Synchronization

Huseyin Arslan

Electrical Engineering Department
University of South Florida

- The objective is to understand synchronization to digital signals at the receiver.
- Time synchronization
  - Coarse synchronization
  - Fine synchronization via correlating the received signal with a preamble
- Frequency synchronization, reasons for frequency mismatch; (Local oscillator differences)
- Symbol synchronization
- Sample timing estimation
- Effect of synchronization in the system; observing the effects of time/frequency offsets in the constellation diagram
- Understanding how to do frame synchronization, burst and symbol synchronization, and symbol timing synchronization.
- Learning how to sample the data and convert these samples to symbols, then, convert symbols to bits.
- AFC loop, channel tracking
- Feedforward versus data directed tracking
- Blind synchronization versus data directed approaches. Some examples for blind synchronization (note that for blind synchronization we need to generate the data with single numerology)
- Detailed description of Frame Structure. Training period and data period
- Various training approaches
- Various synchronization techniques

Coherent Reception

This lab is considered as one of the most important labs; as you will capture a real wireless transmitted signal, and design a digital baseband receiver to analyze it and extract the data from it.

Coherent Reception

- The type of the receiver that we will design is a coherent digital baseband receiver, which is considered one of the most popular receiver designs.
- Coherent Reception means that the receiver has the ability of synchronizing to what is being transmitted.

Coherent Reception

What makes the receiver Coherent?

- Synchronization
- Channel Estimation, which include:

a) Phase

b) Amplitude

(in this lab we will use the simplest channel estimation, as we will have one tab channel because our data rate and our operating environment is designed so that you will observe one tab channel)

- Detection and estimation
- Source decoding (converting the bits to source)

- Frame structures
- Preamble and midamble
- Pilots
- Preamble + pilots
- Choice of training and choice of frame

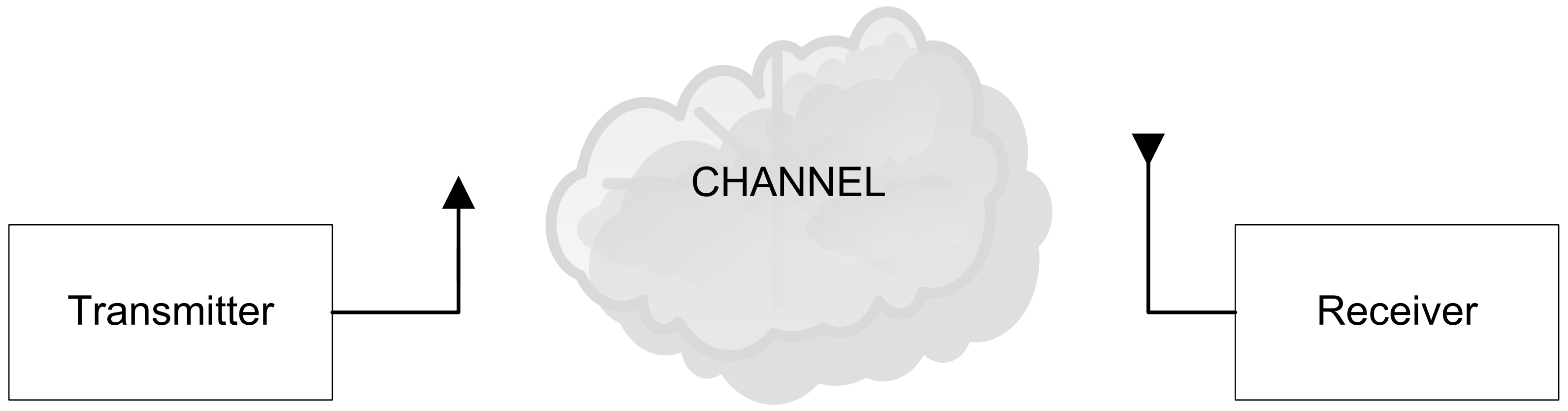
- Feedforward estimations

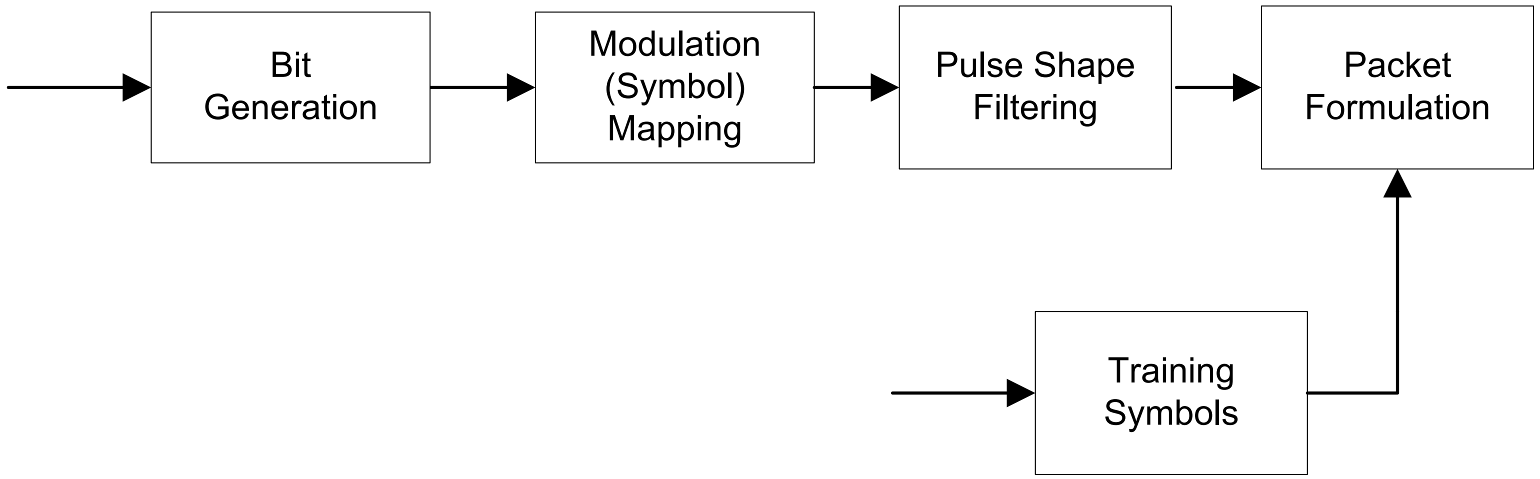
- Regular pilot directed updates
- Blind data directed updated

- 2-D pilots (time and frequency)

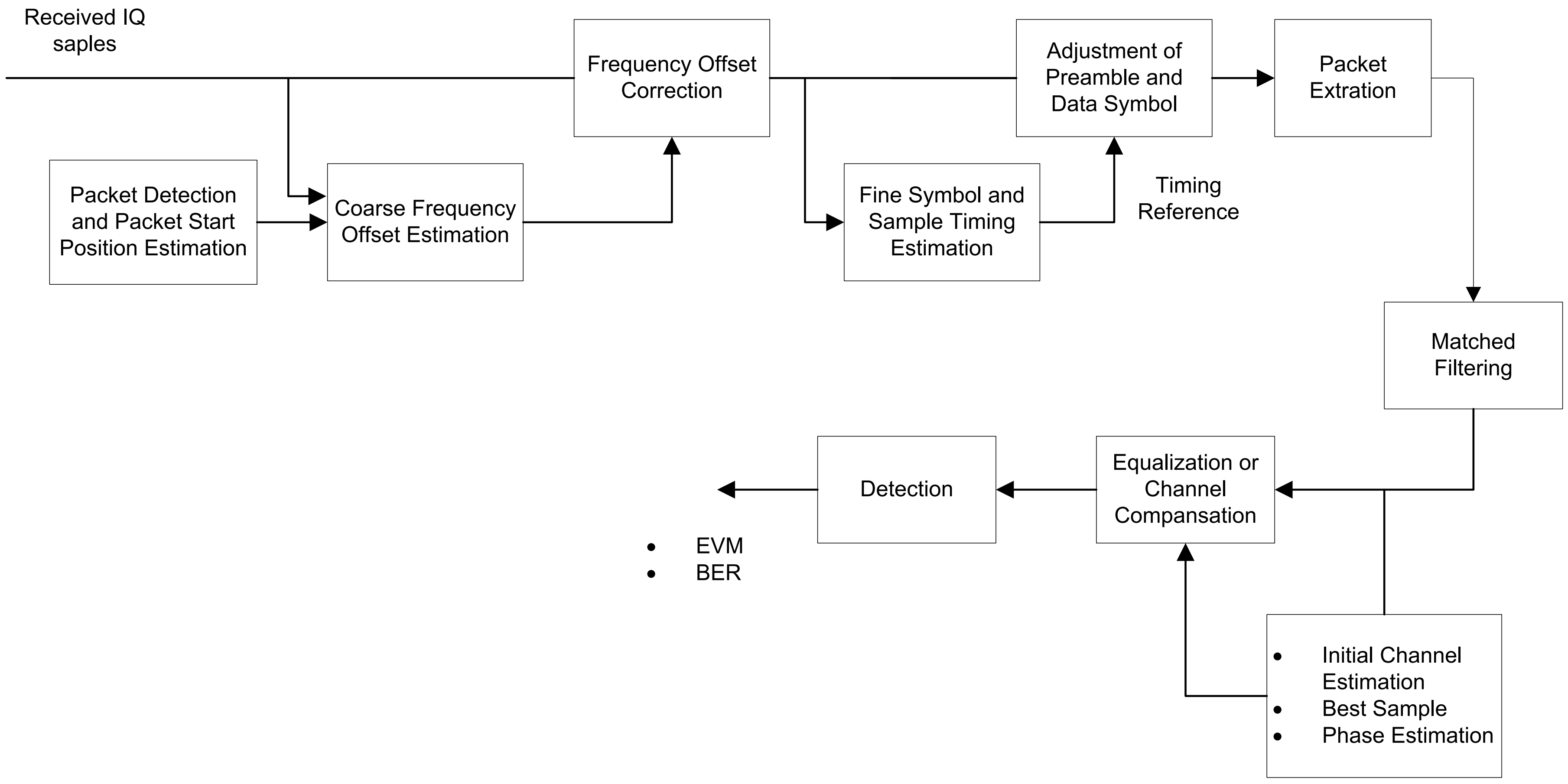
Estimation versus detection

- Estimators
- Models for the estimators
- Estimation of model parameters
- Updates and corrections
- Dependence of the estimator with frame and pilot structures
- Detection
- Data directed approaches (detection and updates)


Transmitter
CHANNEL
Receiver



Transmitter Block Diagrams (Single Carrier Systems)



Received IQ saples
Frequency Offset Correction
Adjustment of Preamble and Data Symbol
Packet Extration
Packet Detection and Packet Start Position Estimation
Coarse Frequency Offset Estimation
Fine Symbol and Sample Timing Estimation
Timing Reference
Matched Filtering
Detection
Equalization or Channel Compansation
• EVM
• BER
• Initial Channel Estimation
• Best Sample
• Phase Estimation

Receiver Block Diagrams (Single Carrier Systems)

USF UNIVERSITY OF SOUTH FLORIDA

Synchronization

The synchronization part will include:

1- **Time synchronization:**

- Coarse synchronization:

Which means that we roughly need to know when the packet starts

- Fine symbol timing synchronization.

Once we know when our packet start, we need to do a fine synchronization to detect the reference synchronization point where we will be able to distinguish each symbol (usually we try to find the first transmitted symbol)

- Optimum sample timing synchronization.

Find the best sample position.

Synchronization

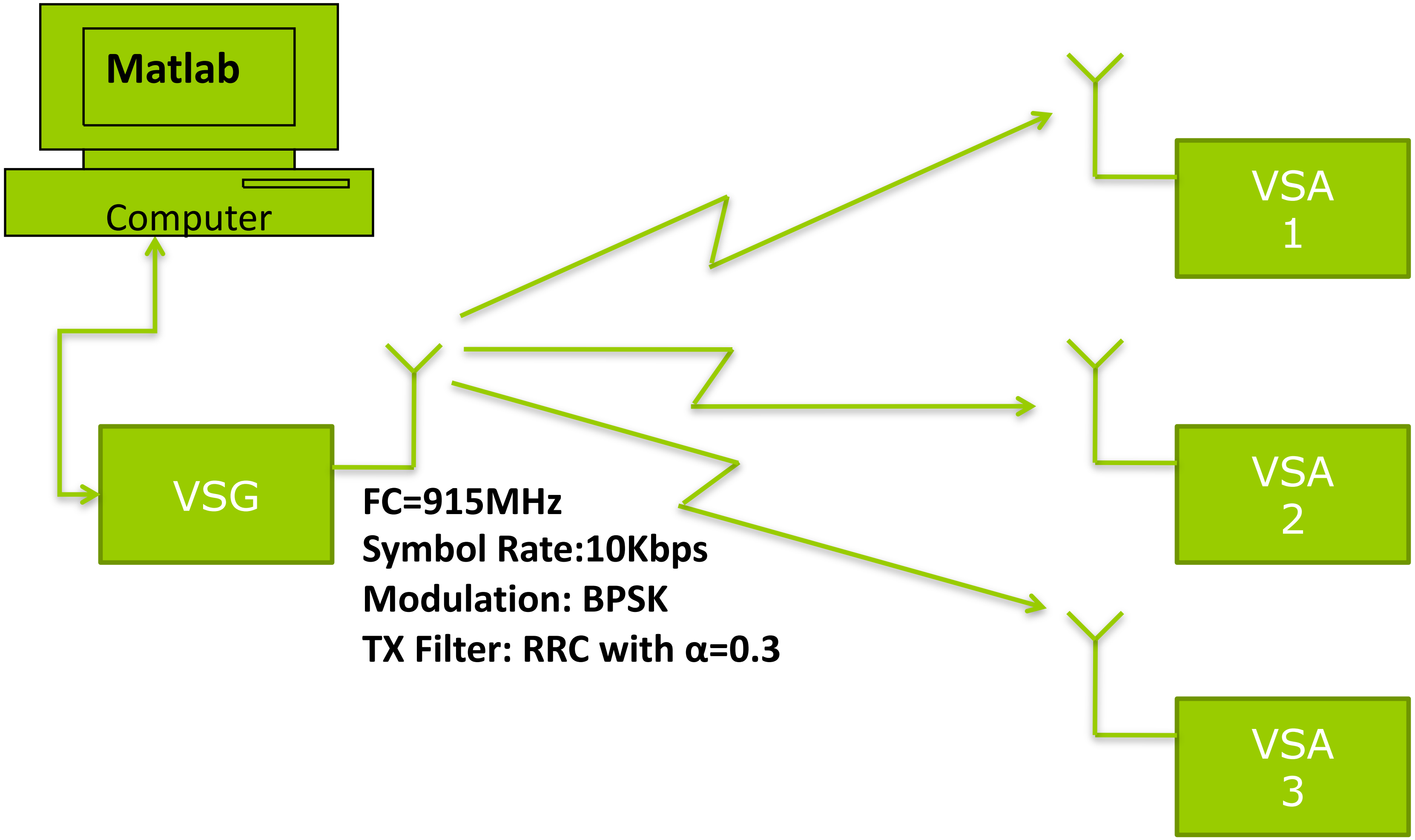
2- **Frequency Synchronization.**

Frequency offset estimation and compensation. The received signal will suffer from frequency offset, due to the mismatch of the transmitter and receiver. We need to know this offset to compensate it otherwise we will not be able to detect the data.

3- **Phase Synchronization**.

Phase offset estimation and compensation.

The Setup of the Lab



FC=915MHz
Symbol Rate:10Kbps
Modulation: BPSK
TX Filter: RRC with α=0.3

USF UNIVERSITY OF SOUTH FLORIDA

Steps

- First Step:

Tune your VSA to the signal, and perform a demodulation analysis in your VSA, and check the received signal.

Tip: you will notice that your VSA will have some modulation points at the zero. This due to the burst type of our signal.

Steps

- Second Step:

Download the received signal to a matlab file, so that you will process it later by matlab.

Tip: after you download the signal, check it using matlab, and plot the power vs time.

Plot(10*log10(abs(Y)))

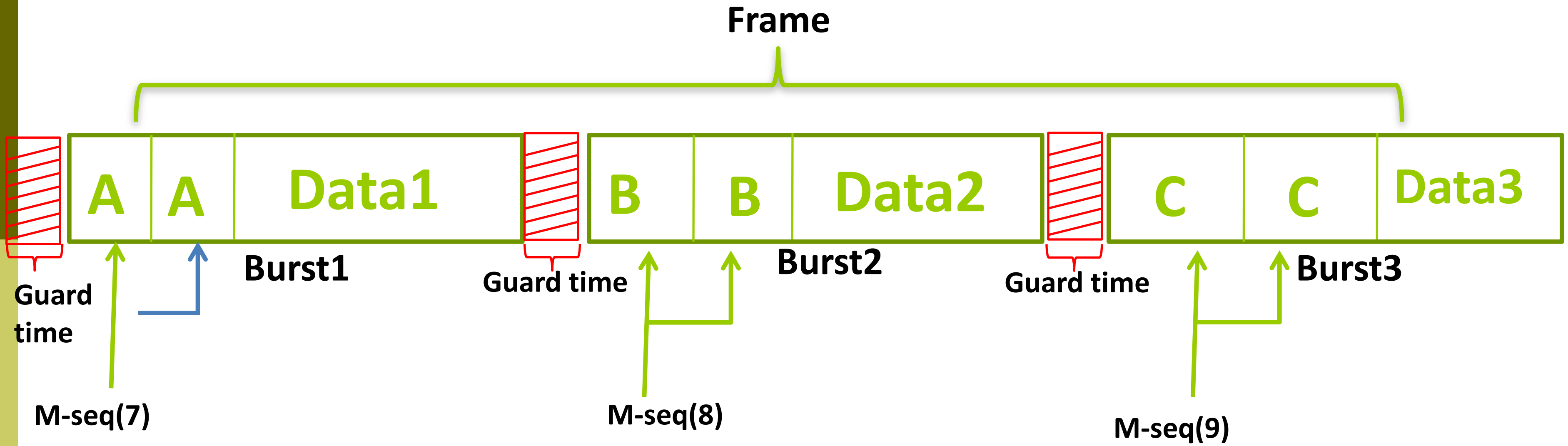
Steps

- Before developing any algorithms to process data from a received signal, we need to know and understand the transmitted signal.

-TX wave form.

-Frame and data structure.

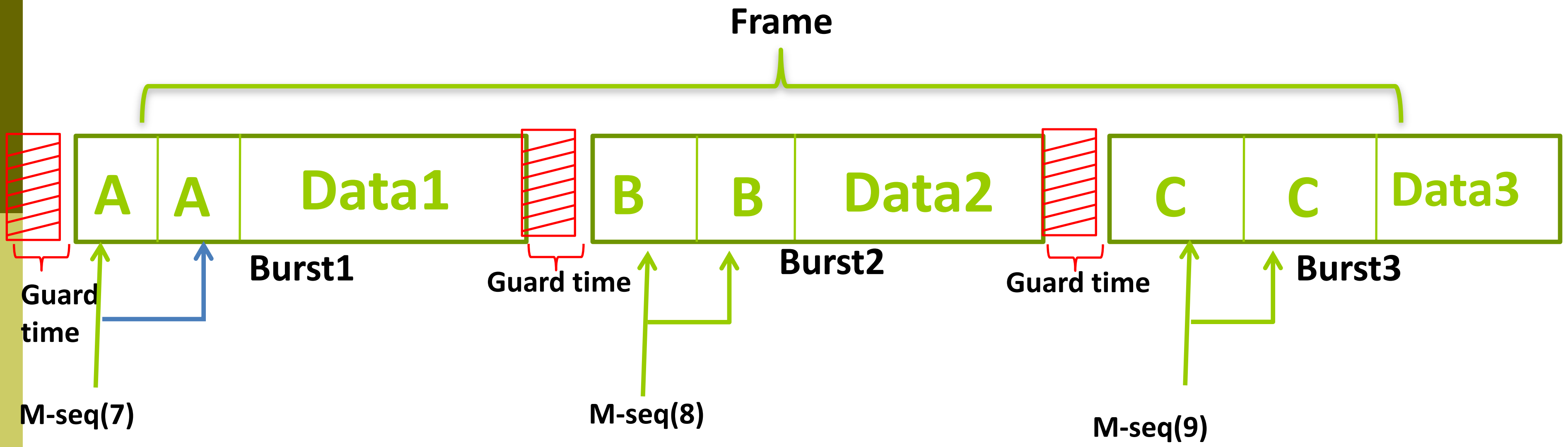
TX Waveform\Frame structure



- The transmitted signal is a frame of three bursts of data.
- Each burst of data will be assigned to a different bench.
- You will design your algorithms to process the data in the burst that is related to your bench (similar to the TDMA system in GSM, where each time slot is assigned to a different user)

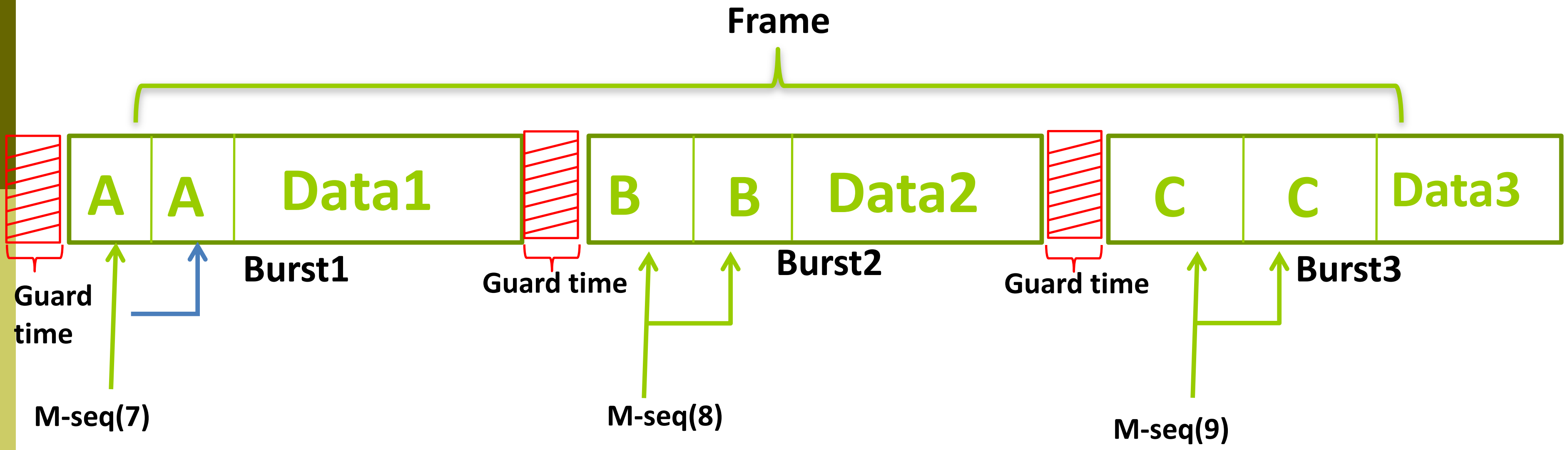
UNIVERSITY OF SOUTH FLORIDA

TX Waveform\Frame structure



- Each Burst begins with a training M-sequence, repeated twice, so as to ease the detection of the beginning of your burst.
- The length of each M-sequence, and the symbols are different in each burst.
- The data starts immediately after the two training sequence. After the data, there is a gab to separate adjacent bursts.

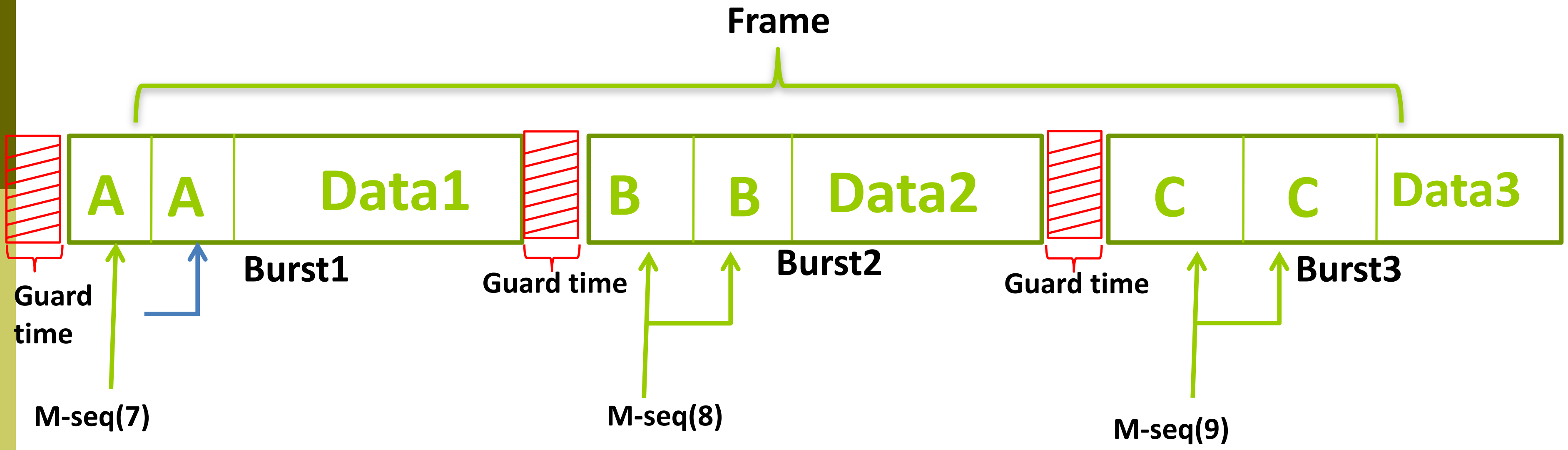
TX Waveform\Frame structure



- Each M-sequence have a length of 2^n-1
- Bench One will have a sequence with n=7.
- Bench Two will have a sequence with n=8.
- Bench Three will have a sequence with n=9.

UNIVERSITY OF SOUTH FLORIDA

TX Waveform\Frame structure



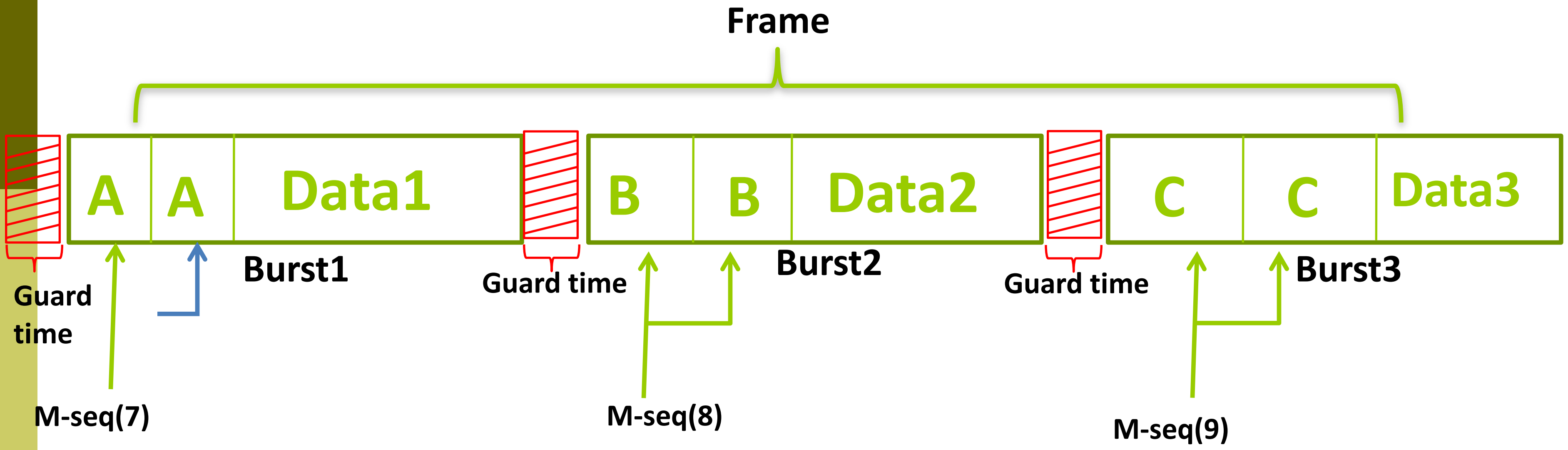
- The Modulation is BPSK.
- The mapping of the modulation is

$1 \longrightarrow +1$

$0 \longrightarrow -1$

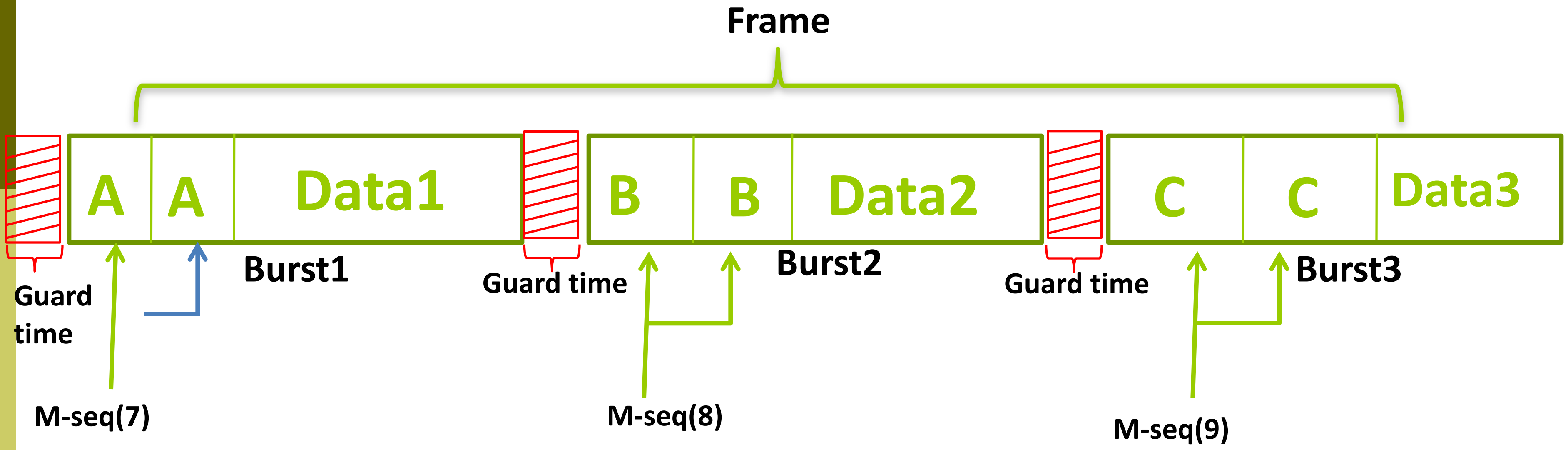
UNIVERSITY OF SOUTH FLORIDA

TX Waveform\Frame structure



- The data Length is 91 Symbols, in each burst.
- The Guard time between each Burst is the same as the data duration, which is 91 Symbols.

TX Waveform\Frame structure



- The M-sequence subroutine is already uploaded to the blackboard.
- To call the M-sequence function you just need to call the function:

mseq(2,n), where 2 represent that we will produce a binary digital sequence, and n represents the order assigned to each bench(7,8,9)

UNIVERSITY OF SOUTH FLORIDA

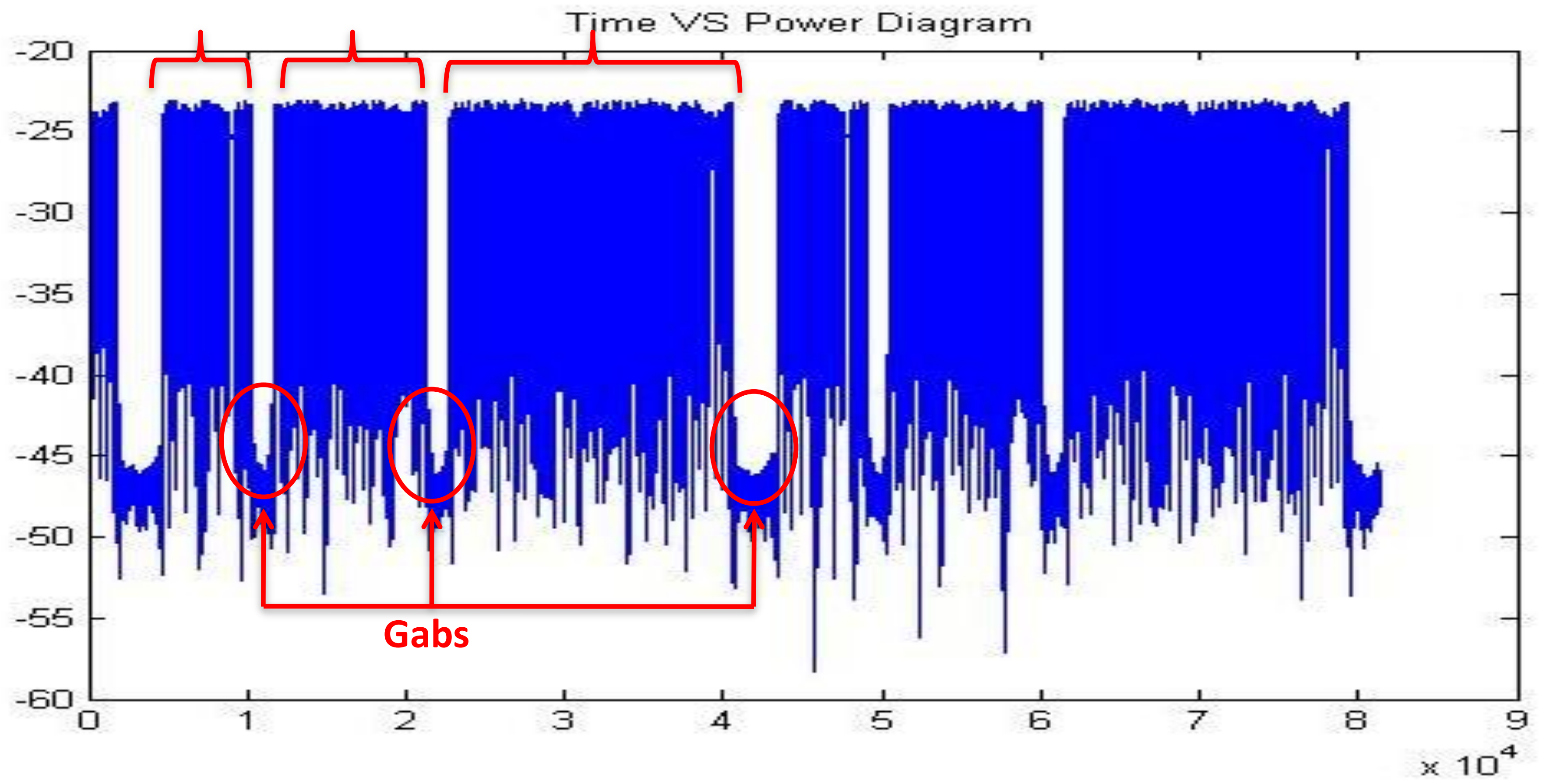
Downloading the Signal through the VSA

- We need to adjust the capture time in the VSA to proper value that will guarantee that we will capture at least one full frame.
- Frame duration= 2328 bits. and Our bit rate is 10Kbps.
- In order to capture two frames we will need about 0.5 sec.
- Also make sure that your VSA will have an oversampling rate of 16 when you capture the signal.

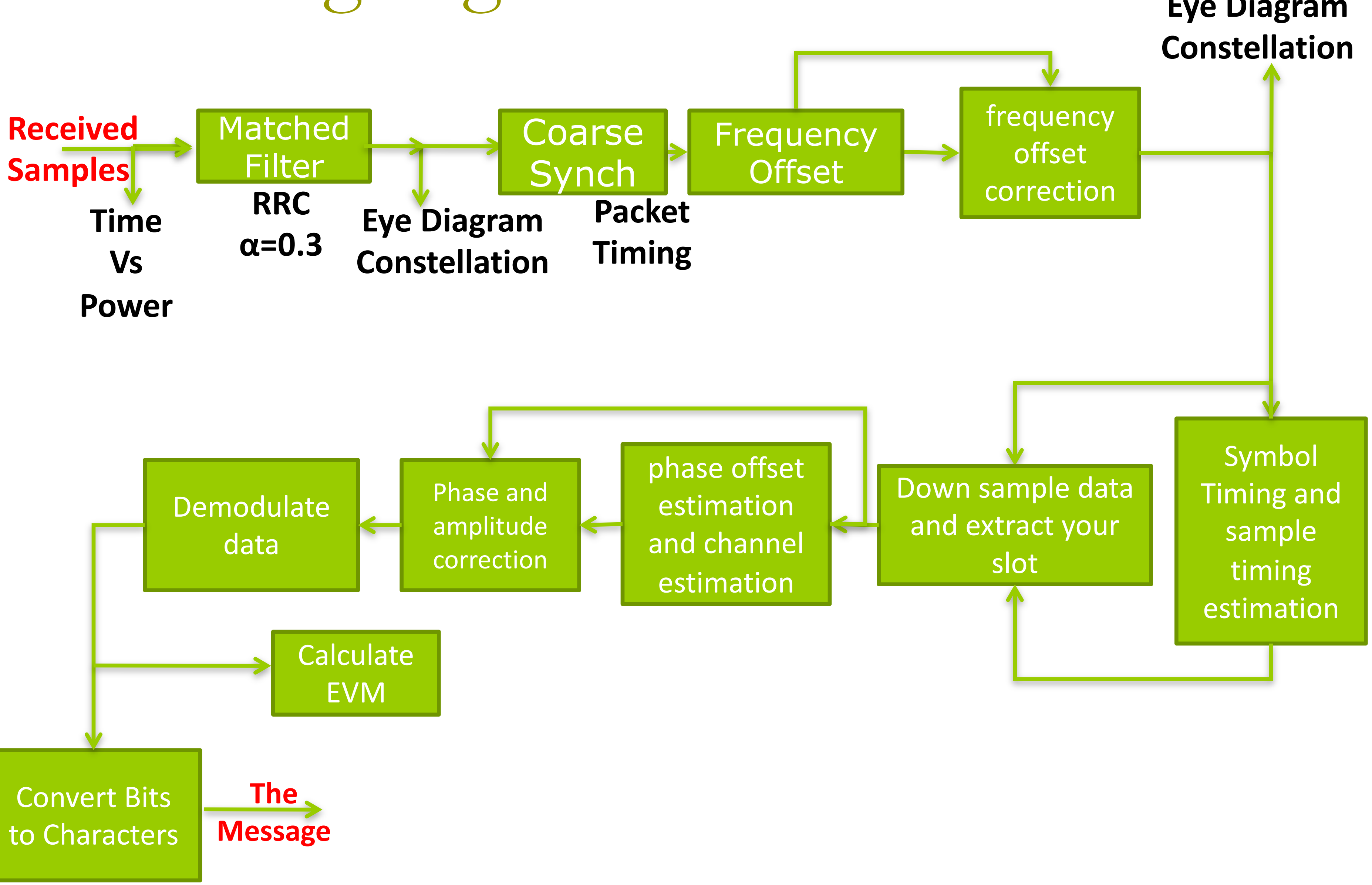
Tip: <u>you will control the oversampling rate by adjusting the span as you learned earlier in the second lab.</u>

Downloading the Signal through the VSA

- Save the captured signal as a matlab file, check the time VS power diagram of your captured data and make sure that you have at least one full frame.



Processing Algorithms


Received Samples
Time Vs Power
Matched Filter
RRC α=0.3
Eye Diagram Constellation
Coarse Synch
Packet Timing
Frequency Offset
frequency offset correction
Eye Diagram Constellation
Symbol Timing and sample timing estimation
Down sample data and extract your slot
phase offset estimation and channel estimation
Phase and amplitude correction
Demodulate data
Calculate EVM
Convert Bits to Characters
The Message

Matched Filtering

- You could use the standard defined filter in matlab.
- Or you could use your own defined filter.
- Decide what kind of filter you will use, and use it with the proper parameters and proper ɑlpha

The standard rcosine filter in matlab could be defined by: rcosine(x,y,'sqrt',z)

Where: y=oversamping rate, z = roll of factor

- Filtering operation can be implemented by convolving the signal and the filter. Or you can use the filter command in Matlab.

Coarse Synchronization
Packet Timing estimation

- There are many methods to perform coarse synchronization. Some of the possible ways are:

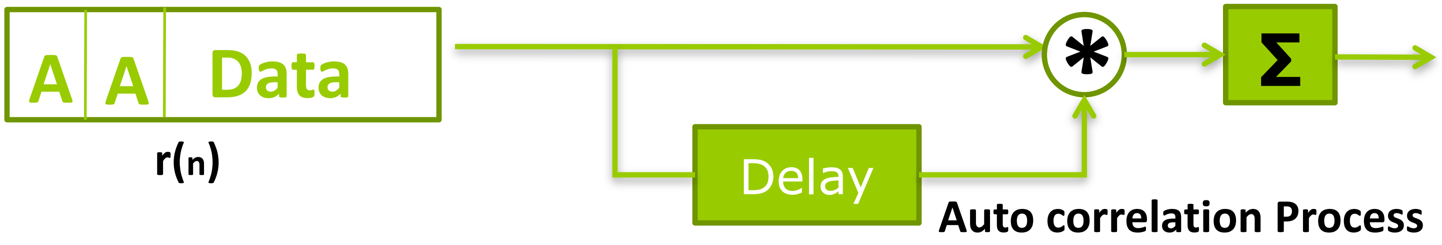
I. Energy Detection.

Detect the positive edge of the burst. But this may give you some hard time because you have more than one edge as you have three bursts. You need to find out your burst.

Coarse Synchronization
Packet Timing estimation

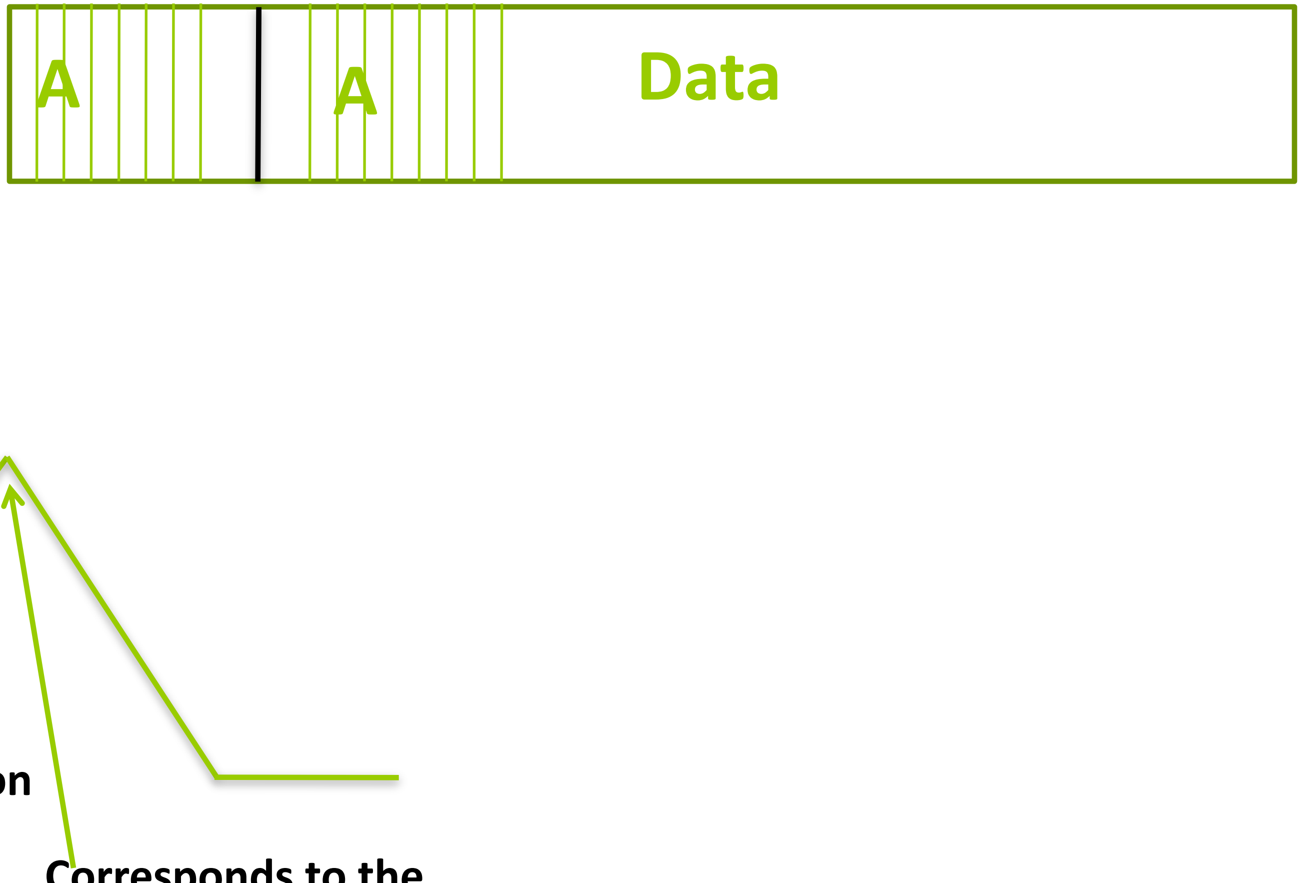
II. Auto correlation.

A very easy and straight forward method to perform a coarse synchronization.



Since we have a training sequence repeated twice, the auto correlation should give us a detection on where the first training begin

Coarse Synchronization
Packet Timing estimation



The result
of the Auto correlation

Corresponds to the
packet beginning

Coarse Synchronization Packet Timing estimation

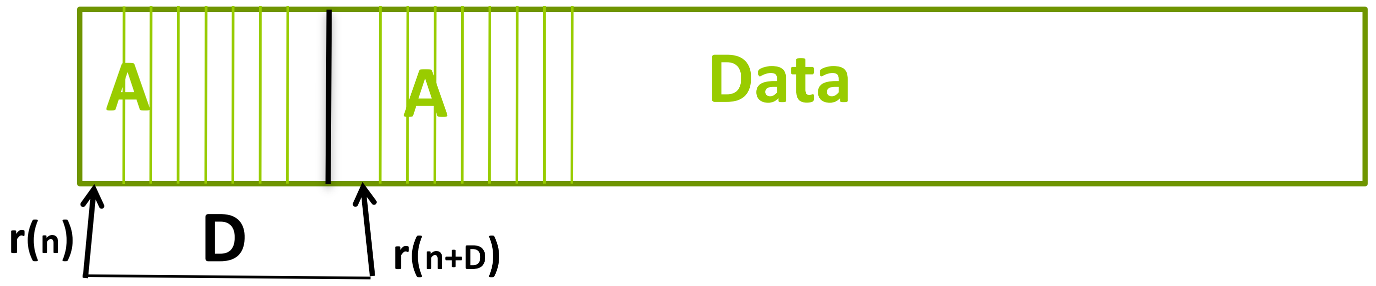
- Due to the noise the resulted auto correlation function will be distorted and the peak of this function may not be very accurate.
- We need to do a fine synchronization to find out the exact symbol position

USF UNIVERSITY OF SOUTH FLORIDA

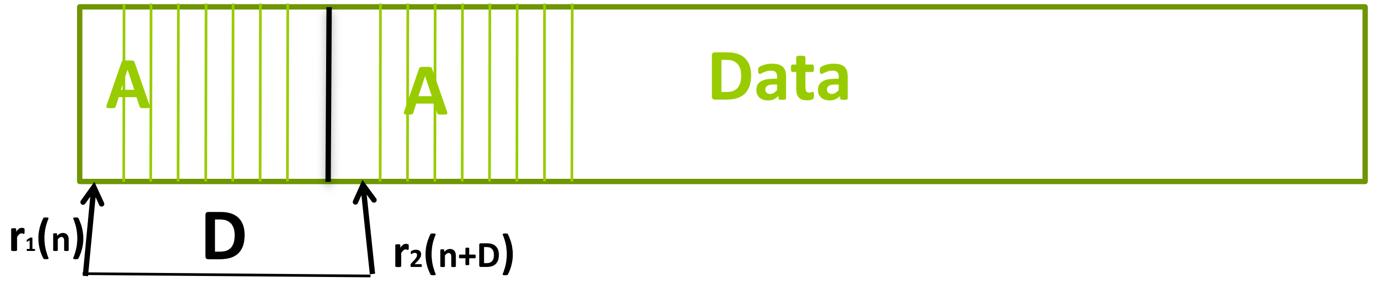
Frequency Offset estimation.

- Due to the mismatch between the transmitter and receiver oscillators, the generated frequencies are different .
- This introduces a frequency offset in the received data symbols.

Frequency Offset estimation.



Frequency Offset estimation.



- Calculate =**r(n).conj r(n+D)**
- Find the angle, then divide it by 2∏D, this will give you an estimate for the frequency offset fo.

USF UNIVERSITY OF SOUTH FLORIDA

Symbol timing estimation(fine synch.)

- There are also many ways to do the fine synchronization and symbol timing.
- One of the simple ways is to perform a cross correlation on your signal.
- To get the optimal sample timing:

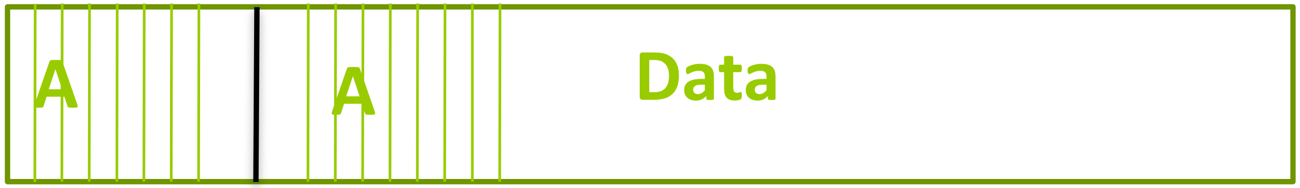
1. Down-sample each possible sample timing.
2. Cross correlate with the M-sequence that you generate locally.
3. The best sample time is the one that maximize the cross correlation
4. Since we don’t know the best symbol timing, we will need to do the above steps for all possible symbol timing. That means we will use two for loops.

Phase offset estimation

- Phase offset is: What you received after down-sampling is what you transmitted plus some amplitude and phase error. Which may lead to some rotation in your constellation diagram. So we need to estimate the channel amplitude and phase.

Phase offset estimation

- Again, there is many possible ways of doing it.
- Here is one of the simple possibilities.



- $\hat{r}(n) = Tx(n).Ae(j\theta)$
- To estimate **A** & **ϴ**: calculate $\mathbf{r(n)/\overline{Tx}(n)}$
- $\overline{\mathbf{Tx}}\mathbf{(n)}$ is training sequence, generated at **Rx**
- Get $\mathbf{Ae(j\Theta)}$
- Since we know what we are transmitting over the training sequence

UNIVERSITY OF SOUTH FLORIDA

Demodulate the data

- Now since you corrected the phase and frequency, you can demodulate the data, as you learned earlier in the first lab.

Converting the bits to characters.

- The last thing you need to do now, since you have the bits.
- Convert the bits to characters. Search the matlab and find out how to convert the bits to characters.
- If you needed help you could refer to the TA's