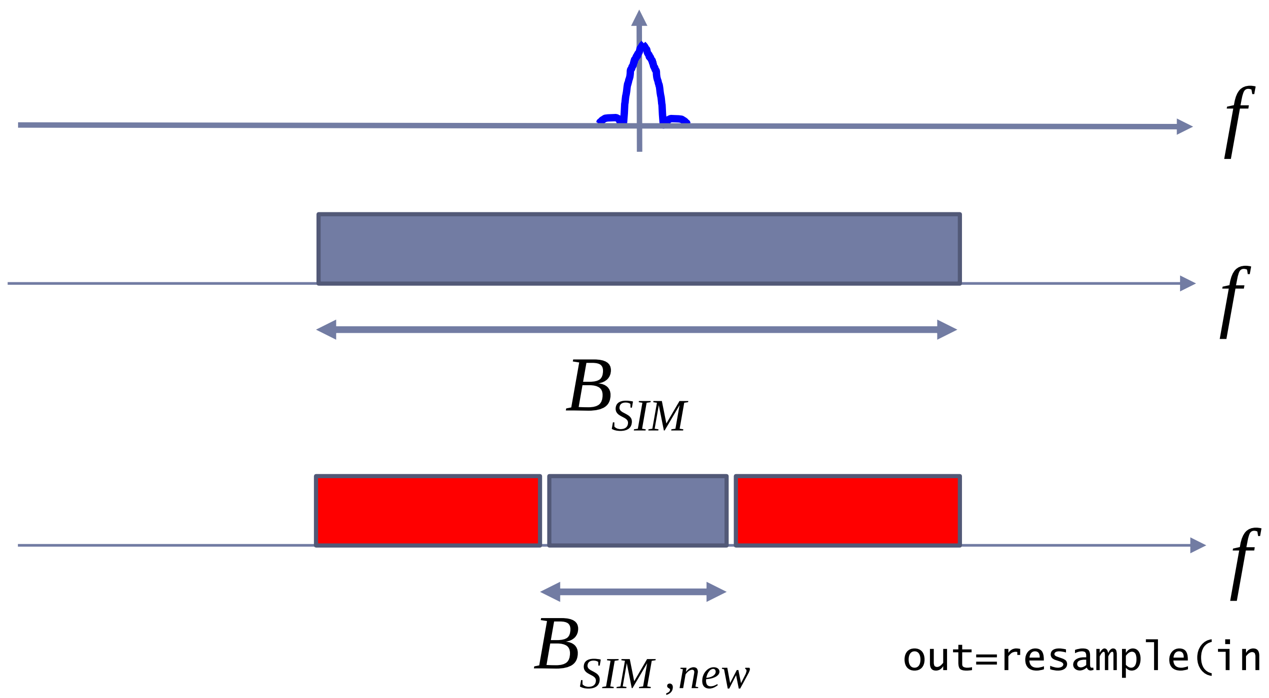
BASEBAND DIGITAL RECEIVER

- Use the code from LAB2 for then digital baseband receiver
  - Filter
  - Sampling (with selection of the optimum sampling instant t_{opt})
  - Decision
  - Error Counting
- This baseband signal comes from a passband signal downshifted
  - The number of SpS at transmitter was increasing to allow the passband modulation: Oversampling
  - Now it is not need and can impact processing time and memory
  - We can down-sample to a smaller number of SpS

DOWNSAMPLING

- It means reducing the number of SpS, but…

$$B_{SIM} = SpS \cdot R_S$$



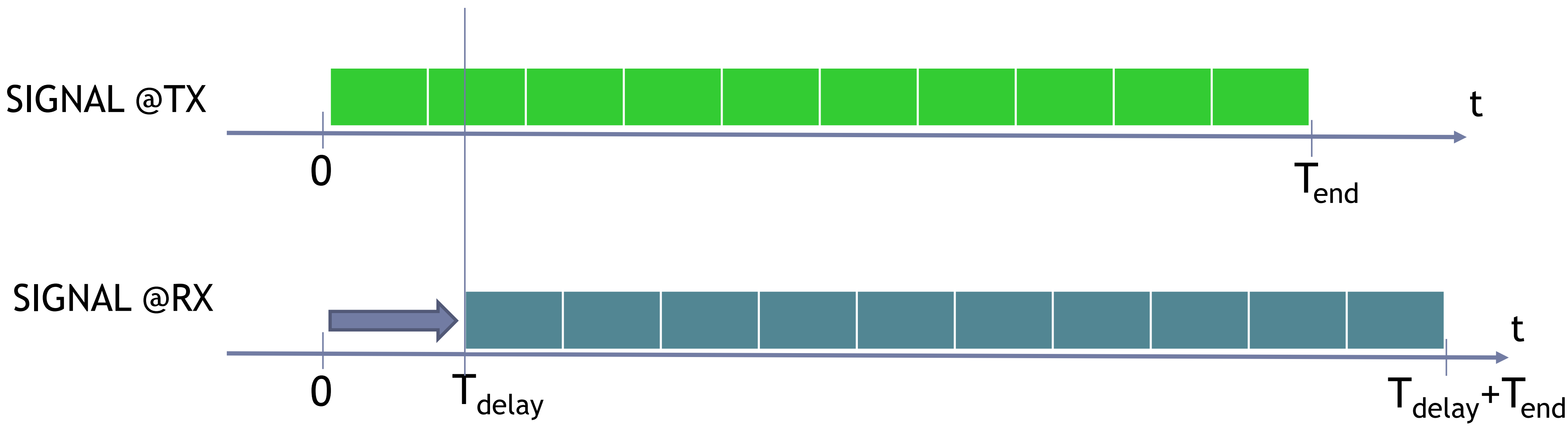

```
out=resample(in,SpS_new,SpS);
```

or

```
out=resample(in,1,Q);
```

SYNCHRONIZATION

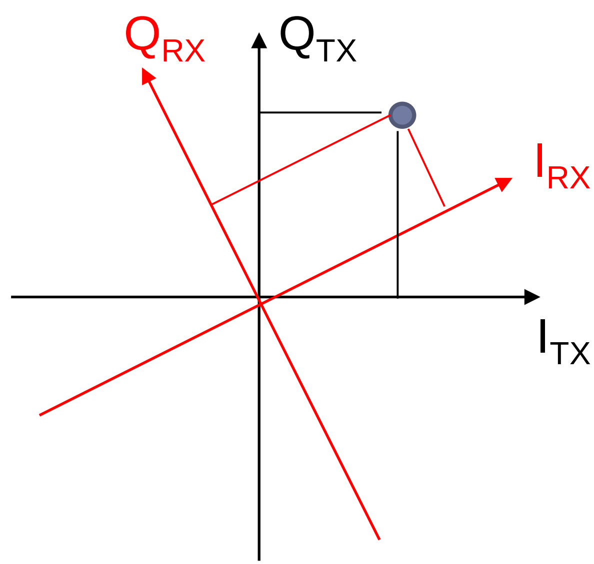
- When you record the audio file you introduce a delay in the received signal




```
[signal_tx_aligned,signal_rx_aligned,T_Delay] = alignsignals(signal_tx,signal_rx);
signal_out=signal_out(T_Delay:end);
```

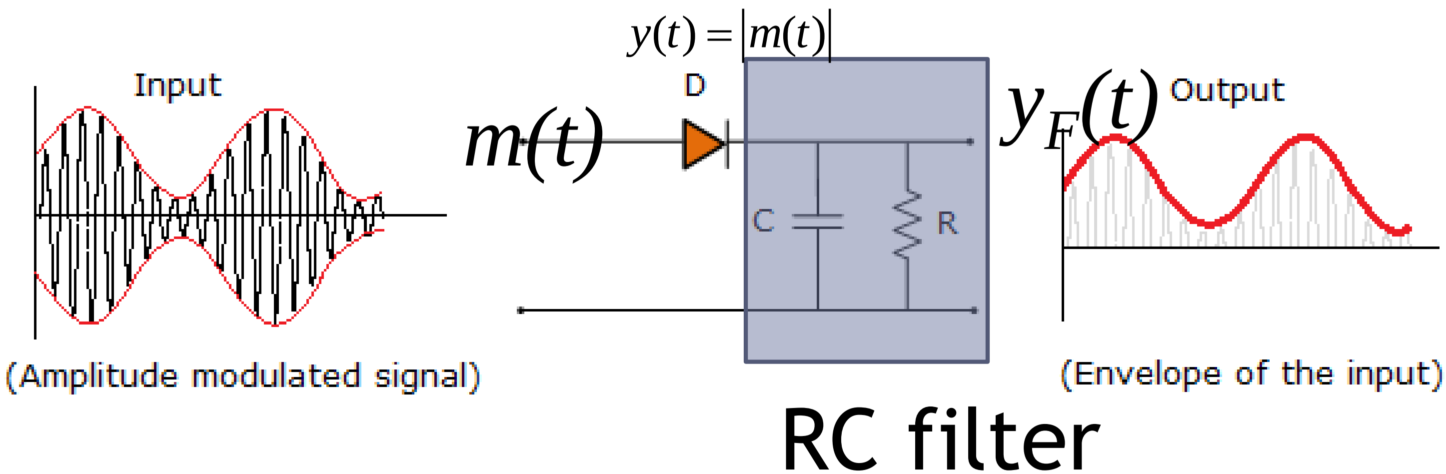
PHASE RECOVERY

- When using a coherent demodulator, phase of the local oscillator can be an issue
  - It should be "locked" (equal) to the phase of the carrier used at transmitter
  - Otherwise is like to receive on wrong reference axes

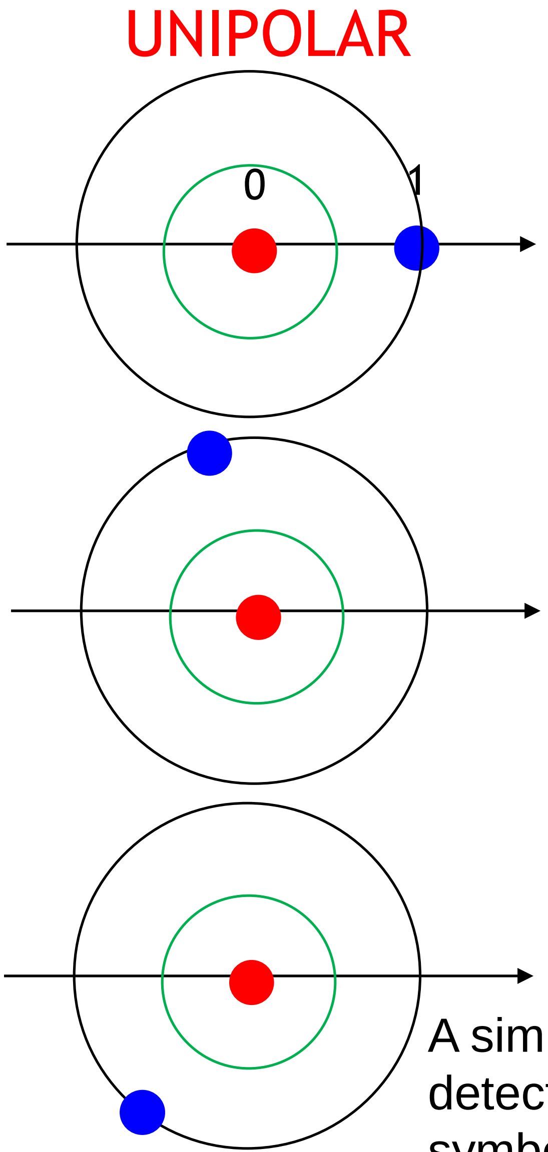


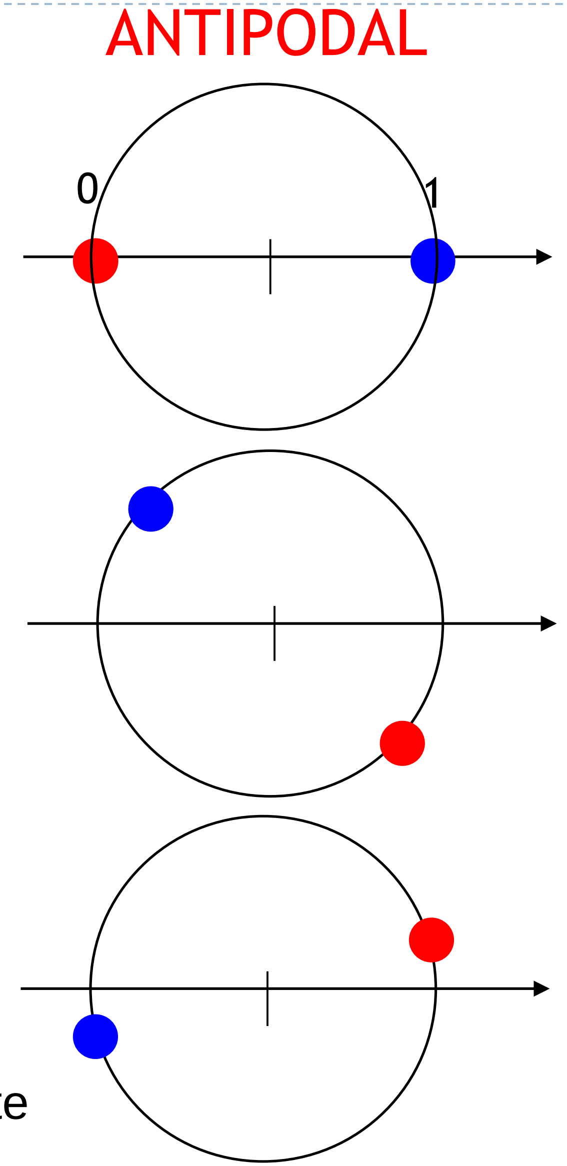
PHASE RECOVERY

- To avoid the need to recover phase, we can use “pure” amplitude modulation, i.e. truly mono-dimensional
- 2-PAM unipolar or any M-PAM unipolar does not need phase recovery
- Such system can also be received using am envelope detector



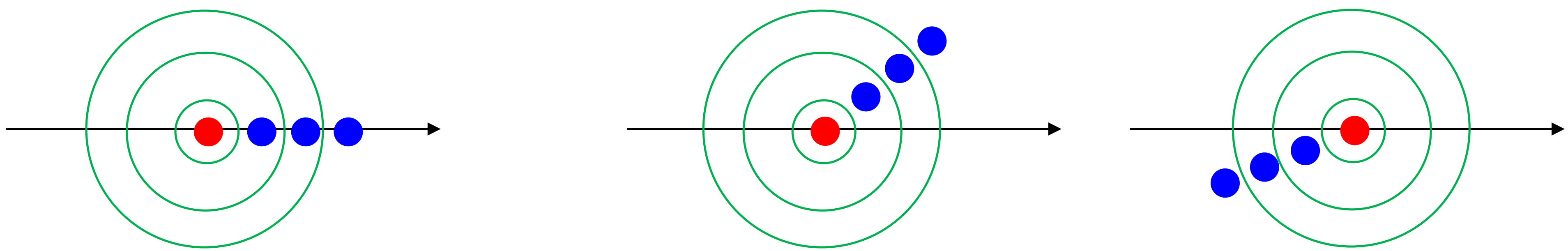
PHASE RECOVERY





A simple threshold detector can discriminate symbols

PHASE RECOVERY



A simple multi-threshold detector can discriminate symbols