POLITECNICO
MILANO 1863

REMOTE SENSING FOR EARTH OBSERVATION AND SURVEILLANCE: HOMEWORK ASSIGNEMENT 1

Aufiero Luca

Course of Remote Sensing for Earth Observation and Surveillance
Space Engineering
Academic Year 2019-2020

Contents

List of Figures

List of Tables

Chapter 1

Problem 1: Estimation in SAR Interferometry

Synthetic Aperture Radar (SAR) systems are designed to acquire images of the same scene under the same geometry every few days. This feature is used to deliver accurate information about the displacement of the Earth's surface using SAR Interferometry (InSAR). The information about surface displacement is embedded in the interferometric phase $\phi(n)$, which represents the phase difference of the SAR image acquired at time t_n with respect to the one acquired at time t_0. A simple, yet sufficiently detailed, model of the interferometric phase is:

$$\phi(n) = k_z(n)z + k_v(n)v + w(n) - w(0) \tag{1.1}$$

where:

- $\phi(n)$ is the phase difference with respect to the image acquired at time t_0,
- n=1,2,...N,
- z represents terrain topography expressed in m,
- v represents surface displacement rate (or velocity) expressed in mm/day,
- $k_z(n)$ is the height-to-phase conversion factor,
- $k_v(n)$ is the velocity-to-phase conversion factor,
- w(n) is the phase noise, modeled as a stochastic process such that: $E[w(n)] = 0, E[(w(n))^2] = \sigma_w^2, E[w(m)w(n)] = 0$ if $n \neq m$

NOTE: as SARs produce images, all of the terms above are to be intended as pixels at a particular location (x,y). Hence: $\phi = \phi(x, y, n), w = w(x, y, n), z = z(x, y), v = v(x, y)$. The only exception is represented by the two conversion factors, which only depend on n: $k_z = k_z(n)$ and $k_v = k_v(n)$. DATA: a sample data-set is found in subfolder /HW1 in the Drop Box folder of the course. The data-set includes a set of N = 4 phase images, the conversion factors k_z and k_v, the noise variance σ_w^2, and two images representing the true topography and the true displacement rate. The role of phase noise can be modeled as in 1.1. ASSIGNMENT:

1. Propose an algorithm to estimate terrain topography and displacement rate (z,v) at any pixel location from the set of interferometric phases $\phi(x, y, n)$.

2. Evaluate estimation accuracy on a theoretical basis.
3. Apply the estimation algorithm on the data-set, observe the results and discuss them.
4. Assess estimation accuracy empirically based on the knowledge of true topography and displacement rate.

1.1 Propose an algorithm to estimate terrain topography and displacement rate (z,v) at any pixel location from the set of interferometric phases ϕ(x,y,n)

In the data there are four realizations of the phase, that are four 650x1000 matrices containing the pixels of the observed images. Our objective is to estimate the real terrain topography z and displacement rate v at any pixel location extracting them from the phase images using an estimator. As we can see from the model of the interferometric phase 1.1, it linearly depends on z and v. So to estimate them we have to solve the linear problem

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} = \begin{bmatrix} k_z & k_v \\ . & . \\ . & . \\ . & . \end{bmatrix} \begin{bmatrix} z \\ v \end{bmatrix} + \begin{bmatrix} w_1 - w_0 \\ w_2 - w_0 \\ . \\ . \end{bmatrix} \tag{1.2}$$

$$y = Ax + w \tag{1.3}$$

As we have a linear problem, and the phase noise w(n) is modeled as a stochastic process such that $E[w(n)] = 0, E[(w(n))^2] = \sigma_w^2, E[w(m)w(n)] = 0$ if $n \neq m$, so that it has a normal-like distribution, the most accurate estimator we can use is the Best Linear Unbiased Estimator BLUE. In general with this estimator we obtain

$$\hat{x}_{BLUE} = B^T y \tag{1.4}$$

with

$$B = C_w^{-1} A (A^T C_w^{-1} A)^{-1} \tag{1.5}$$

and so

$$B^T = (A^T C_w^{-1} A)^{-1} A^T C_w^{-1} \tag{1.6}$$

In this problem:

$$A = \begin{bmatrix} k_z & k_v \\ . & . \\ . & . \\ . & . \end{bmatrix} \tag{1.7}$$

and because of the previous consideration about the phase noise w, its covariance matrix

$$C_w = \sigma_w^2 \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix} \tag{1.8}$$

with $\sigma_w^2 = 3$ given in the data.

1.2 Evaluate estimation accuracy on a theoretical basis

In a BLUE estimator the error

$$\epsilon = \hat{x}_{BLUE} - x \tag{1.9}$$

can be estimated as:

$$E[\epsilon\epsilon^T] = B^T C_w B = (A^T C_w^{-1} A)^{-1} = \begin{bmatrix} \sigma^2_{\hat{x}_1} & 0 & 0 & 0 \\ 0 & \sigma^2_{\hat{x}_2} & 0 & 0 \\ 0 & 0 & . & 0 \\ 0 & 0 & 0 & . \end{bmatrix} \tag{1.10}$$

As BLUE is an unbiased estimator, we can compute the theoretical root-mean-square error (RMSE) as the square root of the variance, that is the standard deviation σ:

$$theoreticalRMSE = \sqrt{\begin{bmatrix} \sigma^2_{\hat{x}_1} & 0 & 0 & 0 \\ 0 & \sigma^2_{\hat{x}_2} & 0 & 0 \\ 0 & 0 & . & 0 \\ 0 & 0 & 0 & . \end{bmatrix}} = \sqrt{B^T C_w B} = \sqrt{(A^T C_w^{-1} A)^{-1}} \tag{1.11}$$

In this way we can find the theoretical root-mean-square error of the BLUE estimation of terrain topography z and displacement rate v:

theoretical RMSE of z estimation	35.6985
theoretical RMSE of v estimation	0.3892

Table 1.1: Theoretical root-mean-square error of z and v estimation with BLUE

1.3 Apply the estimation algorithm on the data-set, observe the results and discuss them

Applying the BLUE estimator on the 4 phase images we can find an estimated solution of the linear problem 1.2, and so we estimate the terrain topography z and the displacement rate v. Now we can plot the estimated values we obtained and compare them with the real values we have in the data.

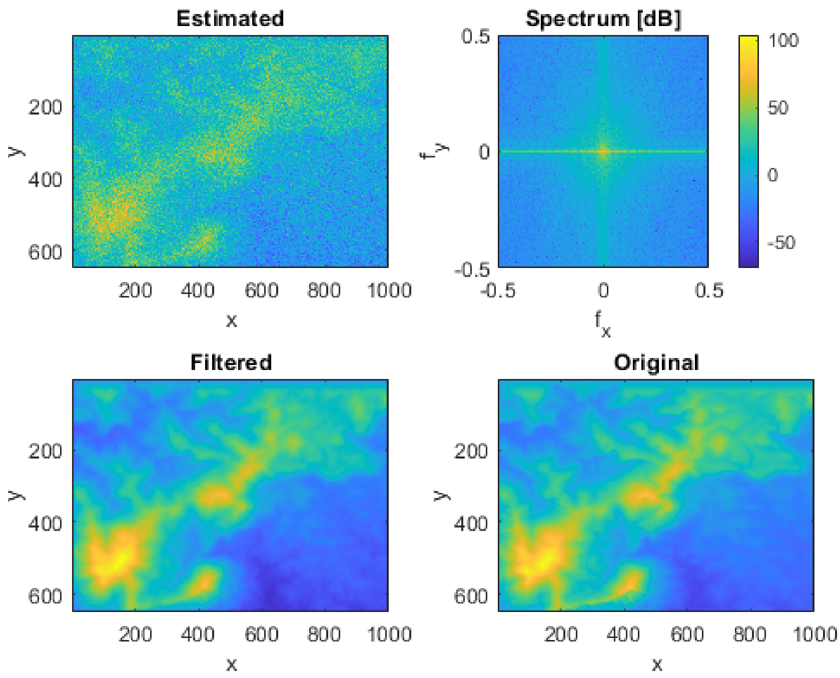

Figure 1.1: Estimated values, spectrum, filtered estimated values and real values of terrain topography z

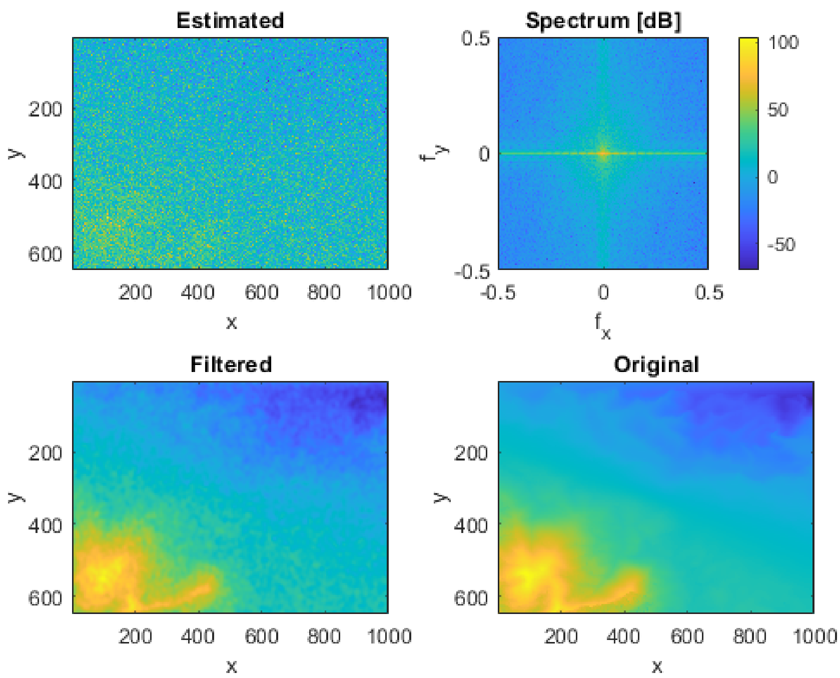

Figure 1.2: Estimated values, spectrum, filtered estimated values and real values of surface displacement rate v

In both cases it's clear that the estimated values give a good approximated representation of the original images, but they aren't very accurate. We expected a considerable estimation error, as stated in the table 1.1. To increase the accuracy of the estimated images, making them more similar to the original ones, we can filter the estimated values. In this case we have to use two filters as it's a 2D representation: one filter for the row values and the other one for the column values. We used two equal filters: a 30th order lowpass digital filter with a cutoff frequency of 0.05 Hz. This frequancy can't be too low, not to cut useful frequancies losing useful data to reconstruct the real images, but it can't even be too high, because in this way the reconstruction wouldn't be accurate. As we can see from the spectrum plot, most of the data is in a very narrow bandwidth, so a cutoff frequancy of 0.05 Hz makes the filters work well.

1.4 Assess estimation accuracy empirically based on the knowledge of true topography and displacement rate

As in the data we have real values of terrain topography and surface displacement rate, now we can compute the real estimation error and compare it with the theoretical estimation accuracy evaluated in the paragraph 1.2. We can compute the real root-mean-square error in z and v estimation:

$$realRMSEz = \sqrt{mean((z_{estimated} - z_{real})^2)}; \tag{1.12}$$

$$realRMSEv = \sqrt{mean((v_{estimated} - v_{real})^2)}; \tag{1.13}$$

In this way we obtain two matrixes with the RMSE of every single pixel. We can average them and we obtain:

real RMSE of z estimation	51.3646
real RMSE of v estimation	0.8921

Table 1.2: Real root-mean-square error of z and v estimation with BLUE

In both estimations the real RMSE is higher then the theoretical one. This could happen if the real characteristics of the phase noise are different with respect to the w model we considered ($E[w(n)] = 0, E[(w(n))^2] = \sigma_w^2, E[w(m)w(n)] = 0$ if $n \neq m$), and so the BLUE doesn't work perfectly as it would do with a perfectly normal distrubuted noise.

Chapter 2

Problem 2: target detection using a passive radar

Passive Radars are implemented by capturing the echoes scattered by targets on the ground in response to signals emitted by any large-bandwidth source of opportunity. A suitable mathematical model of the received signal is:

$$y(t) = x(t) + \sum_{p=1}^{P} A_p x(t - \tau_p) \tag{2.1}$$

where:

- $x(t)$ is the direct signal emitted by the large bandwidth source and received by the Radar
- $A_p x(t - \tau_p)$ is the echo produced by the p-th ground target, with τ_p the travel time relative to the direct path and A_p an amplitude factor.

The direct signal $x(t)$ can be modeled as a complex stochastic process, zero mean, and white in the frequency band $\left(-\frac{B}{2}, \frac{B}{2}\right)$. DATA: The data-set includes: the received signal $y(t)$, the direct signal $x(t)$, and the information about the sampling interval. ASSIGNMENT:

1. Propose an algorithm to detect the number of ground targets and estimate the associated parameters τ_p and A_p assuming that the observed data includes both $y(t)$ and $x(t)$.
2. Repeat point 1 assuming that the observed data includes only $y(t)$.
3. Apply the estimation algorithm on the data-set, observe the results and discuss them.
4. What is the minimum temporal separation $\tau_2 - \tau_1$ required to resolve two targets? Note: two targets are said to be resolved if they appear as two distinct peaks.
5. As an effort to reduce the data volume, both $y(t)$ and $x(t)$ are shortened by retaining only the first 800 samples. What is the impact of this choice on the results?

2.1 Propose an algorithm to detect the number of ground targets and estimate the associated parameters τ_p and A_p assuming that the observed data includes both $y(t)$ and $x(t)$

If we have both the direct signal $x(t)$ and the received signal $y(t)$, we can detect the number of ground targets and estimate the required parameters τ_p and A_p calculating the cross-correlation between the two signals we have: R_{yx}. Then we can find the number of ground targets, the amplitude factors A_p and the travel times relative to the direct path τ_p analyzing R_{yx} graph. Before computing the cross-correlation we must find some parameters: the observation time is equal to the product between the number of samples of the signals N_t and the sampling interval dt:

$$T_{obs} = N_t * dt \tag{2.2}$$

The input time axis and the number of frequency samples to compute the Fourier transform are:

$$t = (0 : N_t - 1) * dt \tag{2.3}$$

$$N_f = 2^{ceil(log_2(N_t))} \tag{2.4}$$

Now we can compute the Fourier transform using the Matlab discrete Fourier Transform function dft:

$$[X, f] = dft(x, t, N_f) \tag{2.5}$$

For dimensional consistency we multiply the obtained Fourier transform X with a scale factor equal to dt:

$$X = X * dt \tag{2.6}$$

We apply equations 2.6 and 2.7 also to y, obtaining its Fourier transform Y, and then we can compute the cross spectrum and the cross correlation with the inverse Fourier transform:

$$S_{yx} = \frac{1}{T_{obs}} * Y * X^* \tag{2.7}$$

$$R_{yx} = idft(S_{yx}, f, \Delta t) \tag{2.8}$$

The signal is complex, but we analyze only its real part. This generates an error equal to the maximum of the imaginary part of the cross-correlation in absolute value, but it's very small, approximately $1.7654 * 10^{-9}$. Also in this case we multiply R_{yx} by a scale factor, equal to $df * N_f$, with $df = f(2) - f(1)$, for dimensional consistency (as we use Matlab built-in function $ifft$, that divides the result by the number of frequency points N_f). Now we can plot the cross correlation and analyze the graph to obtain the number of ground targets and the parameters τ_p and A_p.

2.2 Repeat point 1 assuming that the observed data includes only $y(t)$

If the direct signal $x(t)$ isn't available, and we have only the received signal $y(t)$, we can still find the parameters τ_p and A_p by calculating the power spectral density of $y(t)$ and then its auto-correlation with the inverse Fourier transform:

$$S_y = (1/T_{obs}) * abs(Y)^2 \tag{2.9}$$

$$R_y = idft(S_y, f, \Delta t) \tag{2.10}$$

Also in these computations we have an error as we consider only the real part of the signal; it's equal to the maximum of the imaginary part of the auto-correlation in absolute value but also this error is very small, approximately $3.927 * 10^{-9}$. For dimensional consistency we multiply R_y by the same scale factor we used before, and then we can analyze its plot to check the parameters we obtain in this way.

2.3 Apply the estimation algorithm on the data-set, observe the results and discuss them

First we analyze the cross-correlation:

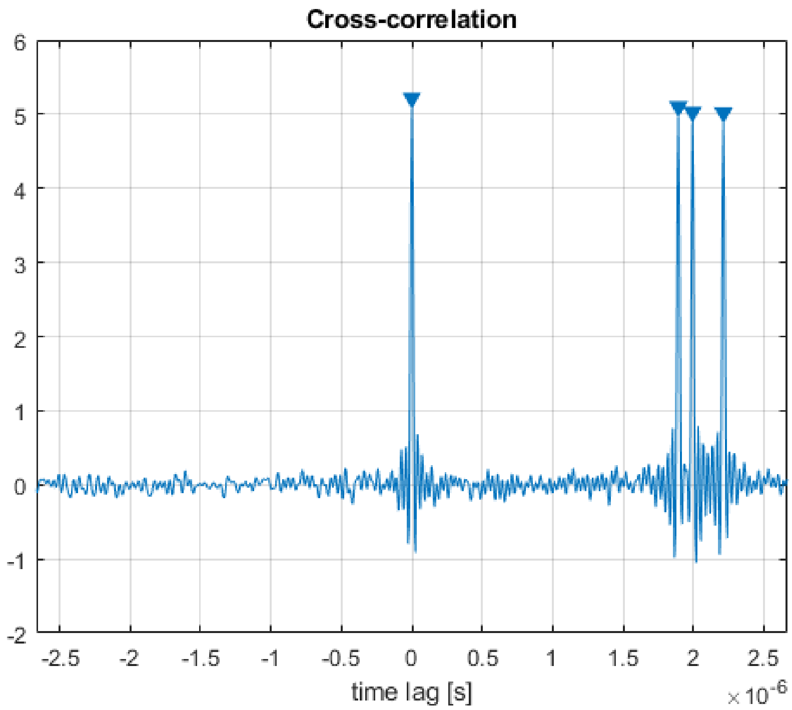

Figure 2.1: Cross Correlation

To ease the writing of the next equations, we introduce s:

$$s = \sum_{p=1}^{P} A_p x(t - \tau_p) \tag{2.11}$$

so that

$$y = x + s \tag{2.12}$$

To interprete the cross-correlation graph, we can analyze the equation which describes it:

$$R_{yx} = E[(x+s)*x^*] = E[x*x^*] + E[s*x^*] = R_x + R_{sx} \tag{2.13}$$

with

$$R_{sx} = E[(\sum_{p=1}^{P} A_p x(t-\tau_p)) * x^*] = \sum_{p=1}^{P} A_p R_x(\tau_p) \tag{2.14}$$

R_x represents the central peak, while R_{sx} represents the 3 peaks we can see on the right side of the graph. Those 3 peaks are the ground targets, and from the graph we can also obtain the parameters A_p (equal to the height of the ground targets peaks divided by the height of the central peak) and τ_p:

Ap_1	0.9777
Ap_2	0.9632
Ap_3	0.9613
τp_1	$0.1893 * 10^{-5} s$
τp_2	$0.1993 * 10^{-5} s$
τp_3	$0.2213 * 10^{-5} s$

Table 2.1: A_p and τ_p values obtained from the cross-correlation graph

Now we analyze the auto-correlation of the received signal $y(t)$:

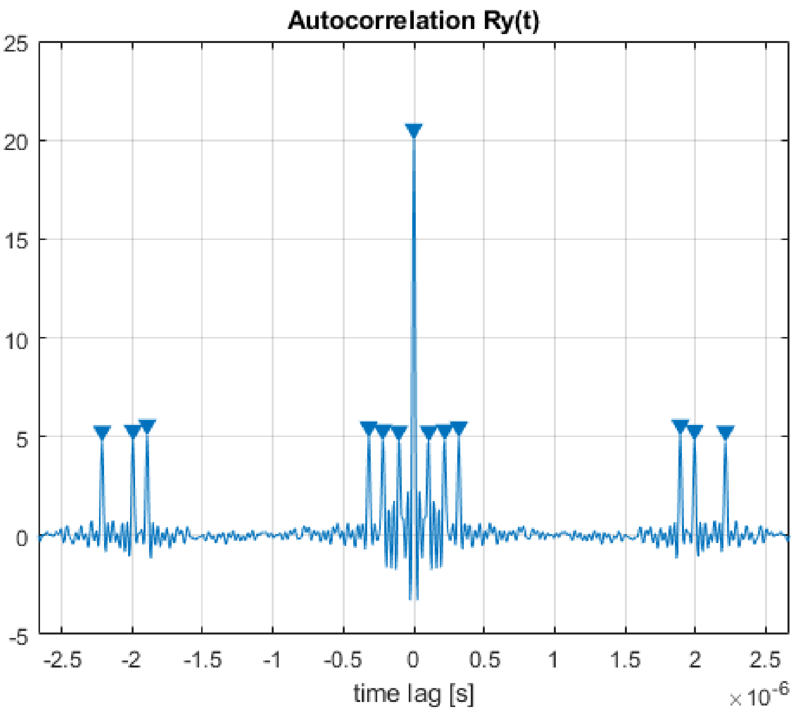

Figure 2.2: Auto Correlation

We can understand this graph better if we analyze R_y equation:

$$R_{yy} = [(x+s)*(x+s)^*] = R_x + R_{xs} + R_{sx} + R_s \tag{2.15}$$

with:

$$R_{sx} = E[(\sum_{p=1}^{P} A_p x(t-\tau_p)) * x^*] = \sum_{p=1}^{P} A_p R_x(\tau_p) \tag{2.16}$$

$$R_{xs} = E[x * (\sum_{p=1}^{P} A_p x(t-\tau_p))^*] = \sum_{p=1}^{P} A_p R_x(-\tau_p) \tag{2.17}$$

$$R_s = E[(\sum_{i=1}^{P} A_i x(t-\tau_i)) * (\sum_{j=1}^{P} A_j x^*(t-\tau_j))] = \sum_{i=1}^{P}\sum_{j=1}^{P} A_i A_j(\tau_i - \tau_j) \tag{2.18}$$

If $i = j$

$$R_s = A^2 R_x \tag{2.19}$$

these terms are added to the central peak; while if $i \neq j$

$$R_s = \sum_{i=1}^{P}\sum_{j=1}^{P} A_i A_j(\tau_i - \tau_j) \tag{2.20}$$

these terms correspond to the six peaks close to the central peak of the graph, 3 on its right, and 3 simmetrically on its left. R_x is another term that contributes to the central peak, R_{sx} are the 3 peaks far from the center on the right side of the graph and R_{xs} are the same peaks on the left side. Writing the equation more in detail, we find that in this case the amplitudes A_p are equal to the height of the peaks divided by $R_x(0)$, that is the central peak minus the 3 peaks closer to it, either the ones on the right or the ones on the left (as they have the same height), as the contribute R_s gives to the central peak is the same one it gives to the three close peaks. We find the delays τ_p in the same way as we did before.

Ap_1	0.9201
Ap_2	0.8690
Ap_3	0.8611
τp_1	$0.1893 * 10^{-5}$
τp_2	$0.1993 * 10^{-5}$
τp_3	$0.2213 * 10^{-5}$

Table 2.2: A_p and τ_p values obtained from the auto-correlation graph

As the direct signal $x(t)$ is obviously more precise than the received one $y(t)$, the noise is more effective in $y(t)$ auto-correlation than in the cross-correlation, and so we find that now the amplitudes are slightly lower. While, as expected, the delays where we find the ground targets are always the same.

2.4 What is the minimum temporal separation $\tau_2 - \tau_1$ required to resolve two targets?

We can check the minimum temporal separation required to resolve two targets on both graphs. It's equal to the time between the start and the ending of a peak, approximately equal to $3 * 10^{-8}$ seconds. With a lower temporal separation the peaks would overlap, generating a constructive or destructive interference, and so we wouldn't be able to detect them, or we would detect them with different parameters.

2.5 As an effort to reduce the data volume, both $y(t)$ and $x(t)$ are shortened by retaining only the first 800 samples. What is the impact of this choice on the results?

Considering only the first 800 samples out of 19609, we obtain different graphs of the cross correlation and the autocorrelation:

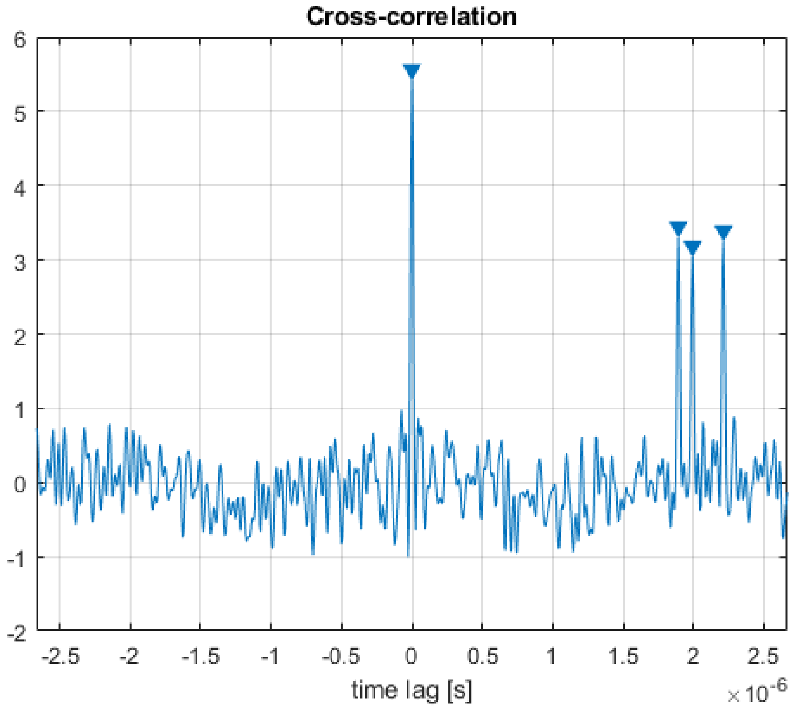

Figure 2.3: Cross Correlation with the first 800 samples

From the cross-correlation graph we can still see the 3 ground targets at the same delays, but they have a lower amplitude, of respectively 0.6093, 0.5607 and 0.5997. Far from the center of the graph the lower amount of data affects the results: the noise gains importance and it lowers the amplitude of the peaks.

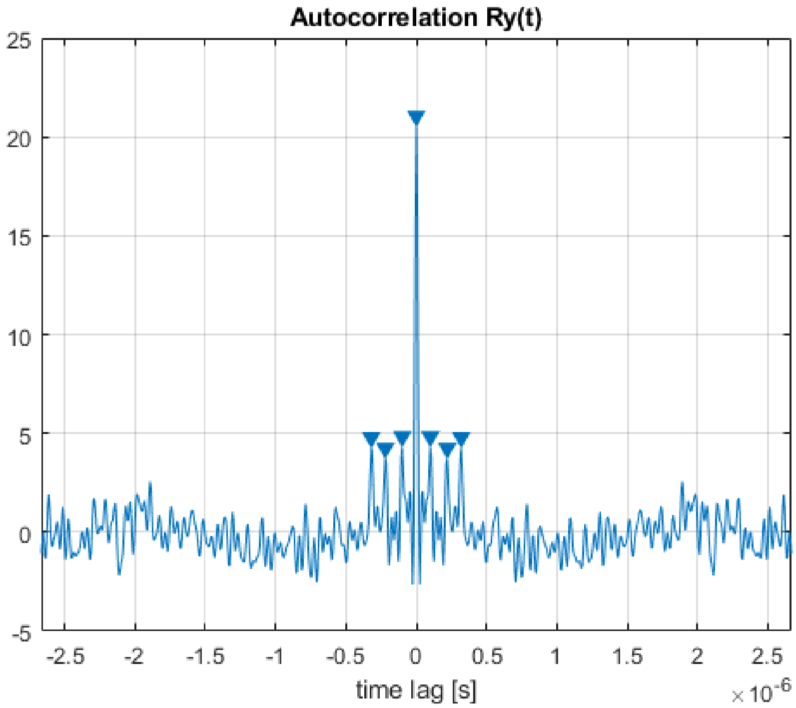

Figure 2.4: Auto Correlation with the first 800 samples

From the auto-correlation graph we can't see the ground targets peaks anymore, while the peaks closer to the center still appear. This happens because the lower number of data only slightly affects the results closer to the center of the graph, that is close to the time 0, while it affects a lot the graph at higher temporal distance from the center. Working with the equations we could still find the presence of the ground targets, even if with lower accuracy. The minimum temporal separation required to resolve two targets remains the same, of approximately $3 * 10^{-8}$ seconds.