

# Homework 4 – Acquisition and Tracking

Due: 4/11/2022

1. Write a controller (matlab or Simulink) for a simple  $1/s$  plant. Assume a unit step input for the reference,  $r(t)$ .
  - a) What type of controller did you use. Provide the Gain Margin, Phase Margin, closed-loop eigenvalues, and steady state error
  - b) What is the steady state error if the reference  $r(t)$  is a unit ramp input
  - c) Redesign the controller to track the ramp input and repeat part (a).
2. Take the sampled 100 Hz sine wave (generate\_signal(1)) from the website (sampled at 1 MHz, i.e.  $T_s=1e-6$ )
  - a) Develop a simple (1 Hz) PLL to track the phase of the signal. Provide plots of phase, phase error, as well as the estimated signal vs. true signal.
  - b) Double the PLL bandwidth and repeat part a.
  - c) Determine the true frequency and repeat part a.
  - d) Modify your PLL to work from the unknown frequency and repeat part a assuming 10 Hz signal.Optional: Develop a FLL and provide the output of the FLL.
3. Modify your PLL from problem #3 to operate as a Costas loop filter. Take the 88 second data (generate\_signal(2)) sampled at 1 MHz and decode the data message on the 100 Hz sinusoid using your Costas loop filter. The data bits are 1 second wide and are comprised of 8 bit ascii characters. The following functions will be of use:

```
>>bit_string= dec2bin('A')
>>value1=bin2dec(bit_string)
>>decoded_eight_bits=[0 1 0 0 0 0 0 1];
>>value2=bin2dec('01000001')
>>decoded_bit_string=int2str(decoded_eight_bits)
>>value3=bin2dec(decoded_bit_string)
>>index=[1 2 3];
>>fprintf('Value #%d is %c\n',index(1),value1);
>>fprintf('Value #%d is %c\n',index(2),value2);
>>fprintf('Value #%d is %c\n',index(3),value3);
>>vector_of_values=[value value2 value];
>>fprintf('The text is: %s\n',vector_of_values);
```

4. Develop a simple DLL to phase align the sequence shown below with the digital signal (generate\_signal(3)) provided on the website. The sequence is sampled at 10 samples per chip. Plot the delay vs. time

Sequence = [1,-1,-1,-1,1,-1,1,1]

5. Combine your Costas Loop filter from problem #3 and your DLL from problem #4 to decode the data signal (generate\_signal(4))
6. Take the PRN code for SV #4 or #7 (i.e. from HW #3) and upsample it such that there are 16 samples at each chip (i.e. the length of this vector will be 1023\*16 long).
  - a) Show the autocorrelation calculation from -5 chips to +5 chips in 1/16 chip increments.
  - b) Repeat with noise ( $\sigma=0.2$ ) added to the non-shifted signal

Recall that the autocorrelation function can be calculated as

$$R(\tau) = x^T x(\tau) = \sum_1^{16 \times 1023} x(k)x(k + \tau)$$

Note that you must roll  $x(\tau)$  around when shifting it, i.e.  $x(16 \times 1023 + 10) = x(10)$

You will need code that does the upsample and code shift for Lab #4 (as well as the problem below).

7. Write your own acquisition software to acquire a single satellite from the IFEN IF data file. You can write a serial or parallel search algorithm. Provide a plot of the acquisition plane (Code and Doppler) and provide the code phase and Doppler results. The C/A (Gold) codes for several satellites in view are on the website.

*Optional: Search for all 32 satellites and show which ones are "in view."*

*Optional: Compare a 1 ms and 10 ms acquisition data length*