Sercan Turkmen 2557739 12.11.2017
Mina Ghobrial 2557713

Report of MATLAB Exercise 2

Task 1.

$$[X_k] = [R_1 cos(\tfrac{2\pi}{72}k) + C_1] + [u_k]$$
$$[Y_k] = [R_2 sin(\tfrac{2\pi}{72}k) + C_2] + [v_k]$$

Eq. 1.1 *Observation matrix and* θ

We will use LS-estimator and our signal model in the problem is given in the ***eq. 1.1***. From the book we know that for a given signal model, we should find the observation matrix first to use linear least squares estimator.

$$H = \begin{bmatrix} \cos(\frac{2\pi}{72} \times 1) & 0 & 1 & 0 \\ \cos(\frac{2\pi}{72} \times 2) & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \cos(\frac{2\pi}{72} \times k) & 0 & 1 & 0 \\ 0 & \sin(\frac{2\pi}{72} \times 1) & 0 & 1 \\ 0 & \sin(\frac{2\pi}{72} \times 2) & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \sin(\frac{2\pi}{72} \times k) & 0 & 1 \end{bmatrix}, \theta = \begin{bmatrix} R_1 \\ R_2 \\ C_1 \\ C_2 \end{bmatrix}$$

Eq. 1.2 *Observation matrix and* θ

$$\hat{\theta} = (H^T H)' H^T X$$

When we implemented the code using the LSE formula in matlab, we got the resulting graph shown in the ***Fig. 1.1***. Here we can clearly see that the trajectory of the celestial overlaps with *B*. Hence we conclude that a and B will collide.

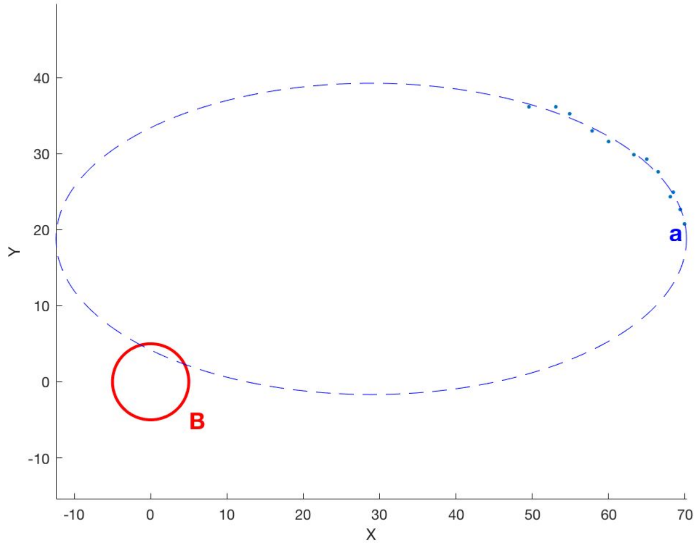

Fig. 1.1 *Estimated trajectory of* ***a***

Task 2.

Given: 200,000 Complex (Inphase and Quadrature) samples, corresponds to real measurement of pulsed signal.

- We need to automatically set the threshold value for power samples [dBm], which will be used in separating the power samples into:

$$H_0 : \text{noise only}$$
$$H_1 : \text{Signal + Noise}$$

We used **eq. 7.4 [1]** to automatically set the threshold value

$$T(x) \; = \; \sum_{n=0}^{N-1} x^2[n] \; > \; \gamma'$$

Note: This method gives excellent threshold for this case, where the signal has high SNR, and the signal is sharp, narrow and deterministic.

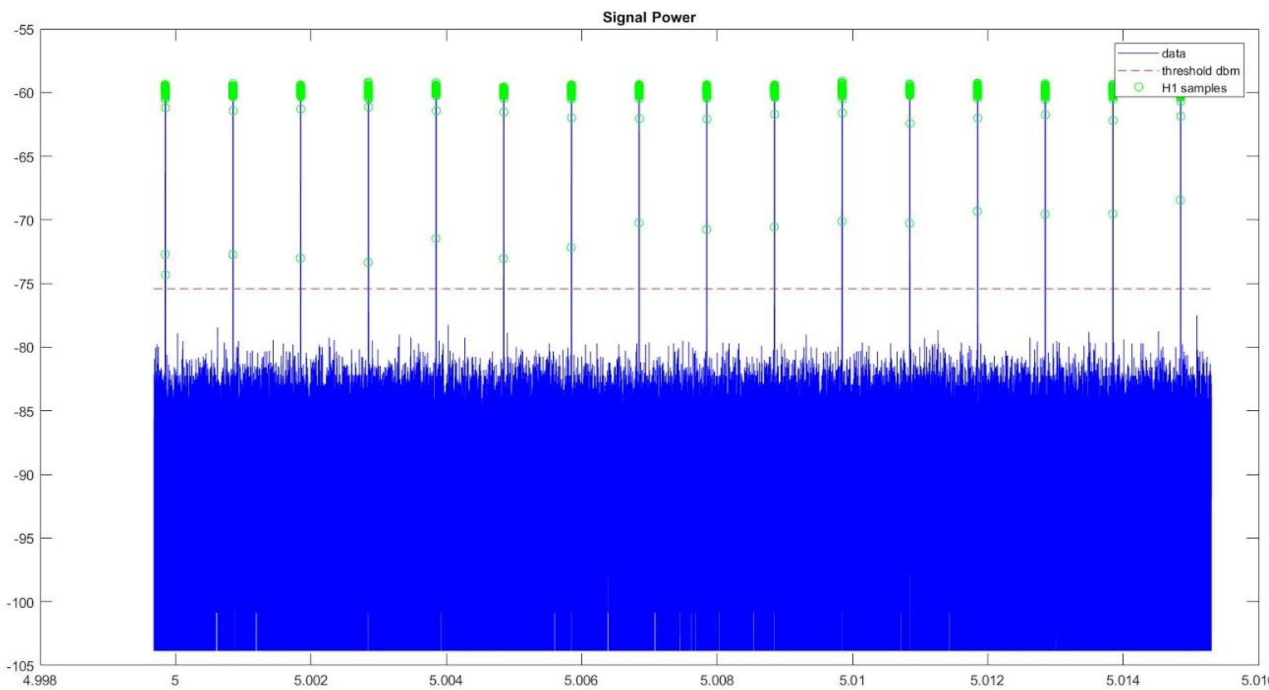

Fig. 2.1 *Data samples in dBm (blue), Threshold in dbm (red dashed line),* H_1 *samples detected (green circles)*

$$\gamma' \; \approx \; -75 \; dBm$$

After obtaining the threshold, it is used to split the signal samples into two portions H_0 and H_1 as stated earlier as shown in ***Fig. 2.1*** . Then we find the distribution of the noise so that we can calculate the probability of false alarm (P_{FA}).

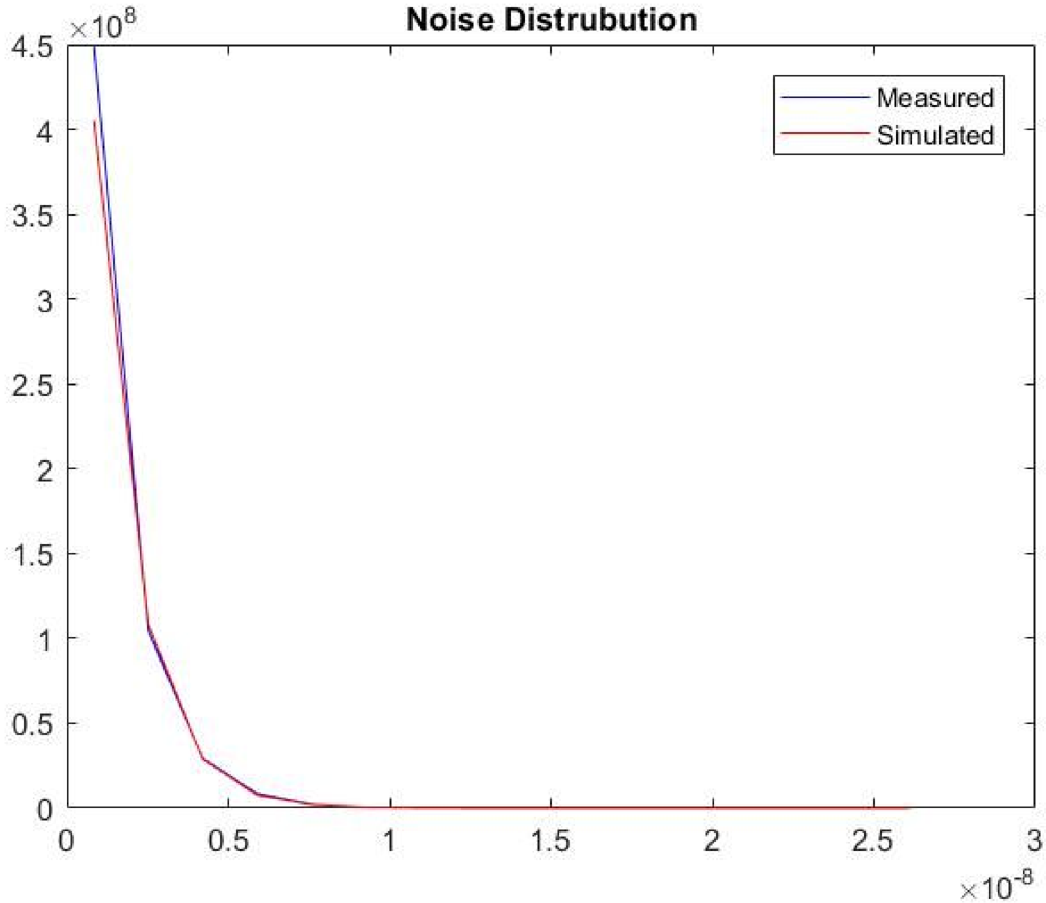

Fig. 2.2 *Noise follows exponential distribution, Measured (Blue), Simulated (Red)*

Using the previous information, probability of false alarm is calculated

$$P_{FA} = 1 - e^{-\lambda x} = 0$$

Since the probability of false alarm should be as low as possible, being zero is an optimal case, so it seems very reasonable P_{FA}. But it should be kept in mind that the data we have has high SNR so having $P_{FA} = 0$ doesn't lower the P_D of the test function. Otherwise having such P_{FA} would result in a very bad detection rate.

- As shown in ***Fig. 2.2***, the partitioning of the data into noise only samples (H_0) and signal + noise samples (H_1) seems very reasonable, because all the signals are being detected above threshold, while samples below threshold are noise only samples.
- When dealing with weak signals (low SNR), we have to set lower thresholds using methods other than the one used specifically in this problem. For example, we might use the RMS of the power samples, which will produce lower threshold, then multiply this threshold by a scalar value, then get slightly higher P_{FA}. When dealing with weak signals in real life we should select a reasonable P_{FA} that suits our needs. Unlike this example, weak signals will have trade off relation in the P_{FA} and P_D values.
- Using MATLAB, we calculated the period signal period. We can see from ***Fig. 2.3*** that the period is nearly constant with very low variance (`Pulse Period mean=0.00093759 variance=6.2422e-08`) N.B. the zero sample here might be a error in our matlab code when we

choose values for matlab's `findpeaks` method. And for the pulse width (duration), we can see from ***Fig. 2.4*** that duration variance is very small and it is nearly constant (`Pulse Width mean=1.9706e-05 variance=6.0176e-11`). According to the these low variances, we can comment that the signal generator is quite accurate.

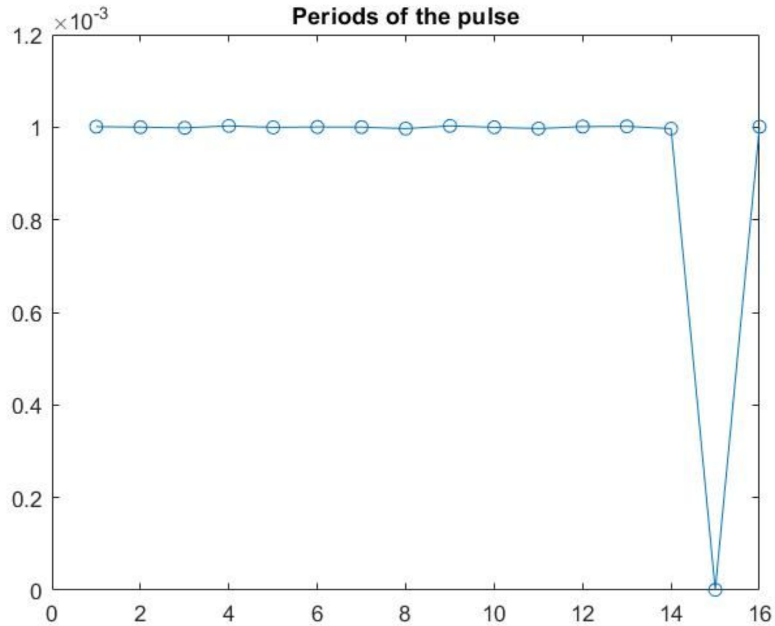

Fig. 2.3 *Period which the signal pulses are repeated*

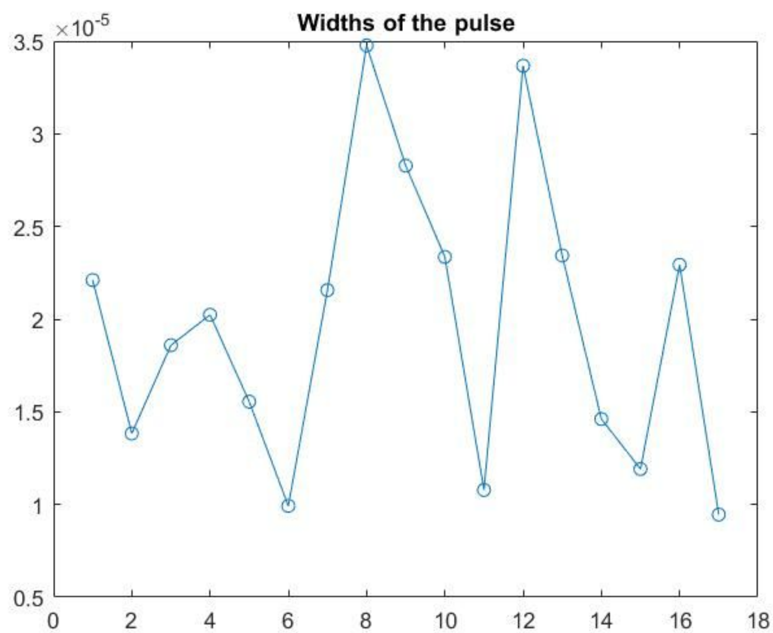

Fig. 2.4 *Duration of the pulses*

- In case of very weak signals, if we have knowledge about the signal then even though the signal is almost buried under the noise we might be able to detect some patterns. Another idea is that if we have a version of the signal in high SNR than we can use machine learning algorithms by adding simulated noise to the signal and training the system to recognize small similarities.

References.

[1] Fundamentals of Statistical Signal Processing Volume 2 Detection Theory, Steven M. Kay.