Jacobs University Bremen

CO-522-B Intersession 2022

Communication Basics Lab

Lab Write-Up 2 : Receiver Simulink Study

Kuranage Roche Rayan Nimnath Ranasinghe

Conducted on : 12/01/2022 & 14/01/2022

Explain briefly how the receiver works. Also, please highlight anything that was new that you didn't know before!

The receiver's task is to 'undo' all the operations that the transmitter did as discussed in the previous lab. First, a multiplication block is used to multiply the signal by the conjugate of its carrier in order to shift the received signal back to baseband. Next, a matched filter is applied to remove the maximum amount of noise from the received signal. After this, the optimal points of the signal are sampled using a sampler (using synchronization blocks in most cases) to obtain an estimate of the original I/Q symbol weights that were generated in the transmitter. Finally, a decision is made regarding which symbol was transmitted using a decision block and these final symbols are mapped onto bits to recover the original message (these bits make up the original message).

Explain how we can get a RC response by using root filters at the transmit and receive. Explain why this is preferable to using just a single (non root) RC filter somewhere.

Using a root RC filter translates to taking the square root of the RC response in the frequency domain. Hence, when we cascade two root RC filter responses (one in the transmitter and one in the receiver side), this produces an RC shape. Doing this process using two root RC filters also helps preserve the zero-ISI (zero intersymbol interference) property which is much preferable than using one normal RC filter which does NOT preserve the zero-ISI property. However, the issue with root RC filters is that the time response is significantly larger than for a simple RC filter.

Explain the need for synchronization in communications systems, and how this was accomplished in this lab. Also describe what a PLL is and what it can do. What are carrier and symbol timing recovery for?

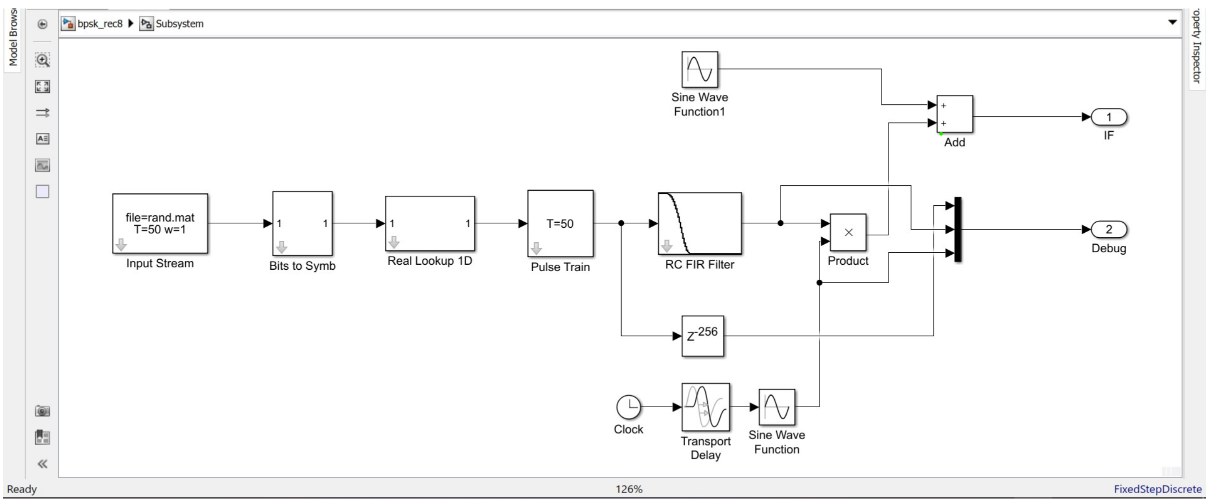
In most real world communication systems, the clocks of the transmitter and the receiver are not the same and the relative delay between transmission and reception is also not fully known. Hence, it is extremely important to correctly identify and locate the optimal sample points in the I/Q signals.This is where synchronization comes in to estimate and correct the sampling points and phase at the receiver in order to interpret the information stream correctly. The synchronization in the lab was done using synchronization blocks; i.e. more specifically, a PLL (Phase Locked

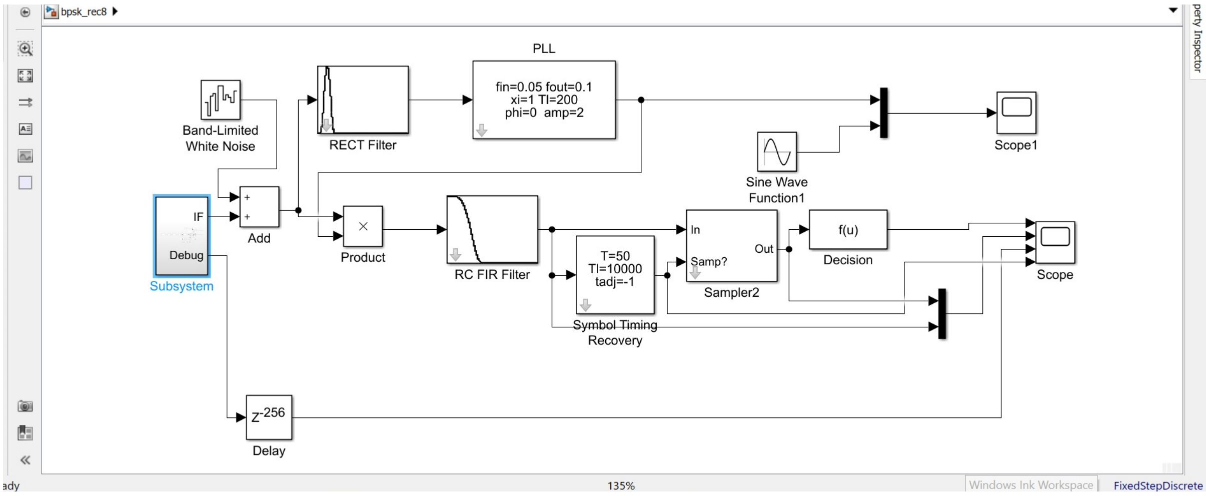
Loop). A PLL is a tracking algorithm for sinusoidal signals. It works by taking in a sinusoidal input and tracking its phase using a feedback loop. Using this input signal, it generates its own sinusoidal signal internally. This generated internal sinusoid's phase is 'locked' to the phase of the input sinusoid. The PLL can also generate any other clock or sinusoidal signals that are needed based on this received phase.

Carrier recovery is used to remove errors in the receiver resulting from a carrier offset between the transmitter and the receiver. In other words, it is used to estimate and compensate for frequency and phase differences between a received signal's carrier wave and the receiver's local oscillator for the purpose of coherent demodulation.

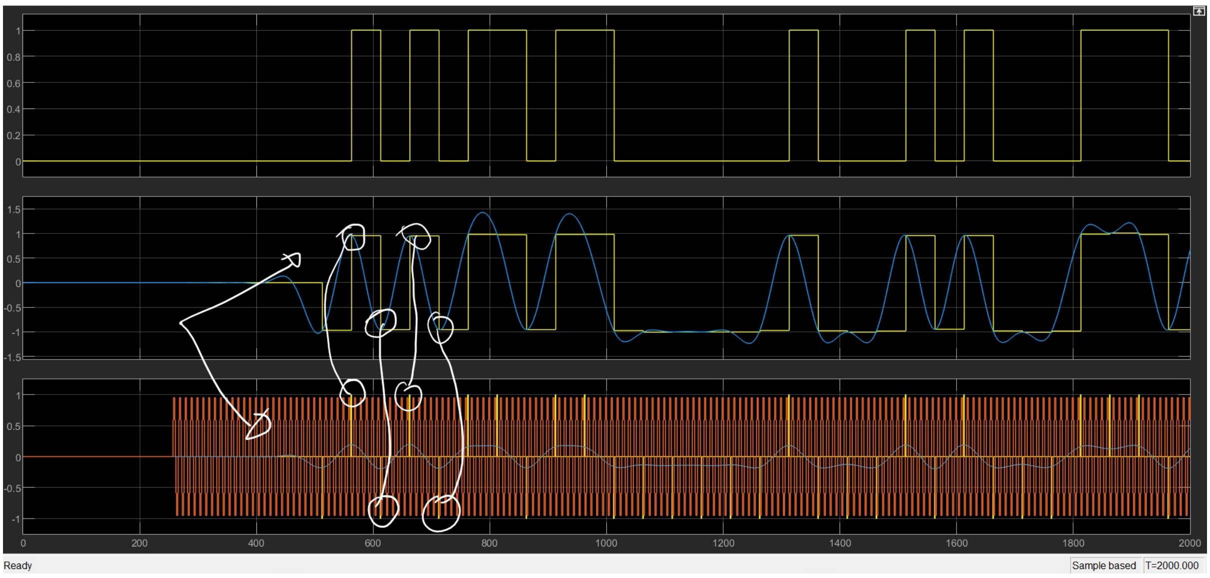
Symbol timing recovery is used to obtain the optimal sample points of the waveform in order for the accurate generation of symbols used to decode the message. It basically works by multiplying the baseband signal with a copy of itself shifted by half of the symbol time. This creates a sinusoid that has a frequency equal to the symbol rate where the zero-crossings of the sinusoid happen at precisely the optimal sample points.

<u>Give plots showing the output of your communications system for two cases: (i) with perfect ideal synchronization, and (ii) with the synchronization blocks. Circle and label points in the printout to "prove" to the TA that it is working.</u>

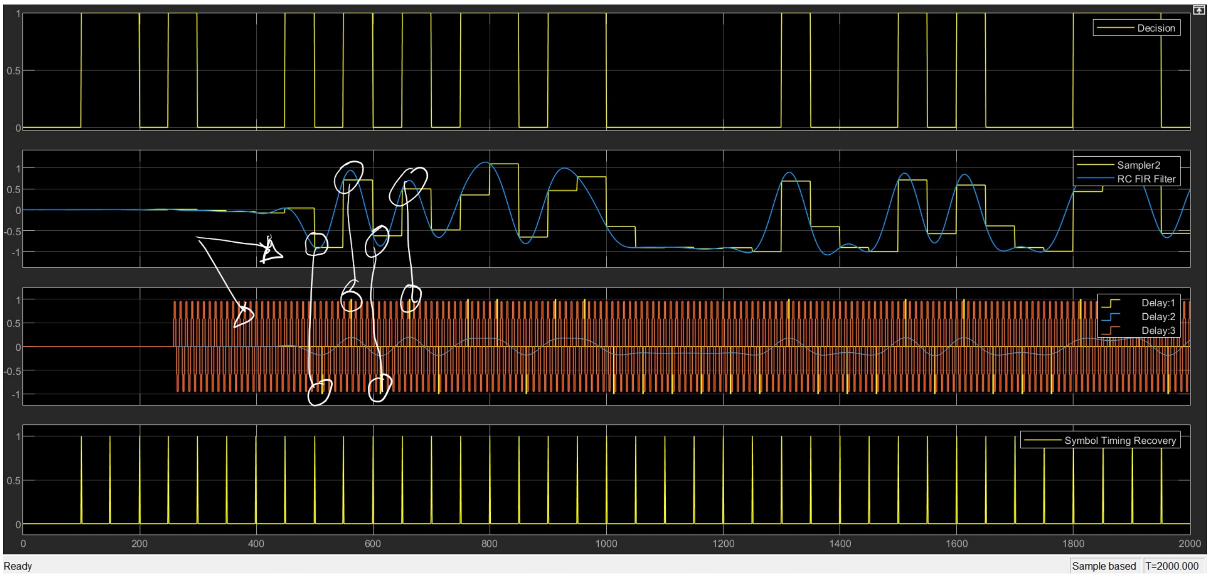




For perfect ideal synchronization:



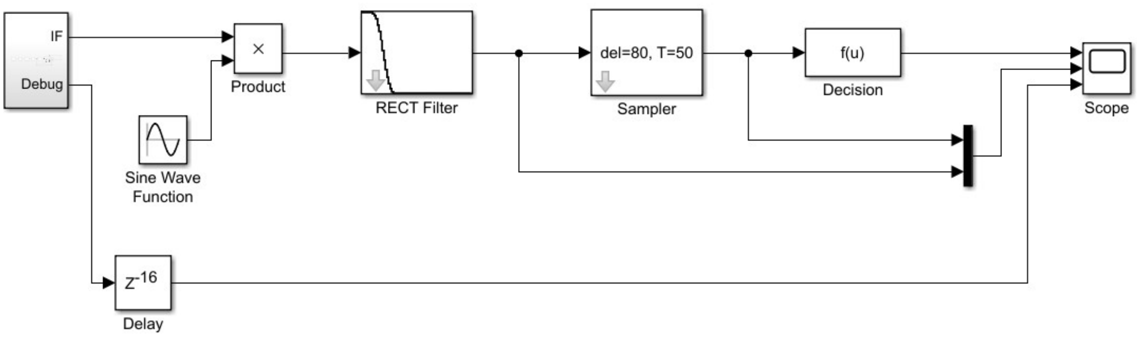
With synchronization blocks:

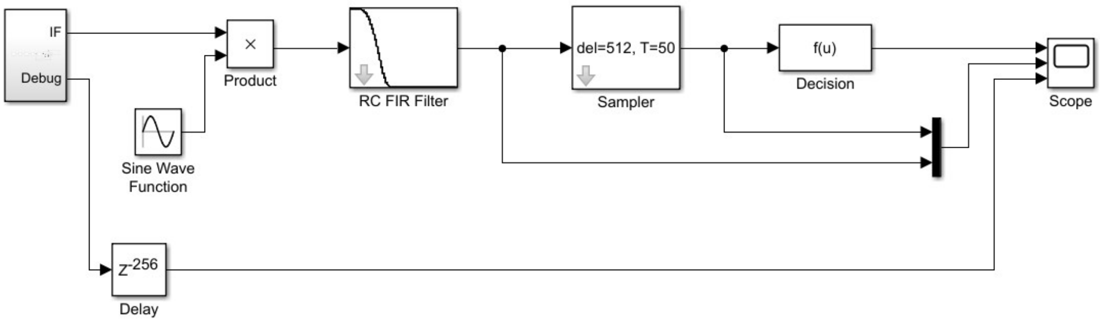


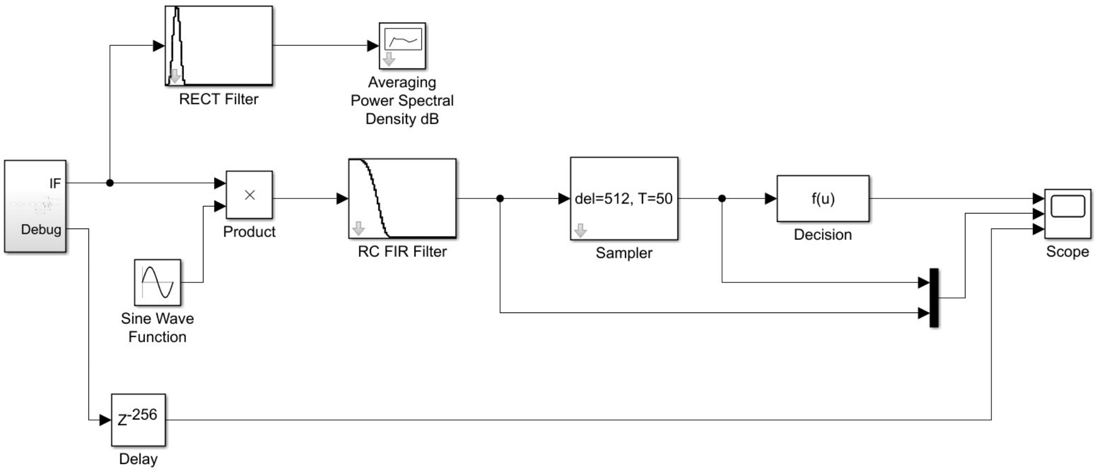
Describe any difficulties you experienced in getting your design to work and how you fixed these problems.

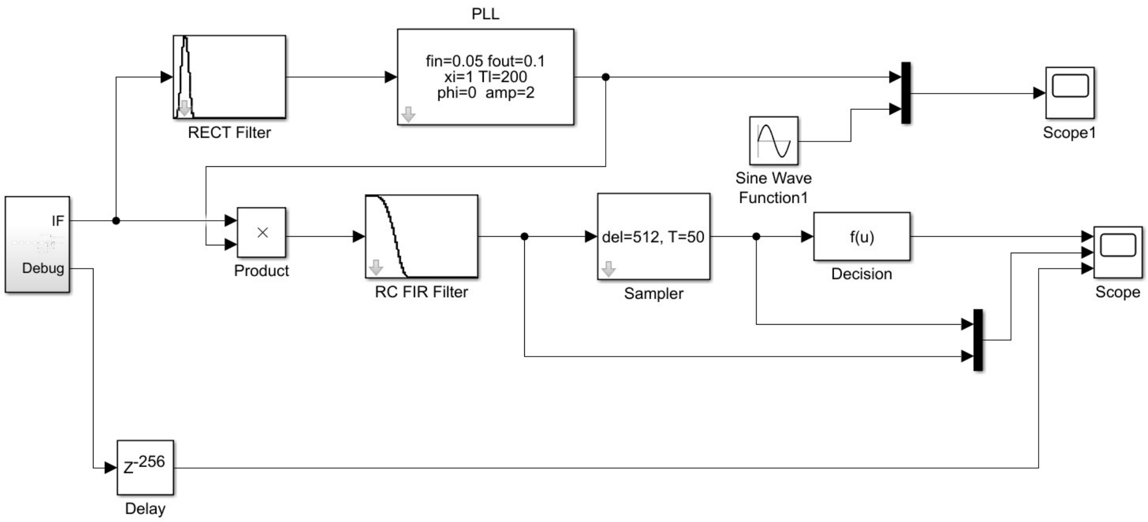
Once again, one of the tougher parts of this exercise was actually understanding what was going on instead of the actual implementation, but this was remedied using a clearer outlook on the blocks themselves and using the scopes to visualise the result as before.

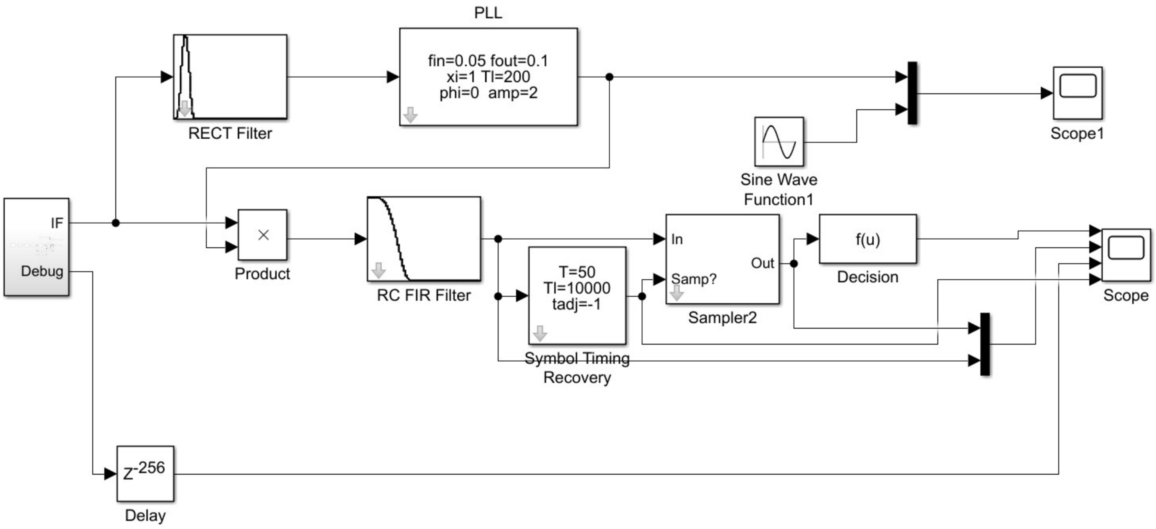
Appendix of images from the lab:

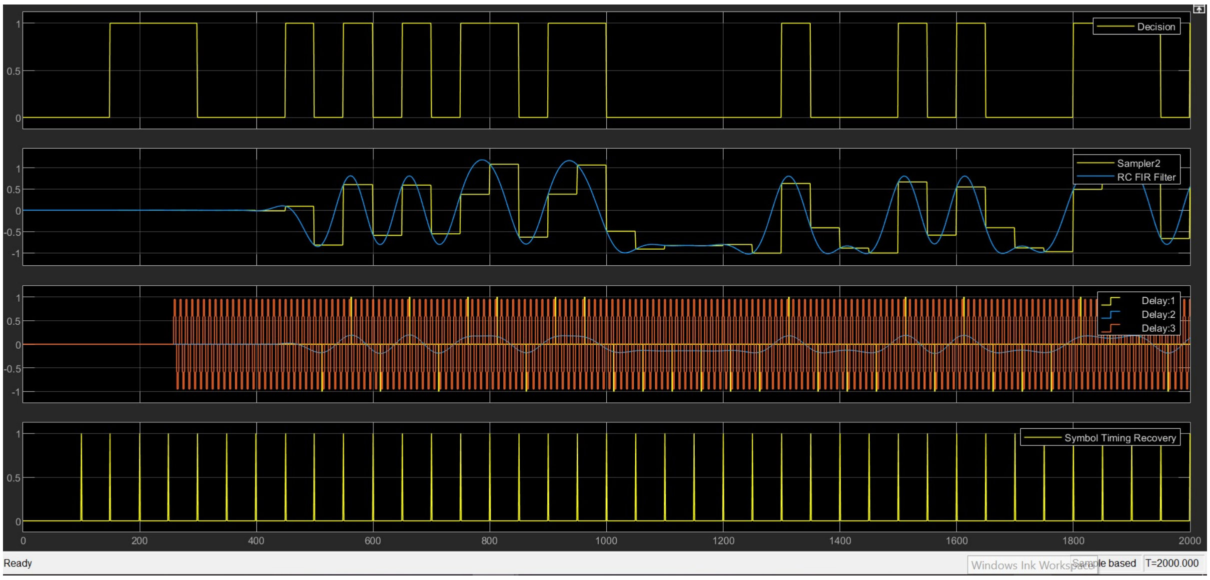


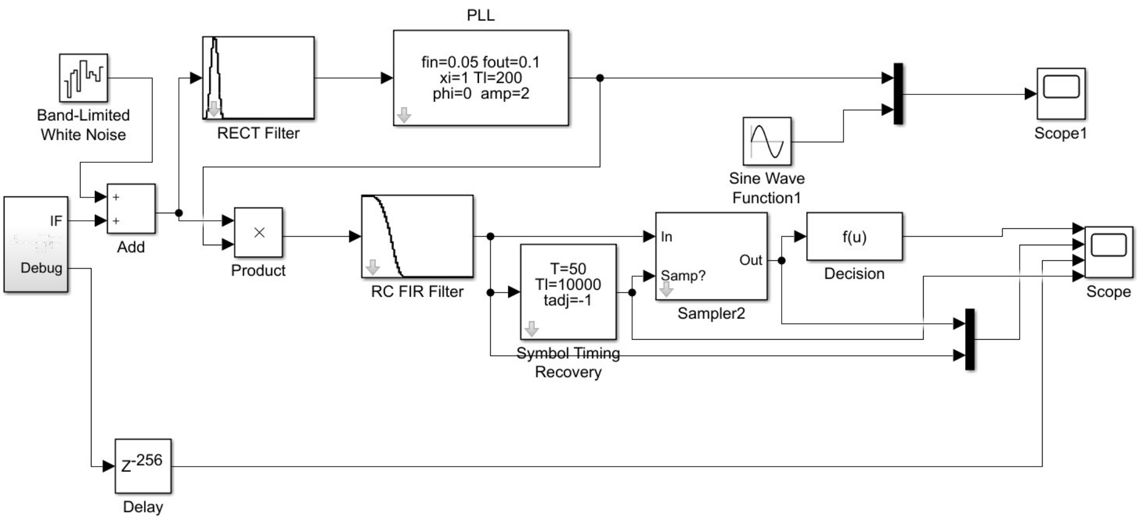

IF
Debug
Product
Sine Wave
Function
RC FIR Filter
del=512, T=50
Sampler
f(u)
Decision
Scope
z-256
Delay

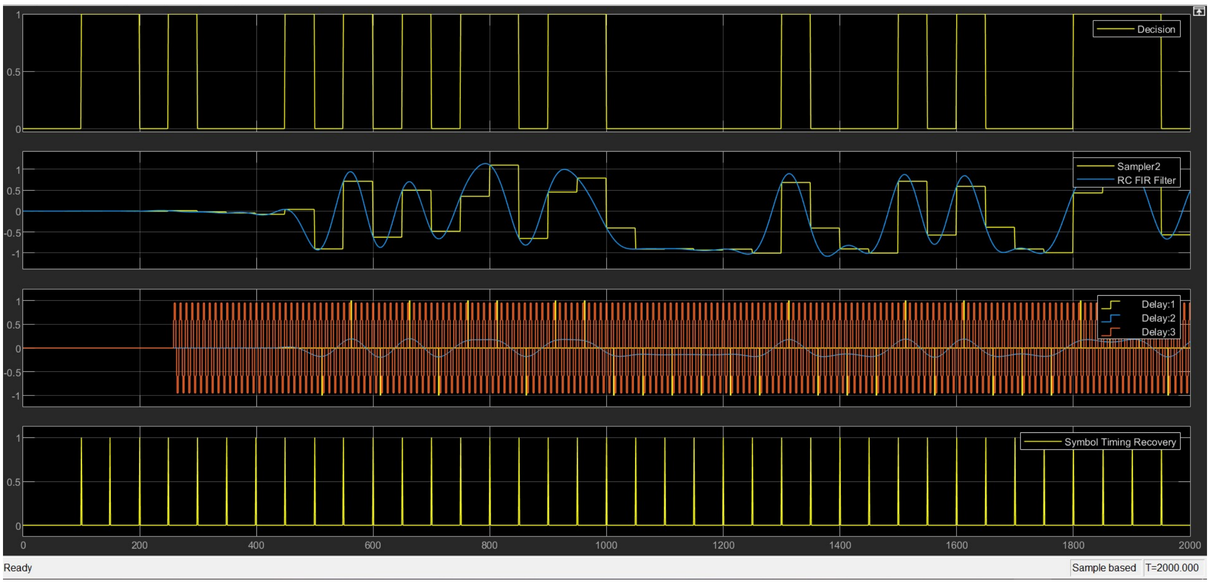

RECT Filter
Averaging
Power Spectral
Density dB
IF
Debug
Product
Sine Wave
Function
RC FIR Filter
del=512, T=50
Sampler
f(u)
Decision
Scope
z-256
Delay


PLL
RECT Filter
fin=0.05 fout=0.1
xi=1 Tl=200
phi=0 amp=2
Scope1
Sine Wave
Function1
IF
Debug
Product
RC FIR Filter
del=512, T=50
Sampler
f(u)
Decision
Scope
z-256
Delay


PLL
fin=0.05 fout=0.1
xi=1 Tl=200
phi=0 amp=2
RECT Filter
Scope1
Sine Wave
Function1
IF
Debug
Product
RC FIR Filter
T=50
Tl=10000
tadj=-1
Symbol Timing
Recovery
In
Samp?
Out
Sampler2
f(u)
Decision
Scope
z^{-256}
Delay


Decision
Sampler2
RC FIR Filter
Delay:1
Delay:2
Delay:3
Symbol Timing Recovery
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Ready
Sample based
T=2000.000


PLL
fin=0.05 fout=0.1
xi=1 Tl=200
phi=0 amp=2
Band-Limited
White Noise
RECT Filter
Scope1
Sine Wave
Function1
IF
Debug
Add
Product
RC FIR Filter
T=50
Tl=10000
tadj=-1
Symbol Timing
Recovery
In
Samp?
Out
Sampler2
f(u)
Decision
Scope
z^{-256}
Delay



Decision
Sampler2
RC FIR Filter
Delay:1
Delay:2
Delay:3
Symbol Timing Recovery
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Ready
Sample based
T=2000.000