EE 340: Communications Laboratory

Autumn 30th March, 2021

Lab 7: Carrier Frequency and Phase Synchronisation in Communication Links

Legends

Question/Observation: Show it to the TA and explain (carries marks)

Recall/think about something

Caution

Additional information - weblink

Aim

- To study carrier frequency and phase offset problem in communication links.
- To design a Costas loop for achieving carrier frequency/phase synchronisation for QPSK signals.
- To get familiar with the dynamics of a phase-locked loop using the Costas loop (which is also a PLL).

PreLab Work

- Go through the prelab study material.
- Revise your concepts of control systems – Bode plots, stability criterion (gain margin, phase margin), pole zero compensation for stability.
- Dynamics of a second order system: calculation of the natural frequency of oscillation of the feedback loop, damping factor and settling time.

PART1: QPSK signal with frequency offset

- Generate QPSK signals with sps = 4 (samples-per- symbol) and sample rate of 320 kHz.
- Use Random Source, followed by Chunks to Symbols, followed by Polyphase Arbitrary Resampler for this.

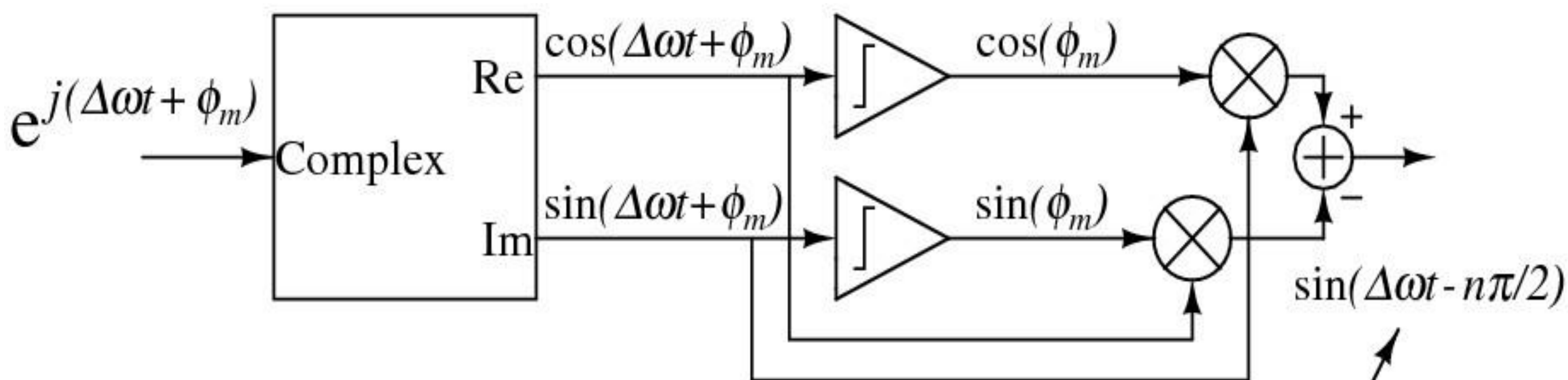
- Now multiplythe incoming QPSK signal with 0.1Hz complex sine wave (i.e with $e^{j* 2*pi*f*t}$)

Use “Polyphase Clock Sync“ with output SPS=1 and plot the constellation in XY mode on a Scope Sink.

Observe that the constellation is rotating (Why?)

The signal from the multiplier output will be used to effectively represent “a down-converted baseband signal with 0.1 Hz carrier frequency offset“.

PART2: Make a QPSK carrier phase detector



Assuming ϕ_m is $\pm(\pi/2 \pm \pi/4)$, and the comparators generate $\pm 1/\text{sqrt}(2)$, we get this output.

Here n is such that $-\pi/4 < \Delta\omega t - n\pi/2 < +\pi/4$

- Make the above Phase Detector. The above phase detector will have phase ambiguity of ±(nπ/2)
- A saw-tooth wave with frequency 40 Hz should be observed if you apply a frequency offset of 10 Hz in the Part 1 (Why?)
- Use Threshold Detector blocks for the above comparators (threshold levels (+0.001,-0.001), followed by a subtraction of 0.5 (Why?)
- The Threshold Detector block implements a Schmitt Trigger (comparator with hysteresis with the two threshold levels).
- Use a low pass IIR filter to suppress noise (FF coefficients: 0.01; FB coefficients: [-1,0.99]; Old style of taps: True)

PART3: Feedback signal through udp port

- GNU-Radio software doesn't allow blocks connected together in a feedback loop (however, a Python or C++ code can be written to have feedback within the block).
- However, Costas loop requires a feedback. To overcome this limitation, we will give our phase detector output to a udp sink port and get it back through a udp source port.
- The Destination address and Destination port of both the udp sink and source is 127.0.0.1(refers to the IP of the localhost) and 12345 (this can be any unused udp port) respectively. Let the rest of the options remain in their default state.
- However, there can be a problem that the udp source sends more number of packets leading to dropping of packets. To avoid this we introduce a delay of 10 samples after the udp source block.
- Make sure thatthe output from the udpsource block was not saturating, by multiplying with an appropriate constant.

PART4: Complete the Costas Loop

- Now give the resulting signal to the Loop filter (discussed in the prelab material) implemented using an IIR filter with the following parameters:

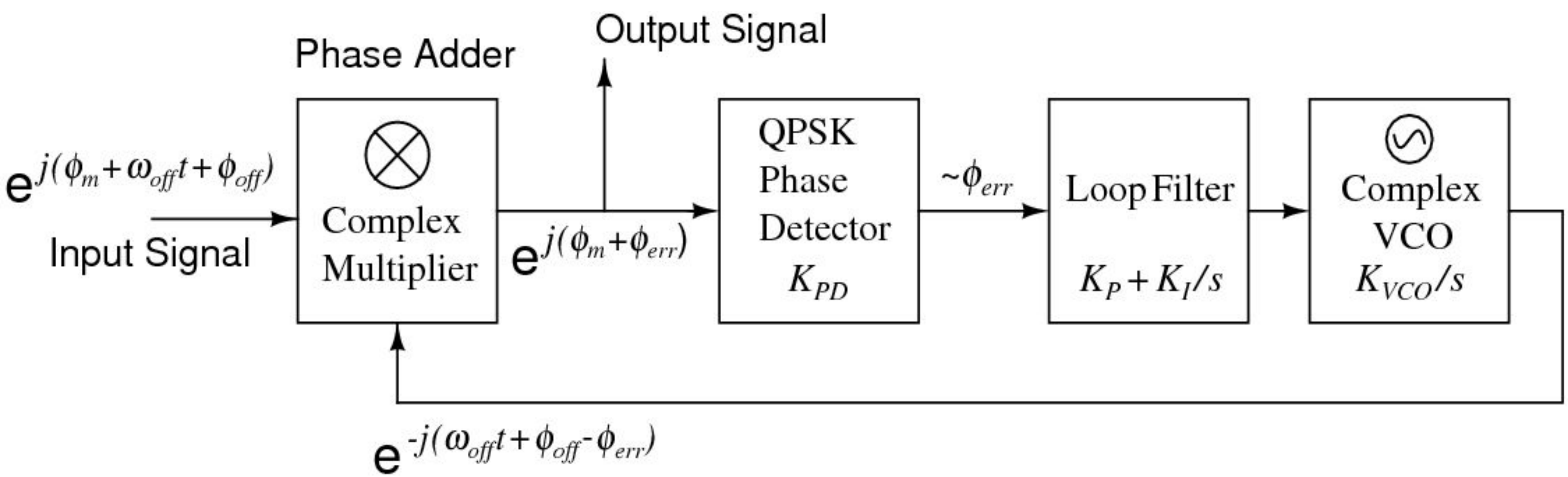
  FF taps: [1.0001,-1]; FB taps: [-1,1]; Old Style of taps: True

- Give this output to a Complex VCO with sensitivity of about -5 (why negative sign?) Multiply the VCO output to the signal with carrier offset that was generated in Part 1.

- Now observe the constellation after Polyphase Clock Sync (make sure the carrier offset was 0.1 Hz or less). It should stop rotating and settle to the desired constellation diagram. Vary the phase offset and determine upto what frequencies we can get back the original constellation.

- Observe roughly the time taken for the constellation to settle. Is it roughly equal to the settling time constant calculated by you?

Costas Loop


Phase Adder
Output Signal
$e^{j(\phi_m+\omega_{off}t+\phi_{off})}$
Input Signal
Complex
Multiplier
$e^{j(\phi_m+\phi_{err})}$
QPSK
Phase
Detector
K_{PD}
$\sim\phi_{err}$
Loop Filter
K_P+K_I/s
Complex
VCO
K_{VCO}/s
$e^{-j(\omega_{off}t+\phi_{off}-\phi_{err})}$