

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA DE
TECNOLOGÍAS Y SERVICIOS DE
TELECOMUNICACIÓN
TRABAJO FIN DE GRADO**

**ANÁLISIS Y DISEÑO DE ALGORITMOS DE
SEGUIMIENTO DE TRAYECTORIAS
RADIALES EN RADARES CON
PROCESADO RANGE-DOPPLER**

**PABLO BIELZA LÓPEZ-MANTEROLA
2022**

GRADO EN INGENIERIA DE TECNOLOGIAS Y SERVICIOS DE TELECOMUNICACION

TRABAJO FIN DE GRADO

Título: Análisis y diseño de algoritmos de seguimiento de trayectorias radiales en radares con procesado Range-Doppler

Autor: D. Pablo Bielza López-Manterola

Tutor: D. José Tomás González Partida

Ponente: D. Félix Pérez Martínez

Departamento: SEÑALES, SISTEMAS Y RADIOPCOMUNICACIONES

MIEMBROS DEL TRIBUNAL

Presidente: D.

Vocal: D.

Secretario: D.

Suplente: D.

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de:

Madrid, a de de 2022

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA DE TECNOLOGÍAS Y
SERVICIOS DE TELECOMUNICACIÓN
TRABAJO FIN DE GRADO**

**ANÁLISIS Y DISEÑO DE ALGORITMOS DE
SEGUIMIENTO DE TRAYECTORIAS
RADIALES EN RADARES CON PROCESADO
RANGE-DOPPLER**

**PABLO BIELZA LÓPEZ-MANTEROLA
2022**

RESUMEN

Los sistemas radar son esenciales en el mundo de la defensa aérea. Este TFG busca dar soporte en presentes y futuros proyectos radar de INDRA.

Se ha desarrollado un software basado en MATLAB que simula la respuesta de dos de los algoritmos de seguimiento de blancos más utilizados: el filtro de Kalman para velocidades casi constantes (*Nearly Constant Velocity*) y el filtro de Kalman adaptado a blancos con aceleración casi constante (*Nearly Constant Acceleration*), todo ello para un radar LFM-CW (Linear Frequency Modulated-Continuous Wave).

Primero, se va a implementar el procesado Distancia-Doppler (*Range-Doppler*), que nos dará la matriz con las distancias a las que se encuentran los posibles blancos y sus velocidades. Despues, se aplicará el algoritmo CA-CFAR con el fin de minimizar las falsas detecciones, además de realizar una interpolación gaussiana que mejorará la precisión de la medida. Seguidamente, las detecciones serán asociadas a *plots* y éstos serán asociados a trazas mediante el uso de la distancia de *Mahalanobis*, que tendrá en cuenta la matriz de covarianzas correspondiente al blanco. La matriz de covarianzas de cada blanco será la que se extraiga a la salida del filtro de Kalman que se haya seleccionado, cuya tarea es estimar dónde se encontrarán los blancos con trazas activas en la siguiente vuelta. Esta estimación la hará comparando lo que había predicho con respecto a lo que se ha detectado. Si el filtro es capaz de seguir correctamente al blanco, lo que ocurrirá es que la matriz de covarianzas convergerá a unos valores, al igual que el error de la traza, que deberá converger idealmente a cero.

SUMMARY

Radar systems are essential in the sector of air defence. This project seeks to support present and future INDRA's radar projects.

A MATLAB-based software has been developed to simulate the response of two of the most widely used target tracking algorithms: the Kalman filter for *Nearly Constant Velocity* targets and the Kalman filter adapted to targets with *Nearly Constant Acceleration*, all this for a Linear Frequency Modulated-Continuous Wave (LFM-CW) radar.

First, the Range-Doppler processing will be implemented, which will give us the matrix with the distances at which the possible targets are located and their velocities. Then, the CA-CFAR algorithm will be applied in order to minimize false detections, in addition to performing a Gaussian interpolation that will improve the accuracy of the measurement. Next, the detections will be associated to plots and these will be associated to traces by using the Mahalanobis distance, which will take into account the covariance matrix corresponding to the target. The covariance matrix of each target will be the one extracted at the output of the selected Kalman filter, whose task is to estimate where the targets with active traces will be found in the next round. This estimation will be done by comparing what was predicted with respect the current detection. If the filter is able to correctly track the target, the covariance matrix will converge to some values, as will the error of the trace, which should ideally converge to zero.

PALABRAS CLAVE

LFM-CW, Matriz Distancia-Doppler, PRF, CFAR, seguimiento de blancos, plot-to-track, Kalman Nearly Constant Velocity, Kalman Nearly Constant Acceleration

KEYWORDS

LFM-CW, Range-Doppler Matrix, PRF, CFAR, target tracking, plot-to-track, Kalman Nearly Constant Velocity, Kalman Nearly Constant Acceleration

GLOSARY

LFM-CW: Linear Frequency Modulated-Continuous Wave

DFT: Discrete Fourier Transform

FFT: Fast Fourier Transform

PRI: Pulse Repetition Interval

PRF: Pulse Repetition Frequency

RDM: Range-Doppler Matrix

CA-CFAR: Cell Averaging-Constant False Alarm Ratio

CUT: Cell Under Test

NCV: Nearly Constant Velocity

NCA: Nearly Constant Acceleration

Agradecimientos

Quiero dar las gracias a Félix por confiar en mí para representar a la ETSIT como becario en INDRA y a Joseto por su ayuda y por todo lo que me ha enseñado a lo largo del TFG.

Muchas gracias a Jose por prestarme su tiempo y por rebajarse hasta el nivel de casi redondear $g = 10 \text{ m/s}^2$.

Por último, gracias a Labola y a mi familia, especialmente al núcleo; gracias por escuchar atentamente mis delirios.

INDEX

1. INTRODUCTION & OBJECTIVES	1
1.1. Introduction.....	1
1.2. Objetives	1
2. SIGNAL PROCESSING.....	2
2.1. LFM-CW Waveform	2
2.2. Pulse-Doppler processing	6
3. DATA PROCESSING.....	8
3.1. CFAR	8
3.1.1. Target Masking	11
3.2. Detection-to-Plot.....	12
3.2.1. Interpolation.....	13
4. TARGET TRACKING.....	15
4.1. Plot-to-Track	15
4.2. Track Update.....	16
4.3. Filtering & Prediction	17
4.3.1. Kalman Nearly Constant Velocity	20
4.3.2. Kalman Nearly Constant Acceleration.....	21
5. PARAMETER SELECTION & RESULTS.....	22
5.1. Parameter Selection	22
5.2. Motion Models Used.....	24
5.2.1. Linear Motion	24
5.2.2. Uniformly Accelerated Linear Motion.....	24
5.2.3. Direct Fire	24
5.2.4. Polynomial Motion	25
5.3. Results.....	25
5.3.1. Target with Constant Velocity	26
5.3.2. Target with Constant Acceleration	28
5.3.3. RPG-7	30
5.3.4. Loitering Drone.....	32
5.3.5. Manoeuvring Target.....	34
5.3.6. Artillery Target	36
5.3.7. Loitering Drone and RPG-7	38
6. CONCLUSIONS & FUTURE LINES OF ACTION	40
6.1. Conclusions.....	40
6.2. Future lines of action	40

7. REFERENCES	41
ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES	42
A.1 INTRODUCCIÓN	42
A.2 DESCRIPCIÓN DE IMPACTOS RELEVANTES RELACIONADOS CON EL PROYECTO	42
A.3 ANÁLISIS DETALLADO DE ALGUNO DE LOS PRINCIPALES IMPACTOS	42
A.4 Conclusiones	42
ANEXO B: PRESUPUESTO ECONÓMICO	43
ANEXO C: SIMULADOR.....	44

1. INTRODUCTION & OBJECTIVES

1.1. INTRODUCTION

The history of RADAR (**R**Adio **D**etection **A**nd **R**anging) started [1] with experiments by Heinrich Hertz, [2] who proved the existence of the electromagnetic waves predicted by James Clerk Maxwell's equations. However, it was not until the early 20th century that systems able to use these principles were becoming widely available, and it was German inventor Christian Hülsmeyer who first used them to build a simple ship detection device intended to help avoid collisions in fog.

The real booster of the RADAR technology was the World War II, in which this technology took part as a decisive factor for the victory of the Allies. [3] During World War I (1914-1918), airplanes played a relatively small role, being used mainly for reconnaissance. But as airplanes increased in size, range, and speed in the 1920s, it became clear that they would become major weapons in future wars. Bombing was the major concern. Airplanes might carry enormously destructive bombs, and there was little to prevent enemy aircraft from reaching a nation's cities. For example, the Battle of the Beams was a period early in the Second World War when bombers of the German Air Force used increasingly accurate systems of radio navigation for night bombing in the United Kingdom. British scientific intelligence at the Air Ministry fought back with a variety of their own increasingly effective means, involving jamming and deception signals.

[4] The Cold War drove mad some countries, especially USA and the Soviet Union. Both parties of the War developed and built intercontinental radars to prevent enemy bombings. Eventually, this bombing threat during WWII and the Cold War is what made radar technology to advance enormously by the hands of the different parties involved.

After the war, radar use was widened to numerous fields: civil aviation, marine navigation, radar guns for police, meteorology and even medicine. Key developments in the post-war period include the travelling wave tube as a way to produce large quantities of coherent microwaves, the development of signal delay systems that led to phased array radars, and ever-increasing frequencies that allow higher resolutions. Increases in signal processing capability due to the introduction of solid-state computers has also had a large impact on radar use. Another post-war development used to improve radar technology is the Kalman filter. [5] The Kalman filter, developed by Rudolf E. Kalman in 1960, is an algorithm for the estimation of non-observable state variables based on observable variables that may have some measurement error. Since this filter is a linear and optimal estimator, from the least-squares perspective, and due to its widespread use in problem solving, it became necessary to extend its use to nonlinear systems. The Kalman filter is better than other algorithms used for estimation due to the small room it needs for storage and its wide variety of uses. However, the impact on the environment surrounding it, errors from measuring equipment, and incorrect parameter selection usually cause system errors in real applications. Researchers have developed different variants or modifications in the last years. These modifications are aimed at solving the problems the algorithm presents due to the increase in the complexity of the equipment in which is applied, and also owing to the accuracy and efficiency that manufacturing, medicine elaboration, and navigation, among other processes, need nowadays.

1.2. OBJETIVES

This project seeks to achieve the simulation of a RADAR, including signal and data processing. The main objective is to compare two of the most used types (on this context) of the Kalman filter (*Nearly Constant Velocity* and *Nearly Constant Acceleration*). The goal is to acknowledge in which conditions one is better than the other. Moreover, radial threats with ambiguous velocity will be addressed.

2. SIGNAL PROCESSING

2.1. LFM-CW WAVEFORM

The transmitted signal by a high resolution FMCW radar consists in a continuous pulse train of frequency modulated signals, with period PRI. In our case (LFM-CW), the frequency modulation is linear [6][7].

$$S_{tx}(t) = \sqrt{2 \cdot P_t \cdot Z_0} \cdot \text{rect}\left(\frac{t}{T}\right) \cdot \sin\left(\omega_0 t + \pi \frac{B}{T} t^2\right) \quad (2.1)$$

Where ω_0 is the sweeping central pulse frequency, from now on the “pulse frequency carrier”, B is the sweeping bandwidth, T is the sweeping time, P_t is the transmitted power measured with a reference impedance Z_0 , and the function $\text{rect}()$ is defined by:

$$\text{rect}(x) = \begin{cases} 1 & |x| \leq \frac{1}{2}, \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

The received signal, produced by the reflection of the transmitted signal in a target at distance R_t of the radar, is an attenuated and delayed version:

$$S_{rx}(t) = \sqrt{2 \cdot P_t \cdot Z_0} \cdot \text{rect}\left(\frac{t-t_R}{T}\right) \cdot \sin\left(\omega_0(t-t_R) + \pi \frac{B}{T} (t-t_R)^2\right) \quad (2.3)$$

Where P_r represents the power received by the radar due to that specific target, and t_R being the delay caused by the round-trip of the signal:

$$t_R = \frac{2R_t}{c} \quad (2.4)$$

Where c is the speed of light in that medium.

If we use a receptor based on a mixer, the received signal mixes with the original one, obtaining the so-called beat signal:

$$S_{tx}(t) \times S_{rx}(t) = K \cdot \sqrt{\sigma} \cdot \text{rect}\left(\frac{t-t_R}{T-t_R}\right) \cdot \cos\left(2\pi t_R \frac{B}{T} t + \Phi_1\right) - \\ - K \cdot \sqrt{\sigma} \cdot \text{rect}\left(\frac{t-t_R}{T-t_R}\right) \cdot \cos(2\omega_0 t + \Phi_2(t)) \quad (2.5)$$

Where $\Phi_1 = \omega_0 t_R - \pi t_R^2 \frac{B}{T} = \text{cte}$ and $\Phi_2(t) = 2\pi \frac{B}{T} t^2 - 2\pi t_R \frac{B}{T} t - \omega_0 t_R + \pi t_R^2 \frac{B}{T}$, σ is the radar section of the target and K is a constant dependent on the system and the propagation losses.

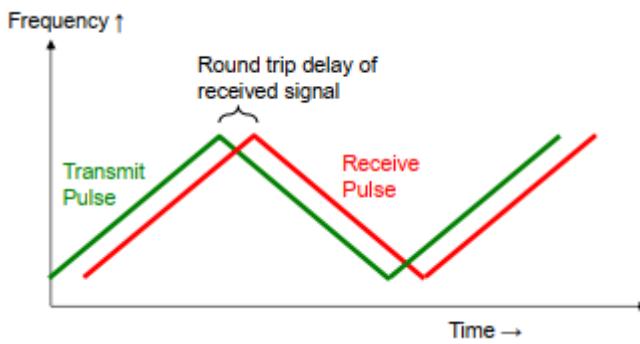


Figure 1: FMCW transmitted and received waveforms

The important summand is the first one, thus we eliminate the second one with a low-pass filter (it is a high frequency component). What we obtain is a sinusoid which frequency is directly proportional to the target's distance. Its spectrum corresponds to a spectral line at frequency f_b (**beat** frequency):

$$f_b = \frac{2R_t B}{c T} \quad (2.6)$$

To measure the distance of the target, we could use n narrow band-pass filters, centered at the expected frequencies. Depending in which filter the target was detected, we could know its distance. We can do this filter implementation in a very efficient way by sampling the signal and applying an n point discrete Fourier transform (DFT).

Ideally, the Fourier transform of a beat would be a spectral delta whose width would be zero. As the time of integration cannot be infinite, what we have in the real world is the windowed version of the signal, causing the DFT to be the famous *sinc* function:

$$\text{rect}\left(\frac{t}{T}\right) = \begin{cases} 1 & -\frac{T}{2} \leq t \leq \frac{T}{2} \\ 0 & \text{otherwise} \end{cases} \leftrightarrow \text{sinc}(f) = \frac{\sin(\pi \cdot T \cdot f)}{\pi \cdot f} \quad (2.7)$$

Being T lapse of time between the beginning of the transmission and the end of the reception.

This sinc has a main lobe with height $\frac{2}{T}$ [8]. The width of the main lobe of the sinc at 3 dB below the maximum is approximately $1/T$. Using the expression (2.6), we can demonstrate that the distance resolution of the systems at 3 dB is inversely proportional to the transmitted bandwidth:

$$\Delta f = \frac{1}{T} \rightarrow \Delta R = \frac{c}{2B} \quad (2.8)$$

The distance resolution is defined as the minimum separation between targets with the same level of reflexed power in order to be differentiated correctly by the radar.

The benefit of using frequency modulation is that we can obtain a distance resolution equivalent to the one earned using very short beats, with the advantages of having a long one. This technique is known as beat compression.

In our case, our beat has not the form of (Figure 1), in which we can see a ramp with positive slope followed by its opposite. What we have instead (Figure 2), is a single positive ramp with a decay time larger than zero to avoid the PLL loop to get loose.

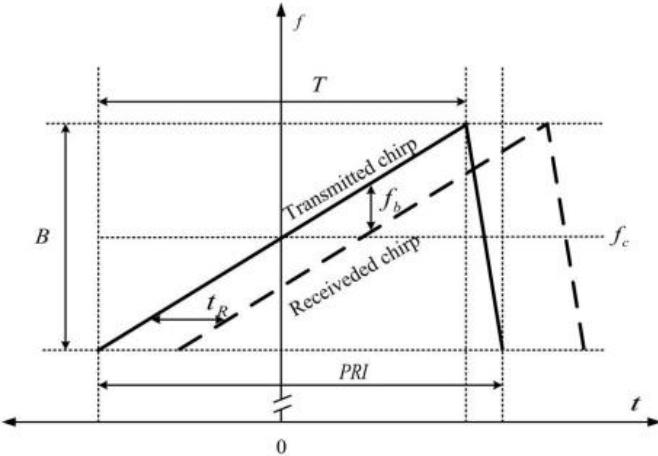


Figure 3: LFM-CW chirp waveform.

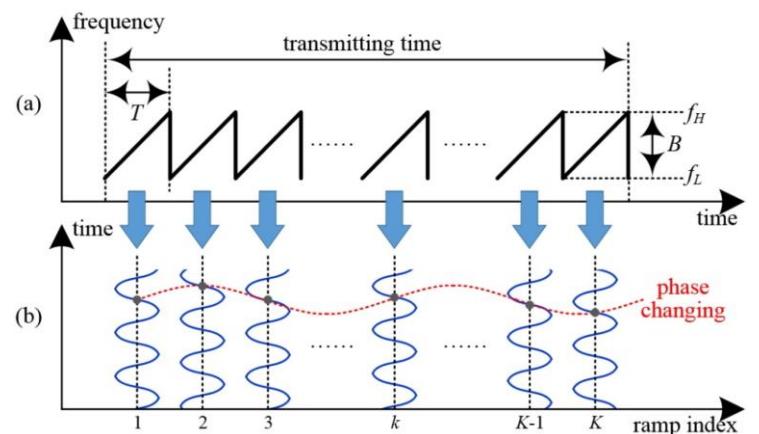


Figure 2: Basic concept of fast-ramp based 2D range-Doppler FMCW radar: (a) transmitted signal in the frequency-time domain; and (b) beat signal for a single moving target in ramp index domain. Here, T is modulation period, B is bandwidth, f_H is maximum instantaneous carrier frequency, f_L is the lowest carrier frequency, and K is the number of ramps. [9]

On the other hand, the Doppler frequency can be measured thanks to the phase shift experimented by the different ramps (**chirps**) of the sent burst. Suppose we have a moving target with a radial velocity v_r . The phase term $\Phi_1(\hat{t})$ of the equation (2.5) will change between ramps due to the target's movement

$$\Phi_1(\hat{t}) = \omega_0 t_R(\hat{t}) - \pi t_R^2(\hat{t}) \cdot \frac{B}{T} \approx \frac{4\pi}{\lambda} v_r \hat{t} + \frac{4\pi}{\lambda} R_{ini} \quad (2.9)$$

Where $t_R(\hat{t}) = \frac{2R_t(\hat{t})}{c} = \frac{2 \cdot v_r}{c} \hat{t} + \frac{2 \cdot R_{ini}}{c}$, \hat{t} is the so-called slow time (time of transmission of each ramp), λ the wavelength of the system at the central frequency and R_{ini} is the initial distance of the target when the measurement began. We have applied the *stop-and-go* hypothesis, which considers that there is no target movement between chirps.

That phase variance between chirps is approximately linear with the slow time, thus a frequency constant sinusoid is formed, proportional to the target's radial velocity, the denominated Doppler frequency:

$$f_D = \frac{1}{2\pi} \frac{d\Phi_1(\hat{t})}{d\hat{t}} = \frac{2 \cdot v_r}{\lambda} \quad (2.10)$$

The Doppler resolution is given by the inverse of the illumination time, due to the Fourier transform's windowing property. Taking the repetition interval between chirps, PRI, and the number of chirps, N, we can come to the expression of the Doppler resolution:

$$\Delta f_D = \frac{1}{N \cdot PRI} = \frac{1}{CPI} \quad (2.11)$$

It is important to notice that we are obtaining the Doppler history of the target with a sampling method, having a sampling frequency $PRF=1/PRI$. Thus, the maximum unambiguous Doppler frequency that can be measured is determined by the direct application of the Nyquist criterion:

$$|f_{D max}| \leq \frac{PRF}{2} \quad (2.12)$$

Applying both expressions to the equation (2.10), we can determine the resolution and the unambiguous margin of the velocities that can be measured:

$$\Delta v_r = \frac{\lambda}{2 \cdot N \cdot PRI} \quad (2.13)$$

$$|v_{r_{max}}| \leq \frac{PRF \cdot \lambda}{4} \quad (2.14)$$

$$R_{max} \leq \frac{c}{2 \cdot PRF} \quad (2.15)$$

On the other hand, the maximum unambiguous distance that can be measured by the radar comes by the equation (2.15). Joining equations (2.12) and (2.15), it is trivial that the PRF must be chosen looking for the best relation between ambiguity in distance and ambiguity in Doppler.

2.2. PULSE-DOPPLER PROCESSING

As mentioned before, we are filtering the signal by applying the DFT (FFT for faster response). DFT-based pulse-Doppler (from now on “range-Doppler”) processing attempts to separate target signals based on their particular Doppler shift [10]. The DFT is a matched filter to ideal constant radial velocity moving target signals, provided that the Doppler shift equals one of the DFT sample frequencies and the interference is white.

When the radar transmits M pulses in a dwell or CPI (Coherent Processing Interval), a set of range gates like those in Figure 4 will be measured for each pulse [11]. These are typically stored in memory as a two-dimensional matrix (**Range-Doppler Matrix**) of complex voltage samples as shown in Figure 5. The interval between samples in a row is the PRI, so the sampling rate in this dimension is the PRF. Because the PRF is much lower than the sampling rate in range, the pulse number axis is also called, as mentioned before, slow-time dimension. Each row of the matrix represents a series of measurements from the same range bin over M successive pulses. The total amount of MT (time of observation) represented by the RDM is CPI.

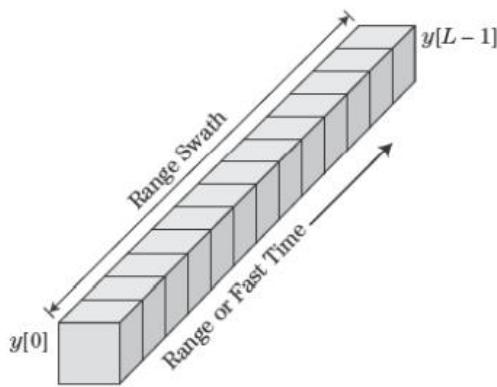


Figure 5: Single column (range dimension) from the Range-Doppler Matrix.

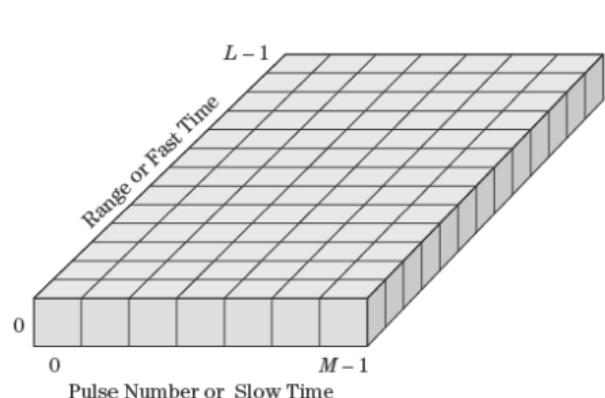


Figure 4: RDM in a single sampling period.

Range-Doppler processing differs from other Doppler processing in the filtering in the slow-time domain, which is replaced by explicit spectral analysis of the slow-time data for each range bin. In our case, we will be filtering both Doppler and range dimensions as depicted in Figure 6. Thus, in the RDM the dimensions are beat frequency (range data) and Doppler frequency. In the range-Doppler data, the energy from a moving target is separated from that of the clutter and competes only with the noise in the target's bin. In addition, an estimate of the Doppler shift (and thus, the radial velocity) of detected targets can be estimated based on the Doppler bin in which the detection occurs.

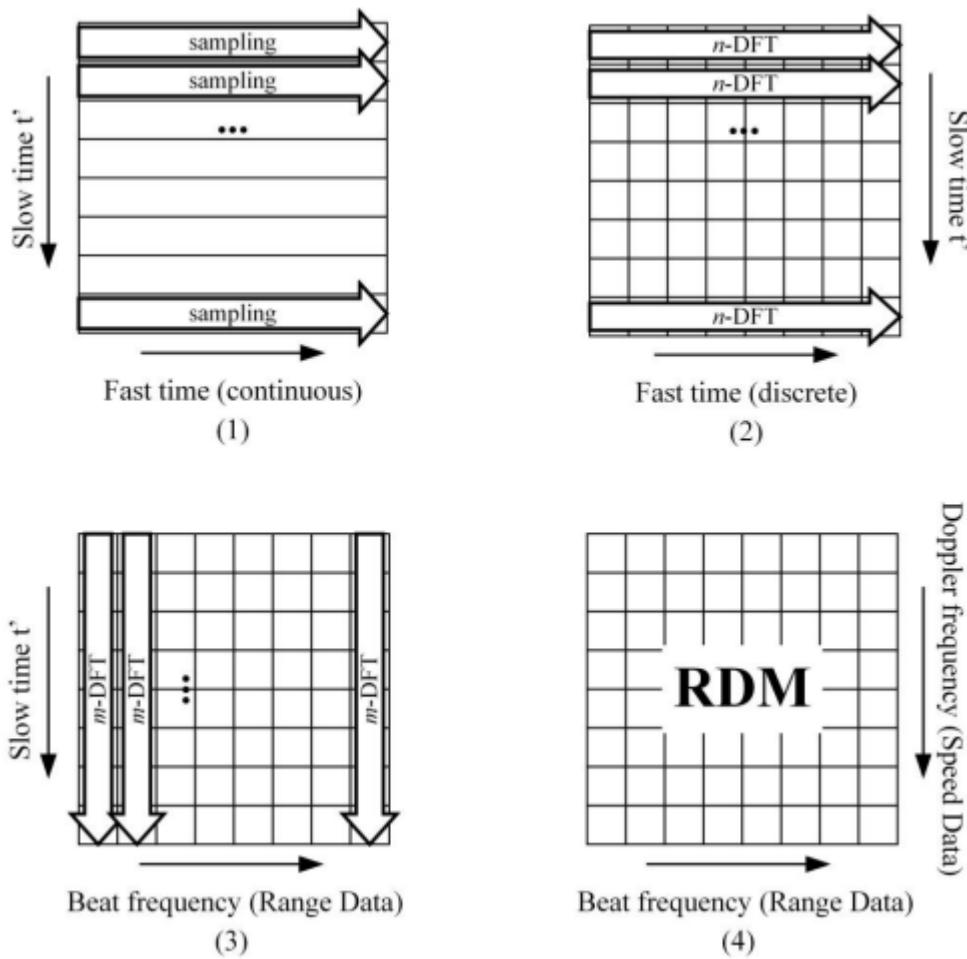


Figure 6: Discrete 2D spectrum procurement procedure. (1) Sampling of every beat signal along fast-time. (2) n -FFT application along fast-time. (3) m -FFT application along slow-time. (4) RDM with information of range and radial velocity in both dimensions respectively.

The advantages of range-Doppler processing are that it provides an estimate of radial velocity component of a moving target and that it provides a way to detect multiple targets (based on detections in multiple bins), provided they are separated enough in Doppler to be resolved. The chief disadvantages are greater computational complexity and longer dwell times due to the use of more pulses for Doppler measurements.

3. DATA PROCESSING

3.1. CFAR

If we stopped the processing here, we would have something like depicted in Figure 7. Visually we can barely distinguish 3 possible **blobs** (set of cells belonging to the same target). To improve this result, there are a set of techniques designed to provide predictable detection and false alarm in realistic interference scenarios, the so-called *constant false alarm rate* (CFAR) detection, also known as “adaptative threshold detection” or “automatic detection”. In this case, we will be using Cell Averaging-CFAR (CA-CFAR).

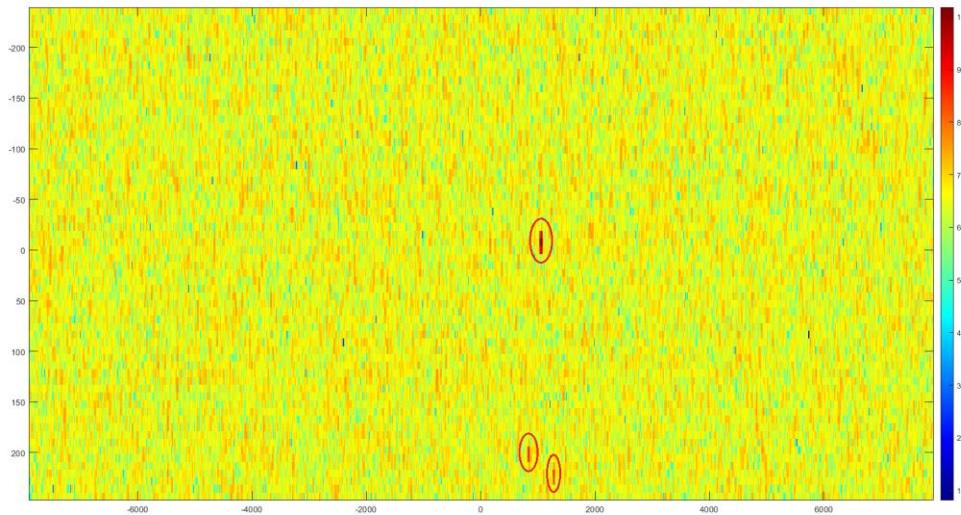


Figure 7: RDM imaging before CFAR.

CFAR techniques consist in windowing each cell of the matrix as schematized in Figure 8 and applying different operations. The data in grey cells to either side, representing data from ranges nearer and farther from the radar than the CUT (Cell Under Test), are averaged to estimate the noise parameter. These cells are called *reference cells*. The cross-hatched cells immediately adjacent to the CUT, called *guard cells*, are excluded from the average. The reason is that a target, if present, might straddle range cells. In that case, the energy in the cell adjacent to x_i would contain both interference and target energy and would therefore not be representative of the interference alone. The extra energy from the target would tend to raise the estimate of the interference parameter.

If the system range resolution is such that targets could extend over multiple range (or Doppler) cells, more than one guard cell would be skipped on each side of the CUT. If the number of guard cells used are lower than the needed, the masking issue is likely to happen. This issue is discussed in the subsequent section. The combined reference cells, guard cells, and cell under test are referred to as the CFAR window [10].

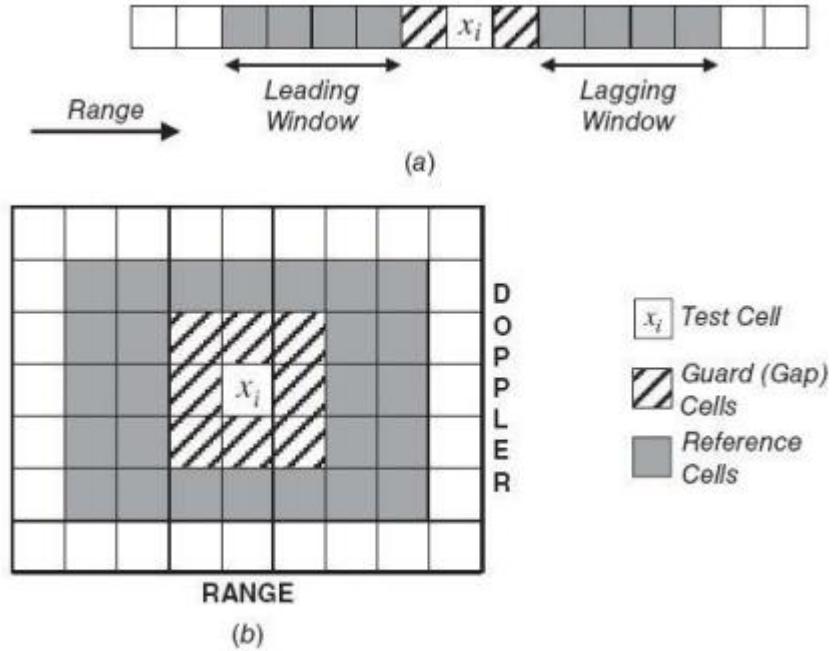


Figure 8: CFAR windows: (a) one-dimensional window for range only processor, (b) two-dimensional window for range-Doppler processor.

CA-CFAR derivation is presented next, considering these assertions:

1. The interference in the reference window and CUT is IID (Independent and Identically Distributed).
2. The interference is Rayleigh distributed voltage.
3. The rectifier is square law, and thus the interference at the output is exponentially distributed.
4. The mean of the interference power at the output of the rectifier is unknown and must be estimated from the samples in the reference window.
5. *The target is modeled as either Swerling 1 or 2 (Rayleigh voltage).

*The developed script allows using other models (Swerling 1 to 5)

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N z_n \quad (3.1)$$

The CA-CFAR applies the maximum likelihood estimator in (3.1) to the samples in the leading and lagging windows to form an estimate of the interference power [11]. The CA-CFAR threshold, T_{ca} , is defined by the product of the power estimate in equation (3.2) and the CA-CFAR constant, α_{ca} , or

$$T_{ca} = \alpha_{ca} \cdot \hat{\sigma}_i^2 \quad (3.2)$$

The CFAR constant is a function of both the desired P_{FA} and the number of samples in the reference window.

$$P_D(\hat{\sigma}_i^2) = \exp\left(\frac{-\alpha_{ca} \cdot \hat{\sigma}_i^2}{\sigma_i^2(1 + SINR)}\right) \quad (3.3)$$

The probability of detection defined in (3.3) is a function of the interference power estimate, which is a random variable, and thus $P_D(\hat{\sigma}_i^2)$ is also a random variable. To compute the average probability of detection, the PDF associated with the interference is needed. After some mathematical mix-up, the expression reduces to a simple closed-form equation

$$\bar{P}_D = \left[1 + \frac{\frac{\alpha_{ca}}{N}}{(1 + SINR)} \right]^{-N} \quad (3.4)$$

The average probability of false alarm is found by setting SINR equal to zero, corresponding to the interference-only condition (no target in the neighbor cells). This gives

$$\bar{P}_{FA} = \left[1 + \frac{\alpha_{ca}}{N} \right]^{-N} \quad (3.5)$$

Note that the average probability of false alarm is independent of the interference power. The detector thus achieves a constant false alarm rate without a priori knowledge of the interference power. This property is what defines a CFAR detector.

As a notational convenience, the overbar on P_D and P_{FA} will not be used in future references. The CA-CFAR constant is found by solving for α_{ca} in (3.5), giving

$$\alpha_{ca} = N \left[P_{FA}^{-1/N} - 1 \right] \quad (3.6)$$

This expression for the CFAR constant applies only to the cell-averaging case and should be associated with other CFAR algorithms.

After all the application of the CA-CFAR algorithm, we would obtain something like shown in Figure 9. Now, the potential targets are easily distinguished, as well as some additional false detections.

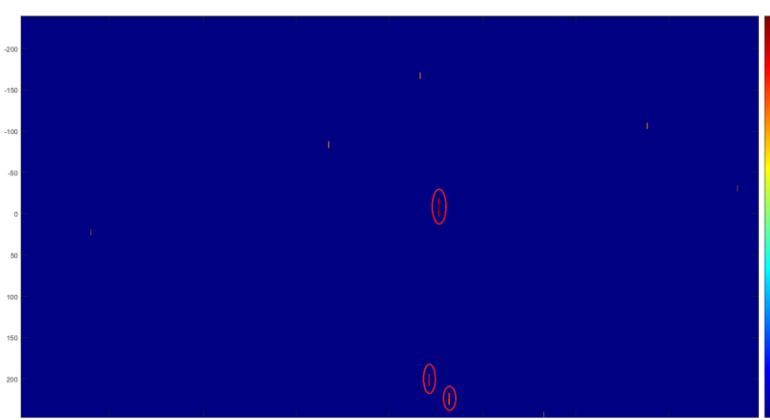


Figure 9: RDM imaging after CA-CFAR processing.

3.1.1. TARGET MASKING

Target masking occurs when target returns located within the reference window bias the threshold above the return in the CUT. Target masking may be partitioned into two categories: self-masking and mutual target masking [11].

Self-masking is associated with an extended target. An extended target is defined as one whose physical extent causes it to occupy more than one resolution cell. With a sample of an extended target located in the CUT, the remaining samples associated with the target bias the threshold if one or more samples lie within the reference window. The biased threshold may mask the presence of the extended target resulting in *self-masking* (Figures 10 and 11).

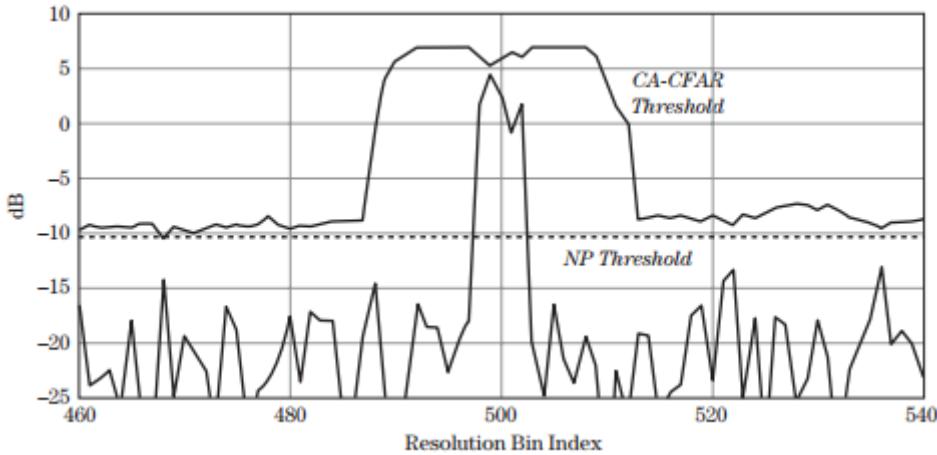
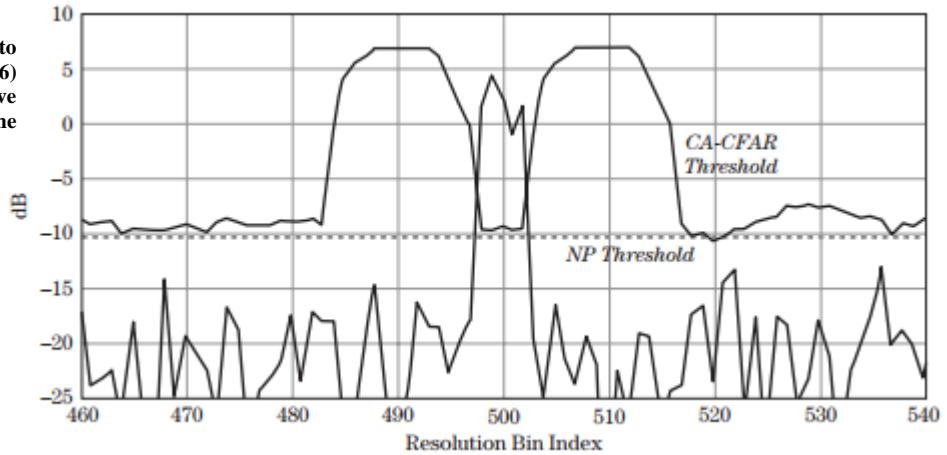


Figure 10: A CA-CFAR ($N = 20$) with no guard cells exhibits self-masking when an extended target consisting of 5 Rayleigh distributed scatterers is encountered.

Figure 11: Adding guard cells to the CA-CFAR ($N = 20$, $N_G=16$) produces an adaptative threshold that detects the extended target.



Mutual target masking occurs when target returns not associated with the target in the CUT fall within the reference window and bias the threshold. To illustrate mutual target masking, the returns from two Swerling 1 targets are plotted in Figure 12. The Swerling targets have an average SINR equal to 20 dB. The target on the right masks the presence of the target on the left. The target on the right is not masked. The bias associated with the smaller target was not sufficient to mask the presence of the larger target.

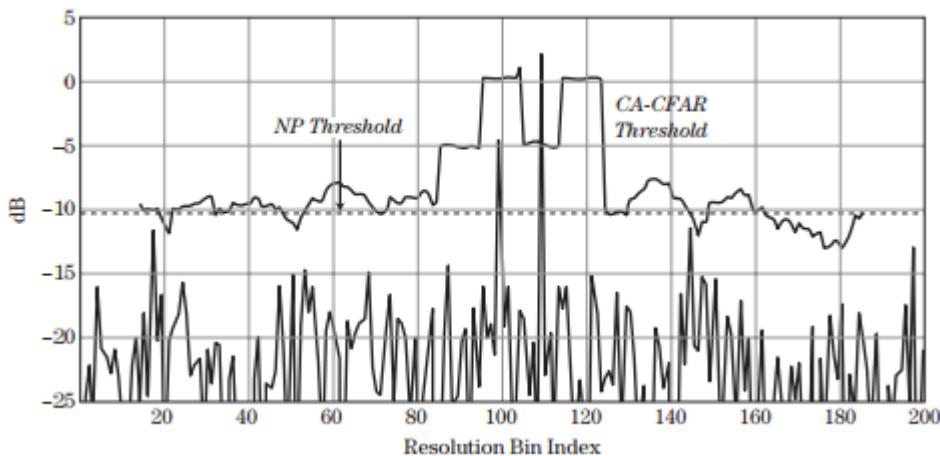


Figure 12: Mutual target masking can occur for closely-spaced targets: two Rayleigh distributed point targets separated by 10 resolution bins with average SINR = 20 dB. CA-CFAR has $N = 20$ and $N_G = 16$.

3.2. DETECTION-TO-PLOT

After the CFAR stage, what we have is a matrix with the resultant detections: each cell of the matrix with a value different than zero will be considered as a detection. As we know, a single target would most likely comprehend multiple cells near to each other (blob). A *plot* is obtained by relating different detections to their corresponding target. Proximity thresholds in both range and velocity dimensions must be set. These thresholds can be adjusted according to the characteristics of the targets we want to detect.

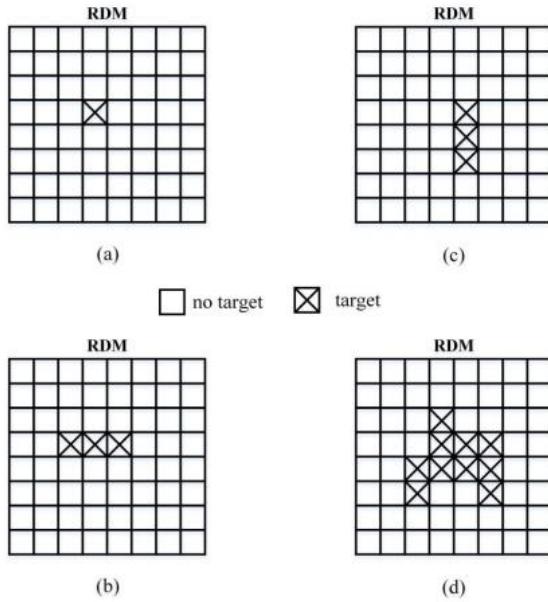


Figure 13: Examples of detections: a) Single cell detection. b) Range line detection. c) Doppler line detection. d) Blob.

To make feasible the posterior filter processing (see 4.3 section), we will have to unify the different cells of the resulting blobs into a one-cell plot. In our case, a gaussian interpolation will be applied before to improve accuracy.

3.2.1. INTERPOLATION

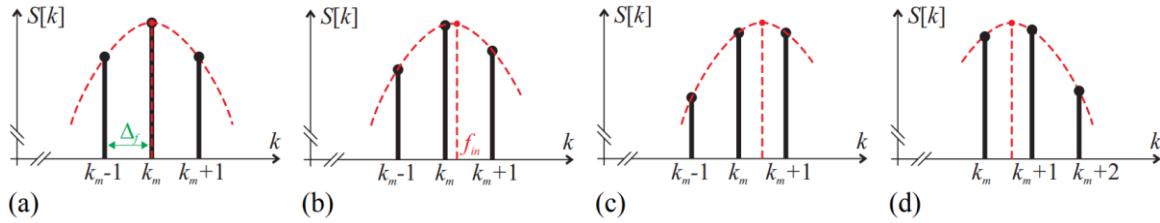


Figure 14: The principle of discrete spectrum interpolation.

The resolution can be increased considerably by discrete spectrum interpolation [12]. The method principle is sketched in Figure 14. When the frequency component f_{in} is located exactly on a local maximum of the discrete magnitude spectrum at bin k_m , f_{in} , can be calculated according to (3.7) with no error. This is the ideal case shown in Figure 14a. When f_{in} increases, the amplitude of bin k_{m-1} gets smaller and of bin k_{m+1} bigger, as presented in Figure 14b. The spectrum value $S[k_m]$ remains the biggest, until f_{in} is equidistant between bins k_m and k_{m+1} , see Figure 14c. In this case, determining f_{in} with (3.7) leads to the largest error (3.8). When f_{in} is increased further, as shown in Figure 14d, bin k_{m+1} becomes the biggest. For f_{in} smaller than $k_m \cdot \Delta_f$, the analysis is similar resulting in symmetrical cases.

$$f_{in} \cong k_m \Delta_f = k_m \frac{f_s}{N} = \frac{k_m}{NT_s} = \frac{k_m}{L} \quad (3.7)$$

$$\varepsilon = \frac{1}{2} \Delta_f = \frac{f_s}{2N} = \frac{1}{2NT_s} = \frac{1}{2L} \quad (3.8)$$

This example illustrates the fact that continuous frequency f_{in} can be estimated, even if it is located between two bins, by calculating the maximum abscissa of an interpolation curve of the discrete spectrum peak. This maximum can be located between $k_{m-1/2}$ and $k_{m+1/2}$, where k_m is the index of the biggest bin within the range of interest. If one needs to resolve the cases presented in Figure 14 and symmetrical ones, it is necessary for the interpolation to have at least **three node points**.

The peak continuous shape, i.e., the window spectrum main lobe, does not need to be accurately reproduced as long as the maximum abscissa of the interpolation shape follows the measured frequency. Therefore, a simple parabolic interpolation (PI) can improve the discrete spectrum frequency resolution by an important factor even when the shape of the window spectrum main lobe is quite far from a parabola.

Let $S[k]$ be the discrete magnitude spectrum of N samples of a signal $s(t)$ containing a sinusoidal component of frequency $\varphi_{in} = f_{in} \cdot L$, and k_m be the index of the biggest bin of the corresponding discrete spectrum peak. Index k_m can be found if the bin constitutes a local maximum within a given range. Fitting a parabola

$$S_p(\varphi) = a(\varphi - \varphi_m)^2 + h \quad (3.9)$$

through interpolation nodes $S[k_{m-1}]$, $S[k_m]$, $S[k_{m+1}]$ and finding the abscissa of the interpolation maximum φ_m , gives

$$\varphi_{in} \cong \varphi_m = k_m + \Delta_m = k_m + \frac{S[k_m + 1] - S[k_m - 1]}{2(2S[k_m] - S[k_m + 1] - S[k_m - 1])} \quad (3.10)$$

under condition $2S[k_m] > S[k_{m+1}] + S[k_{m-1}]$. The quantity Δ_m in (3.10) is the abscissa correction of the discrete spectrum maximum. It is a real number linking both, the discrete and continuous spectra, ranging from $-1/2$ when $S[k_{m-1}] = S[k_m]$, to $1/2$ for $S[k_{m+1}] = S[k_m]$.

Performance of an interpolation method can be characterized by the interpolation gain, defined as the ratio of the FFT frequency resolution (3.8) and the method maximum error

$$G = \frac{\varepsilon}{E_{max}} = \frac{\Delta_f}{2E_{max}} \quad (3.11)$$

The interpolation gain can be significantly improved by fitting a Gaussian shape to find the abscissa of the spectral peak maximum located between two discrete spectrum bins. Since a Gaussian curve

$$S_g(\varphi) = \exp(a'(\varphi - \varphi_m)^2 + h') \quad (3.12)$$

is a parabola in the logarithmic scale, the Gaussian interpolation (GI) reduces to the PI on the natural logarithm of the magnitude spectrum. The GI can be derived from (3.10) using logarithmic spectrum values, i.e., the interpolation nodes $S[k_{m-1}]$, $S[k_m]$, $S[k_{m+1}]$ are replaced by natural logarithms $\ln(S[k_{m-1}])$, $\ln(S[k_m])$ and $\ln(S[k_{m+1}])$. Thus, after logarithm grouping, (3.10) becomes

$$\varphi_{in} \cong \varphi_m = k_m + \Delta_m = k_m + \frac{\ln\left(\frac{S[k_m + 1]}{S[k_m - 1]}\right)}{2\ln\left(\frac{S[k_m]^2}{S[k_m - 1]S[k_m + 1]}\right)} \quad (3.13)$$

Even though the GI obtains better results than PI, we must consider that it needs more computational power due to the application of the logarithm.

4. TARGET TRACKING

A track represents the belief that a target is present and has actually been detected by the radar. An automatic radar tracking system forms a track when enough radar plots are made in a believable enough pattern to indicate a target is in fact present (as opposed of succession of false alarms) and when enough time has passed to allow accurate calculation of the target's kinematic state – usually position and velocity (also acceleration in *Section 4.3.2*). Thus, the goal of tracking is to transform a (time-lapse) plot picture, consisting of target plots, false alarms, and clutter, into a track picture, consisting of tracks on real targets, occasional false tracks, and occasional deviations of track position from true target positions.

Our tracking process can be divided into three steps:

- i. Plot-to-track association, consisting of associating a new plot with an old track or creating a new one. (*Section 4.1*)
- ii. Updating the state of existing tracks and defining new ones (*Section 4.2*)
- iii. Filtering and prediction (*Section 4.3*).

4.1. PLOT-TO-TRACK

[13] The goal of plot-to-track association is to correctly assign radar detections to existing tracks so the track states can be correctly updated. The basis for assignment is a measure of how close together the plot and track are in terms of measurable parameters such as range, doppler, acceleration, angle, and, when available, target signature. As this radar has no azimuth nor elevation dissertation, we will be using only the first three parameters listed. The first step in the measurement-to-track association is validation of the candidate measurements. The measurements are compared with the predicted measurement based on the state estimate of the track. The comparison is achieved by computing the Mahalanobis distance between the plot and the predicted track (4.1) and taking the one with the lower value (Global Strongest Neighbor), which must also meet a certain gating threshold (gating). [11].

$$D_M = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \quad (4.1)$$

Where \vec{x} is the expected track, \vec{y} is the real detection and S is the covariance matrix of the target.

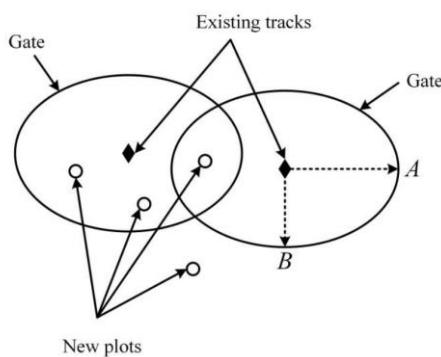


Figure 15: Plot-to-track association example.

4.2. TRACK UPDATE

Track update consists of defining the state of the different traces. The different states that a trace will have during its lifecycle will be the following:

- **Potential Trace.** The plots that have not been associated to any existing trace in the system will be catalogued as *potential traces*, that is, they will start the correlation process with other plots in successive explorations in order to form traces.

- **Tentative Trace.** This occurs when a number of plots below the threshold for establishing a trace in a series of scans correlates. If after a number of consecutive scans the criteria of number of plots that correlate with the *tentative trace* with sufficient quality are met, it will become *firm*, otherwise it will disappear from the list of traces.

- **Firm trace.** A tentative trace becomes *firm* when a number of plots have correlated above the established threshold in a series of scans with sufficient quality.

- **Extrapolated trace.** This state is set when a *firm trace* starts to have a loss of updates and still has a certain quality criterion to remain *firm*. We will use the number of updates as *firm* as threshold to determine this quality criterion.

- **Lost trace.** An *extrapolated trace* will be deleted when in a series of scans the trace has not been updated, and therefore is below the quality criterion.

To update the track update process will follow the next criteria:

1. The internal quality of the trace starts at the established initial quality threshold for a potential trace. The potential state only lasts for one lap.
2. The quality is incremented at each new update. The value by which the quality is increased is a parameter to be set.
3. If a tentative trace reaches the quality threshold to be firm, it becomes firm. The maximum quality value of a firm trace should also be set.
4. If a trace is not updated, its quality is reduced by a parameter to be set.
5. For firm traces, when their quality reaches a certain value to be determined or extrapolates $n+1$ consecutive times, being n the maximum number of permitted extrapolations for a firm trace, whichever comes first, they are deleted.
6. For tentative traces, when their quality reaches 0 or extrapolates $m+1$ consecutive times, being m the maximum number of permitted extrapolations for a tentative trace, whichever comes first, they are removed.

4.3. FILTERING & PREDICTION

Track filtering is the process of estimating the trajectory (i.e., position, velocity and possibly acceleration) of a track from measurements (plots). After this process, we will have an estimation of the future kinematics of the targets, in addition to the covariance matrix to be fed to the Mahalanobis algorithm.

[11] Let X_k denote the kinematic state vector at time t_k and Z_k denote the measurement at time t_k . The state estimate for t_k given measurements through t_j is denoted by $X_{k|j}$. Thus, when the state vector has a single subscript, it denotes a truth or modeled value, and when it has two subscripts, it represents an estimate. The state estimate $X_{k|k}$ is referred to as the filtered state estimate, while $X_{k|k-1}$ is referred to as the one-step prediction of the state. The estimate $X_{k|k+1}$ is referred to as the one-step smoothed estimate. Track filtering algorithms typically fall into one of two groups. The first group uses a parametric estimation approach that presumes a perfect model for the target motion and the time period over which the model is applied is limited to prevent distortion of the data. In this approach, the covariance of the state estimate (or track) will approach zero as more data are processed. As the covariance of the track error approaches zero, the gain for processing new data will approach zero. When this processing gain reaches a very small number, all future data will be essentially ignored. Thus, since all motion models are imperfect in practice, the time-period for which the perfect model is applied is limited to alleviate the distortion that results when new data are ignored. Least squares or maximum likelihood (ML) estimation are examples of the parametric approach to track filtering. The second group uses a stochastic state estimation approach that presumes an imperfect model for the target motion. In this approach, the target motion model includes a random process, and a perfect estimate of the kinematic state is not possible. In other words, the covariance of the track does not approach zero as the window of data expands. As the model is applied over an expanding window of measurements, the covariance of the track settles to a stable, slowly changing value. If the measurement rate is fixed and the data quality is uniform, the filter will achieve “steady-state” conditions in which the covariance is the same value after each measurement update. **The Kalman filter is an example of the stochastic state estimation approach.** One of the most critical items of a tracking system that supports any automatic decision system is track filter consistency. A track filter is considered to be consistent if the following three criteria are satisfied:

1. The state errors should be acceptable as zero mean and have magnitudes commensurate with the state covariance as yielded by the filter.
2. The innovations (i.e., residuals or difference between the measurement and predicted measurement) should be acceptable as zero mean and have magnitudes commensurate with the innovation covariance as yielded by the filter.
3. The innovations should be acceptable as a white error process.

In other words, a track filter that is consistent produces a state error covariance that accurately represents the errors in the state estimate. Thus, track filter consistency is critical for effective fusion of data from multiple sensors with diverse accuracies. Maneuvering targets pose a particularly difficult challenge to achieving track filter consistency. In fact, more data does not necessarily mean better estimates when a Kalman filter is used to track a maneuvering target. Using a Kalman filter to track a maneuvering target will not provide reliable tracking performance because the loss of track filter consistency prevents reliable decision making for simultaneously adapting the filter parameters and performing data association.

Stochastic State Estimation (Kalman)

For stochastic state estimation, measurements and the dynamical motion model are given by (4.2) and (4.3) respectively

$$Z_k = H_k X_k + w_k \quad (4.2)$$

$$X_{k+1} = F_k X_k + G_k v_k \quad (4.3)$$

where

v_k = error in the system processes at time t_k with $v_k \sim N(0, Q_k)$.

G_k = relates the system errors to the target state at time t_k .

For a nearly constant velocity motion model, the state vector of the target in a scalar coordinate is given by

$$X_k = [x_k \quad \dot{x}_k]^T \quad (4.4)$$

where x_k represents the position of the target at time t_k , and \dot{x}_k represents the velocity of the target. Process noise is included to account for the uncertainty associated with unknown maneuvers and unmodeled dynamics of the target under track. Since the evolution of the state includes a stochastic process, the state estimates are a stochastic process, and the covariance of the estimate will not achieve zero and grow in the absence of measurements. The Kalman filter gives the MMSE and minimum variance estimate of the stochastic state X_k . Since the random processes v_k and w_k are additive Gaussian and the Kalman filter is a linear filter, the state estimation error of the Kalman filter will be Gaussian. Thus, only the mean and covariance are needed to fully characterize the state estimation error. The Kalman filter is a predictor-corrector algorithm with the predictor accounting for changes in time and the corrector accounting for the measurement processing. The Kalman algorithm is defined by the following equations, where $X_{k|j}$ denotes the state estimate at t_k given measurements through t_j , and $P_{k|j}$ denotes the state error covariance at t_k given measurements through t_j :

Prediction of the state estimate and covariance to the next time:

$$X_{k|k-1} = F_{k-1} X_{k-1|k-1} \quad (4.5)$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T \quad (4.6)$$

Update of the state estimate and covariance with the measurement:

$$X_{k|k} = X_{k|k-1} + K_k [Z_k - H_k X_{k|k-1}] = X_{k|k-1} + K_k \tilde{Z}_k \quad (4.7)$$

$$P_{k|k} = [I - K_k H_k] P_{k|k-1} \quad (4.8)$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (4.9)$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \quad (4.10)$$

where K_k is referred to as the Kalman filter gain, \tilde{Z}_k denotes the filter residual vector, and the S_k is the covariance of the measurement residual.

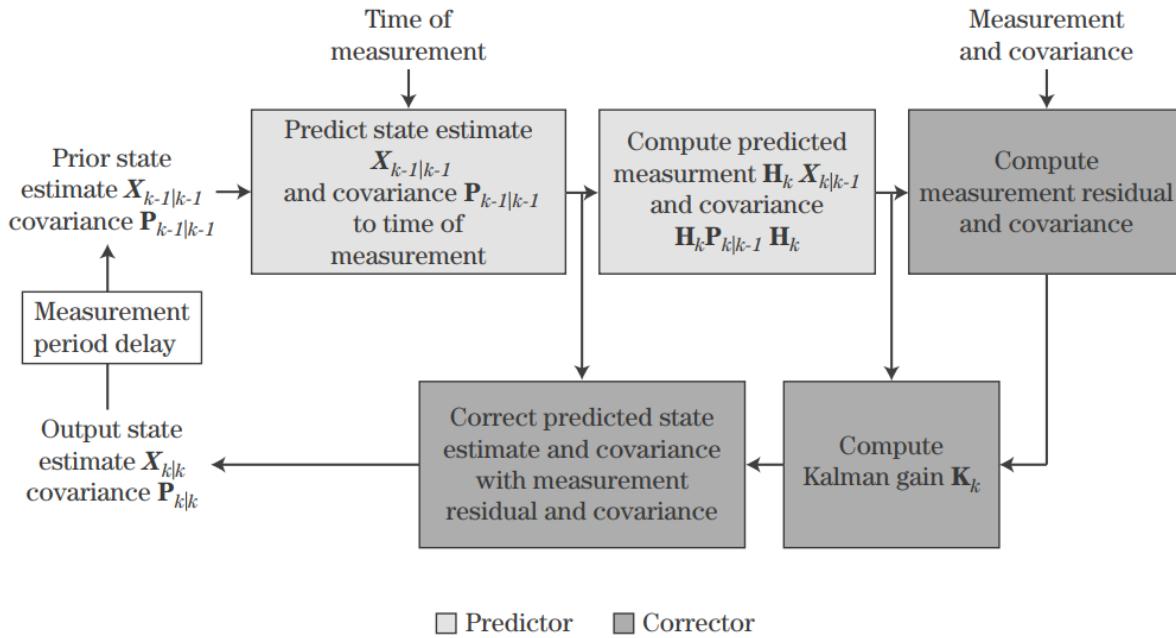


Figure 16: Kalman filter is a predictor-corrector algorithm.

The processing of the Kalman filter is depicted in Figure 16 as a predictor-corrector algorithm. The filter process is recursive. It starts with the initial state estimate $X_{k-1|k-1}$ and covariance $P_{k-1|k-1}$ at time t_{k-1} . The filtered state estimate and covariance are predicted to the time of the measurement t_k to obtain $X_{k|k-1}$ and $P_{k|k-1}$. The predicted measurement and its covariance are then computed to complete the predictor. The measurement and its covariance are then used to compute the measurement residual and its covariance. Then the Kalman gain is computed and the predicted state is updated with the measurement residual and the Kalman gain to form the new filtered state estimate $X_{k|k}$. The predicted covariance is updated with the Kalman gain to form the new filtered covariance $P_{k|k}$. The new filtered state estimate $X_{k|k}$ and covariance $P_{k|k}$ are then retained until the next filter cycle in which they become the new initial state estimate and covariance.

Nearly constant velocity and **nearly constant acceleration** are the most commonly used motion models and are showed in more detail in the following sections. The choice of a model is governed by the maximum acceleration of the target, the quality of the sensor measurements, and the measurement rate. Since no algorithm exists for predicting the best model, experience in the design process and Monte Carlo simulations are typically used to assess the best model for a tracking system.

4.3.1. KALMAN NEARLY CONSTANT VELOCITY

Nearly constant velocity motion with *discrete white noise acceleration* (DWNA) in a single coordinate is given by (4.3) with

$$X_k = [x_k \quad \dot{x}_k]^T \quad (4.11)$$

$$F_k = \begin{bmatrix} 1 & \delta_k \\ 0 & 1 \end{bmatrix}^T \quad (4.12)$$

$$G_k = \begin{bmatrix} \delta_k^2 \\ \frac{1}{2} \\ \delta_k \end{bmatrix} \quad (4.13)$$

$$Q_k = \sigma_{vk}^2 \quad (4.14)$$

where $\delta_k = t_k - t_{k-1}$. The v_k is a white noise acceleration error that is constant or fixed between t_k and t_{k-1} in the state process with $v_k \sim N(0, \sigma_{vk}^2)$. For the nearly constant velocity motion model, the process noise covariance matrix for DWNA is given by

$$G_k Q_k G_k^T = \sigma_{vk}^2 \begin{bmatrix} \frac{\delta_k^4}{4} & \frac{\delta_k^3}{2} \\ \frac{\delta_k^3}{2} & \frac{\delta_k^2}{2} \end{bmatrix} \quad (4.15)$$

The σ_{vk} is the design parameter for the nearly constant velocity (NCV) filter with DWNA errors. Typically, the filter design process begins by setting σ_{vk} greater than one half of the maximum acceleration of the target and less than the maximum acceleration, and Monte Carlo simulations are conducted to further refine the selection of σ_{vk} . Typically, the measurements are the position of the target. In this case, the measurement equation is defined by

$$H_k = [1 \quad 0] \quad (4.16)$$

When radars use an LFM waveform, the range measurement is coupled to the range rate-induced Doppler of the target. For an LFM waveform, DWNA, and fixed track rate, the system for tracking in the range coordinate is given by (4.11) through (4.16) with the changes given by

$$H_k = [1 \quad \Delta t] \quad (4.17)$$

where Δt is the range-Doppler coupling coefficient, which is defined by

$$\Delta t = \frac{f_1 \tau}{f_2 - f_1} \quad (4.18)$$

Given that most if not every target that we expect is going to have an ambiguous velocity, we are going to use (4.16) to avoid false coupling.

4.3.2. KALMAN NEARLY CONSTANT ACCELERATION

Nearly constant acceleration motion with piecewise constant acceleration errors in a scalar coordinate is defined by (4.3) with

$$X_k = [x_k \quad \dot{x}_k \quad \ddot{x}_k]^T \quad (4.19)$$

$$F_k = \begin{bmatrix} 1 & \delta_k & \frac{1}{2}\delta_k^2 \\ 0 & 1 & \delta_k \\ 0 & 0 & 1 \end{bmatrix}^T \quad (4.20)$$

$$G_k = \left[\frac{1}{2}\delta_k^2 \quad \delta_k \quad 1 \right]^T \quad (4.21)$$

where v_k is a discrete Wiener process acceleration (DWPA) error that is constant between t_k and t_{k-1} in the state process with $v_k \sim N(0, \sigma_{vk}^2)$. For the nearly constant acceleration motion model, the process noise covariance matrix for DWPA is given by

$$G_k Q_k G_k^T = \sigma_{vk}^2 \begin{bmatrix} \frac{\delta_k^4}{4} & \frac{\delta_k^3}{2} & \frac{\delta_k^2}{2} \\ \frac{\delta_k^3}{2} & \delta_k^2 & \delta_k \\ \frac{\delta_k^2}{2} & \delta_k & 1 \end{bmatrix} \quad (4.22)$$

where σ_{vk} is the design parameter for the nearly constant acceleration (NCA) filter with DWPA. Typically, the filter design process begins by setting σ_{vk} greater than one half of the maximum change in acceleration between t_{k-1} and t_k and less than the maximum change in acceleration. Monte Carlo simulations are in this case also conducted to further refine the selection of σ_{vk} . Typically, the measurements are of the position of the target. In this case, the measurement equation of (4.23) is defined by

$$H_k = [1 \quad 0 \quad 0] \quad (4.23)$$

5. PARAMETER SELECTION & RESULTS

5.1. PARAMETER SELECTION

This study has been carried out according to INDRA's guidelines. Thus, the parameters of the LFM-CW radar studied where given:

Parameter	Value
Central frequency (f_0)	3.5 GHz
Bandwidth (B)	10 MHz
Pulse Repetition Interval (PRI)	88 μ s
Ascending chirp duration (T)	88 μ s
Chirps per burst	64
Sampling frequency (f_s)	12 MHz
Light speed (c)	299792458 m/s

Consequently, we have from (2.8) and (2.13)

$$\Delta R = \frac{c}{2B} \approx 15 \text{ m}$$

$$\Delta v_r = \frac{\lambda}{2 \cdot N \cdot PRI} \approx 7.60 \text{ m/s}$$

and from (2.15) and (2.14)

$$R_{max} \leq \frac{c}{2 \cdot PRF} \approx 13.20 \text{ km}$$

$$|v_{r_{max}}| \leq \frac{PRF \cdot \lambda}{4} \approx 243.34 \text{ m/s}$$

Parameter	Value
Range resolution (ΔR)	15 m
Maximum unambiguous distance (R_{max})	13.20 km
Velocity resolution (Δv_r)	7.60 m/s
Maximum unambiguous velocity ($ v_{r_{max}} $)	243.34 m/s

The remaining parameters were not previously established. For the CA-CFAR algorithm we have chosen the following values which result in the window shown in Figure 17:

Parameter	Value
Number of guard cells per dimension	1
Number of training cells per dimension	2
Probability of False Alarm (P_{FA})	10^{-6}

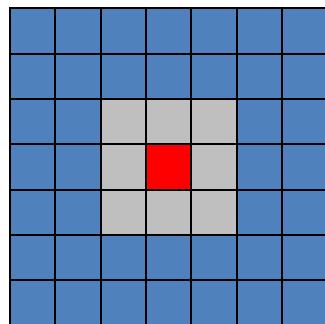


Figure 17: CA-CFAR window.

- Blue: Training cell
- Gray: Guarding Cell
- Red: Cell Under Test (CUT)

5.2. MOTION MODELS USED

5.2.1. LINEAR MOTION

$$v = cte \quad (5.1)$$

$$x = x_0 + v \cdot t \quad (5.2)$$

5.2.2. UNIFORMLY ACCELERATED LINEAR MOTION

$$v = v_0 + a \cdot t \quad (5.3)$$

$$x = x_0 + v_0 t + \frac{1}{2} a t^2 \quad (5.4)$$

$$a = cte \quad (5.5)$$

5.2.3. DIRECT FIRE

Direct fire refers to firing of a ranged weapon whose projectile is launched directly at a target within the line-of-sight of the user (artillery). In this case, the friction of the air is the only acceleration affecting the projectile.

$$v_x = v_0^x e^{-\frac{g}{v_t} t} \quad (5.6)$$

$$x = \frac{v_0^x v_t}{g} (1 - e^{-\frac{g}{v_t} t}) \quad (5.7)$$

Where g is the acceleration of the gravity, v_0^x is the radial velocity and v_t is the terminal velocity of the projectile:

$$v_t = \frac{mg}{C} \quad (5.8)$$

Where m is the mass of the projectile and C is the friction coefficient.

5.2.4. POLYNOMIAL MOTION

$$x = x_0 + v_0 t + b \frac{t^2}{2} + c \frac{t^3}{6} + d \frac{t^4}{12} \quad (5.9)$$

$$v = v_0 + bt + c \frac{t^2}{2} + d \frac{t^3}{3} \quad (5.10)$$

$$a = b + ct + dt^2 \quad (5.11)$$

Where b, c and d are constants to be set (note that b is the initial acceleration).

5.3. RESULTS

Every case is for targets with Swerling 1. Velocities and accelerations are assumed to be radial. The gating threshold is thought to make the plot-to-track association until the end, even if the track diverges. With a more realistic gating threshold what would happen is that traces would eventually die and new ones would be created every time the plot-to-track association failed. Instead, due to this large threshold, when the target is not detected the track might be associated to a false detection very far from the real position. In addition, the initial covariance matrix is roughly optimized for the artillery case, which has the higher ambiguous velocity. Thus, the gains obtained are not the optimal ones for every case.

Detected acceleration is being computed as the difference between the current velocity minus the previous one divided by the time passed. Consequently, detected acceleration is different than zero only when the target moves from a velocity cell to another. Because of this, acceleration is not being plotted for NCA (NCV does not have acceleration estimation) due to its low information contribution. Remember that this is a **tracking radar**, which implies that the important parameter here is the **error in distance**.

5.3.1. TARGET WITH CONSTANT VELOCITY

For a radar capable of detecting a generic target with $SNR = -33 \text{ dB}$ at 2.6 km , and having the case of a target **detected at 1 km** coming at us with **constant velocity of 150 m/s** we obtain

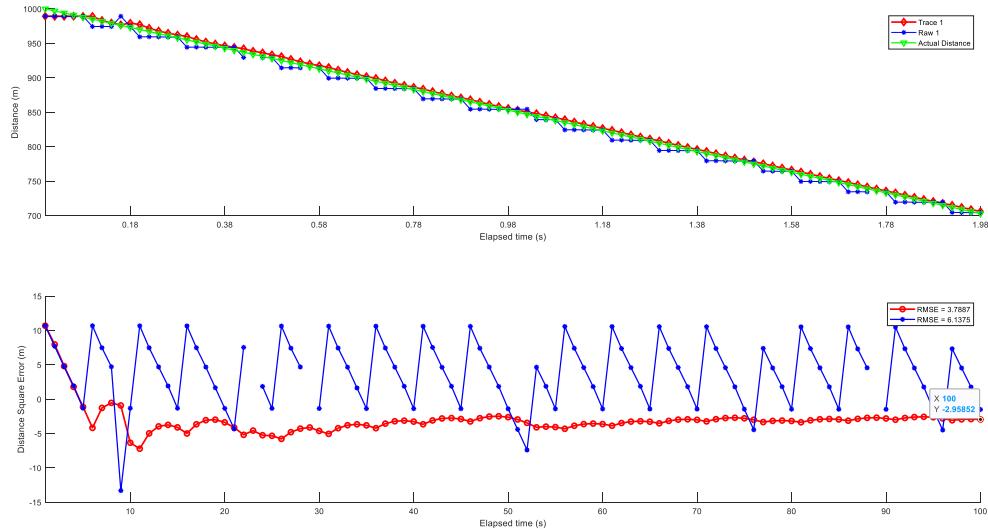


Figure 18: MRU with NCV (distance plot).

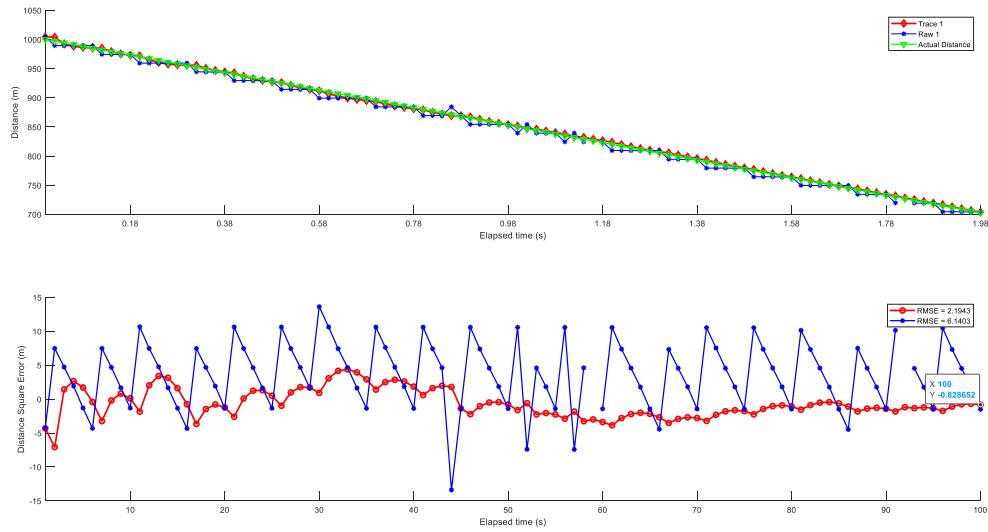


Figure 19:MRU with NCA (distance plot).

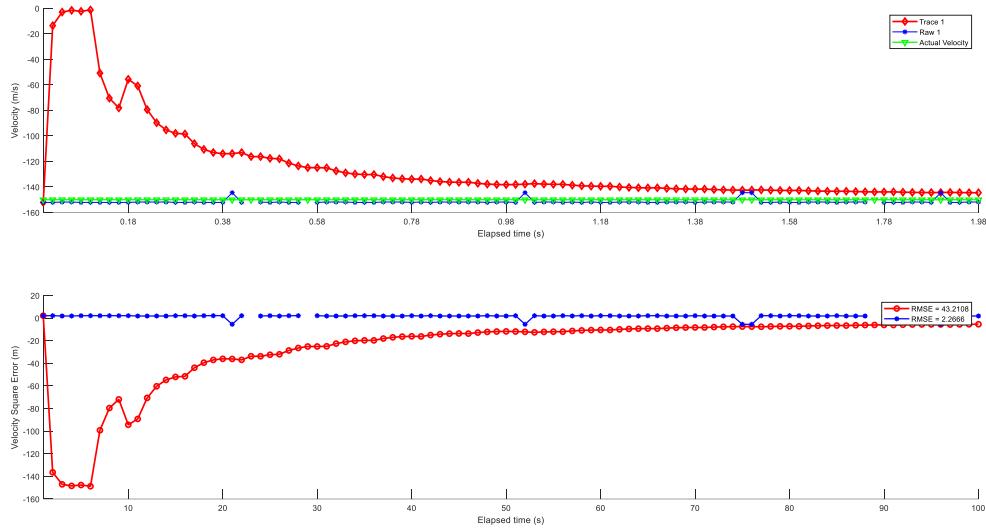


Figure 20: MRU with NCV (velocity plot).

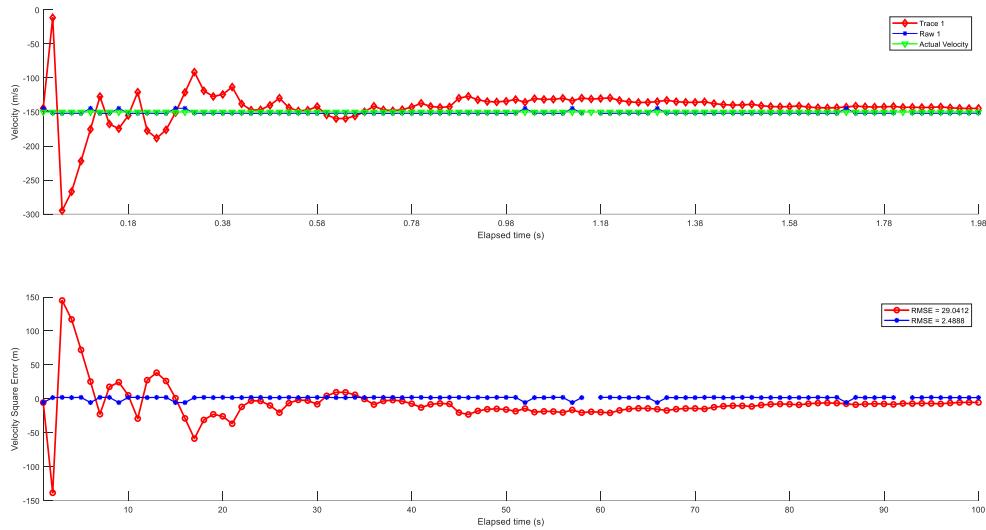


Figure 21: MRU with NCA (velocity plot).

These results may seem unexpected because the NCA is having a better performance in NCV's specialty. However, we must remember that **the covariance matrix is optimized for highly ambiguous velocities**. Thus, as the NCA is noticing faster that the velocity information has low error (Figures 20 and 21), it converges to zero faster.

5.3.2. TARGET WITH CONSTANT ACCELERATION

For a radar capable of detecting a generic target with $SNR = -33 \text{ dB}$ at 2.6 km , and having the case of a target **detected at 1 km** coming at us with **initial velocity of 150 m/s** and **constant acceleration of 100 m/s}^2** we obtain

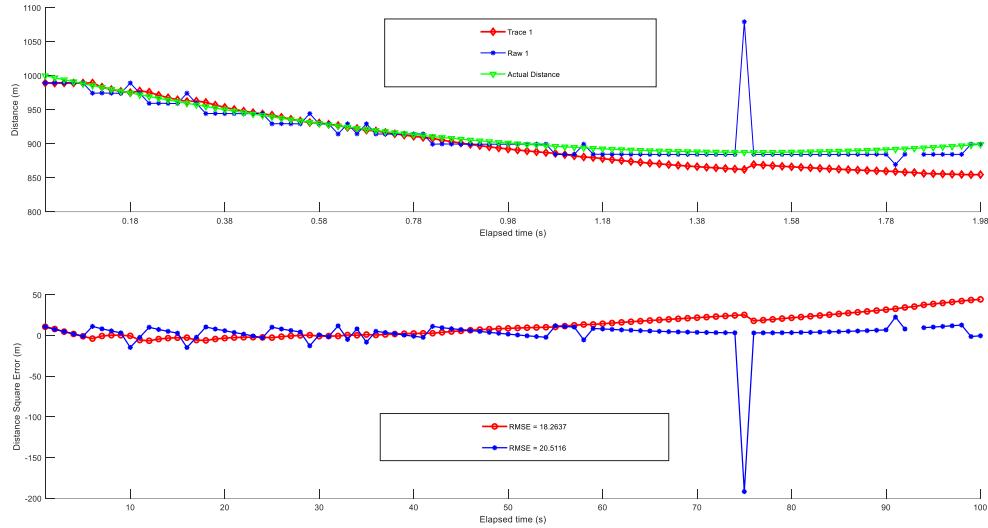


Figure 22: MRUA with NCV (distance plot).

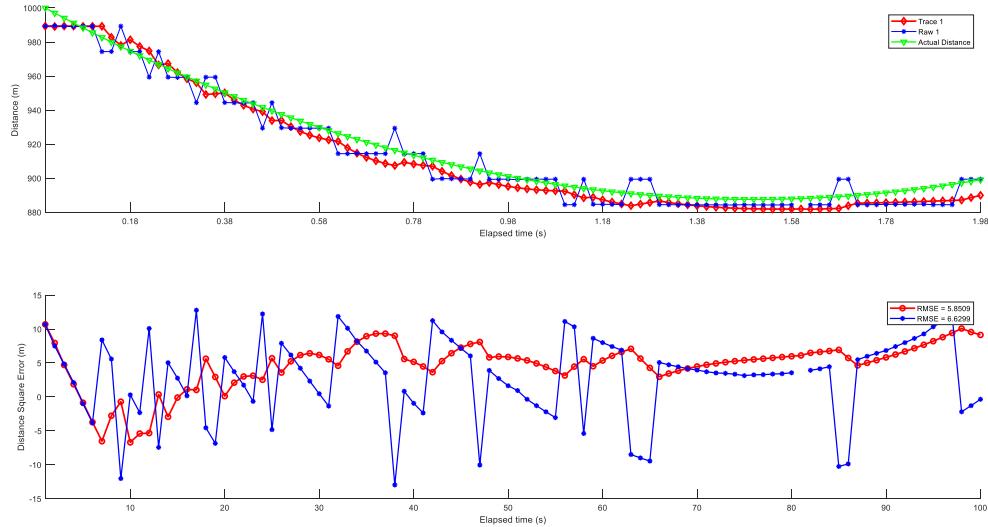


Figure 23: MRUA with NCA (distance plot).

We see in Figures 22 and 24 an example of wrong plot-to-track association caused by the large gating threshold. In Figure 26 we can find what would be plotted in case of a more realistic threshold.

Looking at Figure 24 makes trivial to deduce why the track is not being properly followed. What is happening is that the covariance matrix converges to its final value. Thus, as the filter is *constant velocity*, the velocity is assumed constant from that moment and the track eventually diverges from the real value. Again, the gain for NCA is derisory because the initial covariance matrix is not optimized for unambiguous velocities.

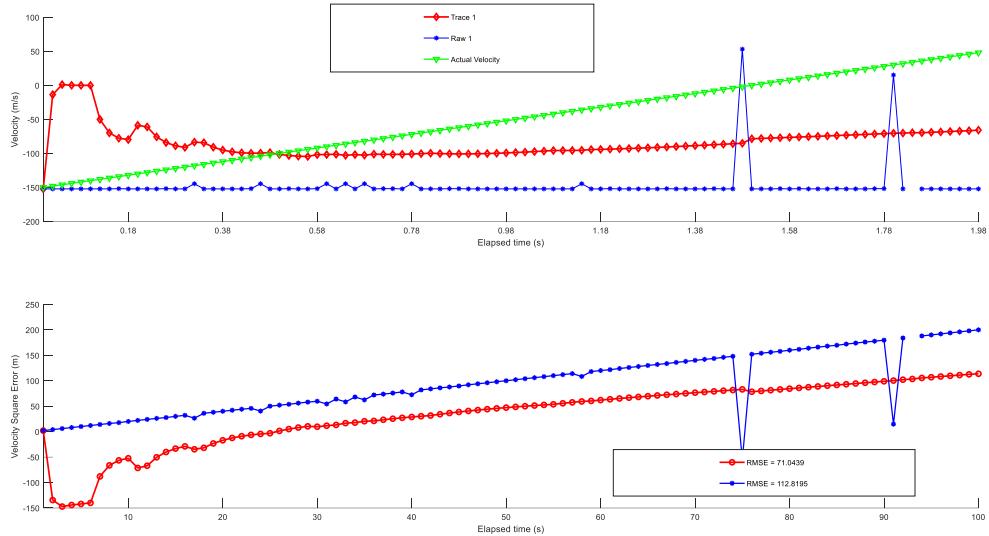


Figure 24: MRUA with NCV (velocity plot).

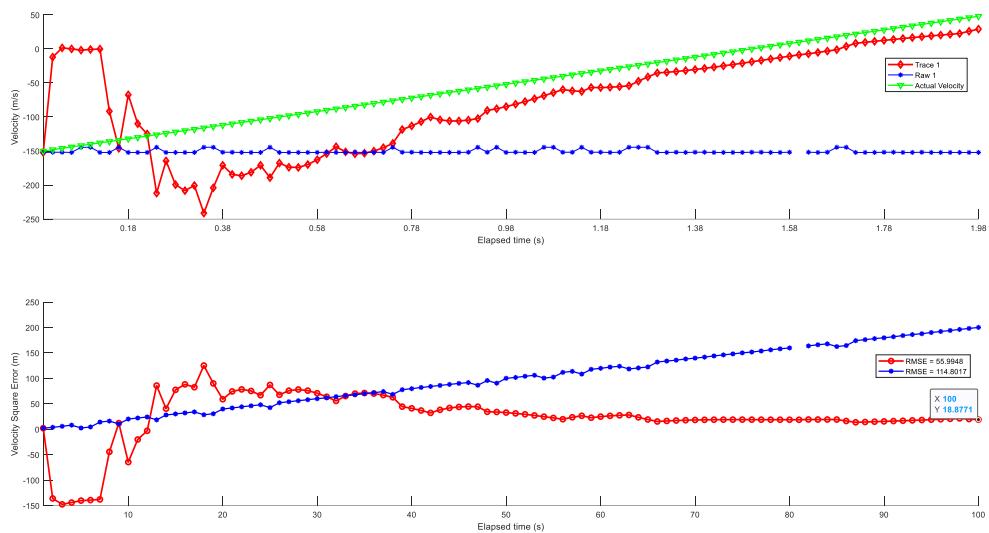


Figure 25: MRUA with NCA (velocity plot).

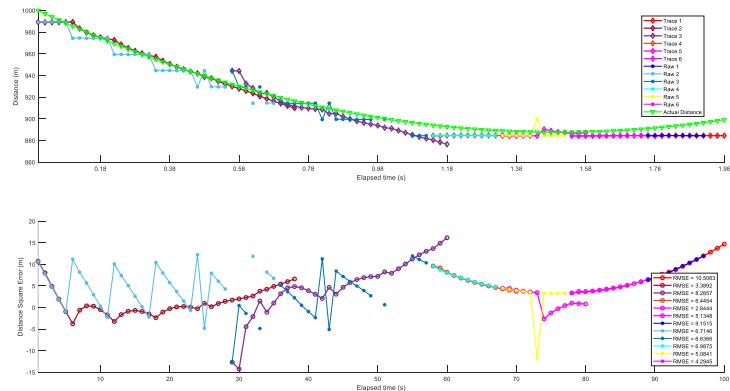


Figure 26: Many tracks are being created while others are being lost with NCV if the gating threshold is realistic.

5.3.3. RPG-7

For a radar capable of detecting an RPG-7 with $SNR = -33 \text{ dB}$ at 7.9 km , and having the case of a target **detected at 300 m** coming at us with **initial velocity of 290 m/s** and **constant acceleration of 126 m/s}^2** (thus, assuming Uniformly Accelerated Linear Motion) we obtain

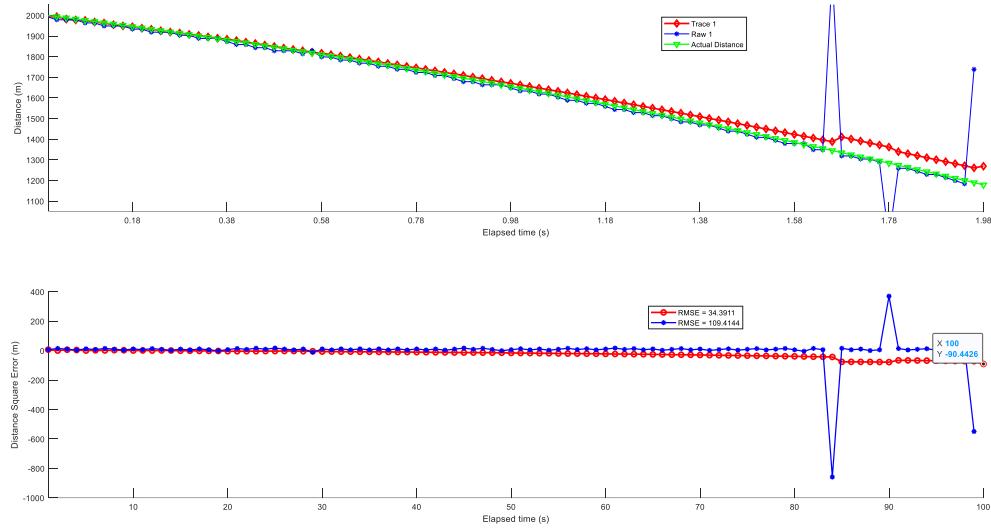


Figure 27: RPG-7 with NCV (distance plot).

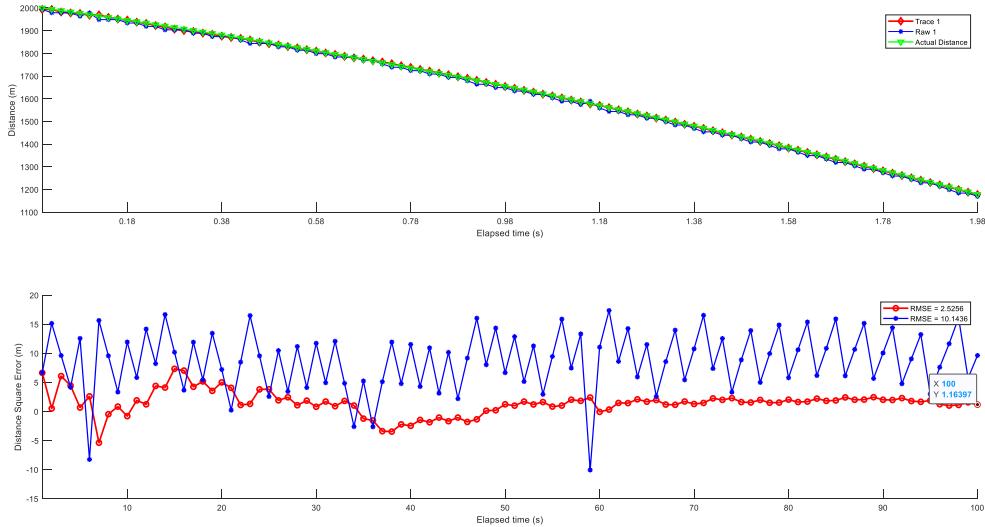


Figure 28: RPG-7 with NCA (distance plot).

As we expected, some fake associations are again present because the NCV wasn't able to keep the track alive with a realistic threshold. In Figure 29 we appreciate again how the NCV stabilizes when the ambiguous velocity is corrected and remains nearly constant for the rest of the tracking. On the other hand, NCA's response is much better in this case because we are treating with ambiguous velocities. The initial ambiguous velocity is quickly corrected (Figure 30) like in NCV, but NCA, as we expected, is capable of following the constant variation on the velocity. With all this, NCA converges to one meter of distance square error and four times lower RMSE than the raw data.

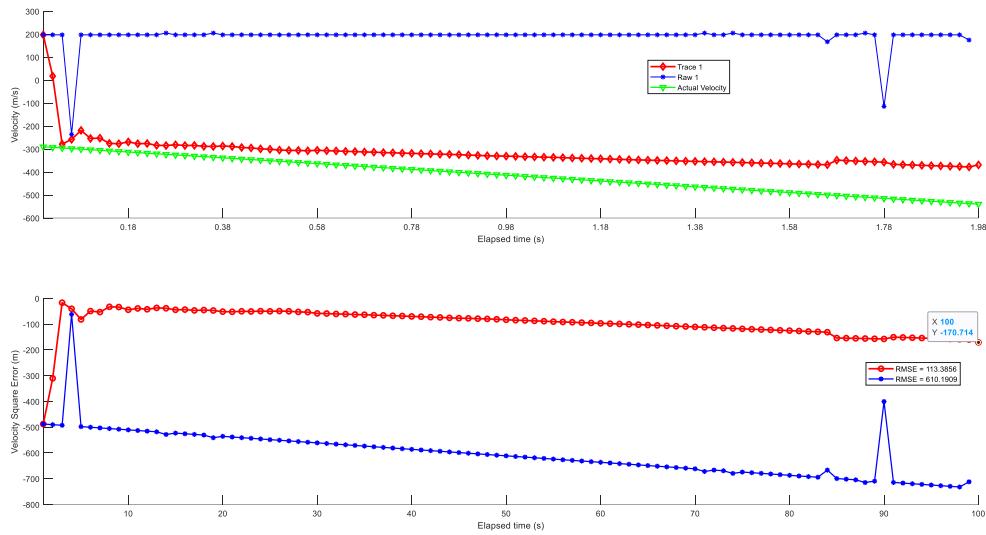


Figure 29: RPG-7 with NCV (velocity plot).

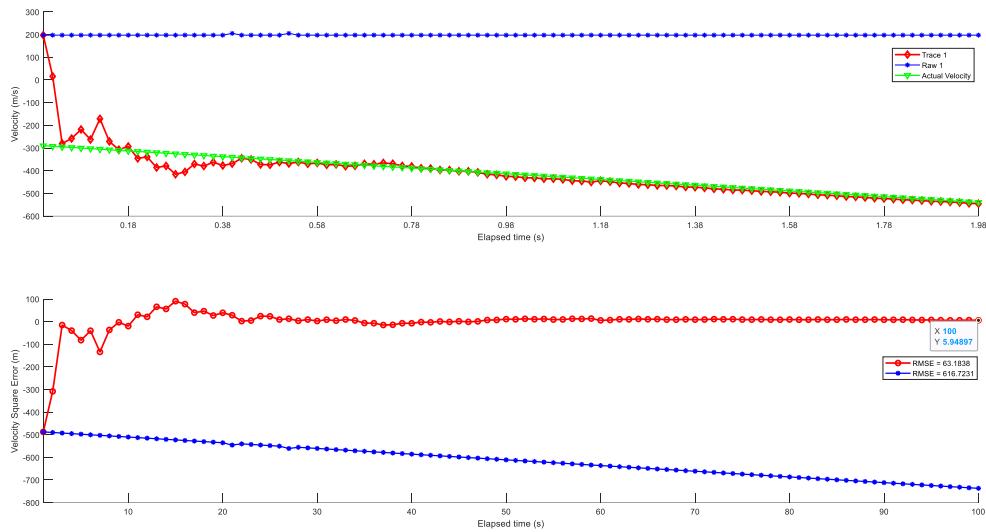


Figure 30: RPG-7 with NCA (velocity plot).

5.3.4. LOITERING DRONE

For a radar capable of detecting a drone (for loitering attacks) with $SNR = -33 dB$ at 790 m, and being firstly **detected at 2 km** at the moment of leaving the loitering position (**initial velocity = 0**) with **constant acceleration of 42.5 m/s^2** (thus, assuming Uniformly Accelerated Linear Motion) we obtain

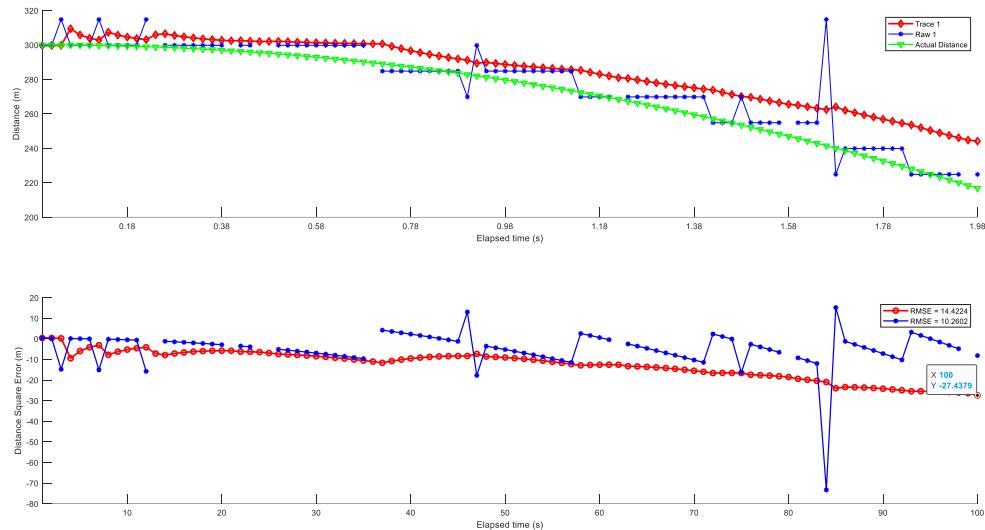


Figure 31: Loitering drone with NCV (distance plot).

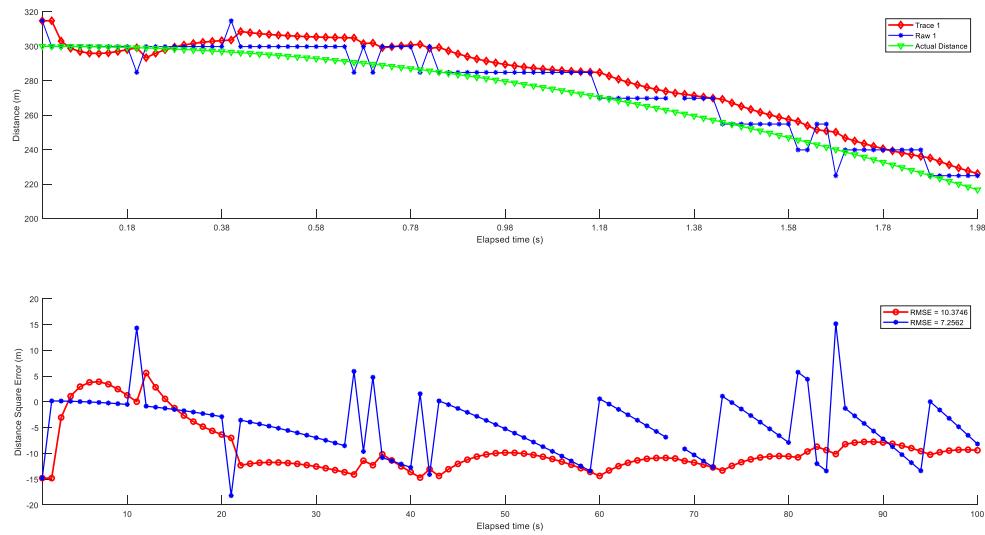


Figure 32: Loitering drone with NCA (distance plot).

Yet again, NCV filter presents a very poor response for an accelerated target. What is surprising is that the tracking with NCA presents worse RMSE than the raw data. This may be caused by an *unlucky* positioning of the target during its trajectory, i.e., due to the relatively low resolution in distance and the low velocity of the target, the drone is being detected always above its actual trajectory (Figure 32). We can confirm this theory in Figure 35, where the initial distance of the drone has been moved 30 meters, which causes the real trajectory to be closer to the values of the cells in the Range-Doppler Matrix.

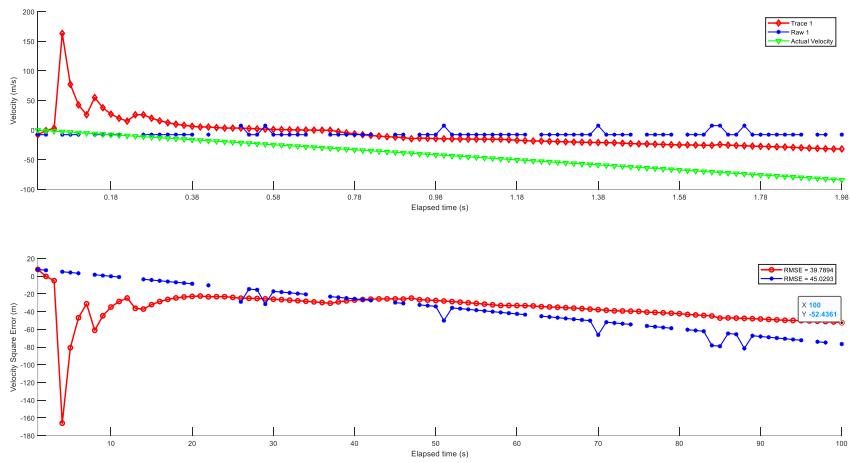


Figure 33: Loitering drone with NCV (velocity plot).

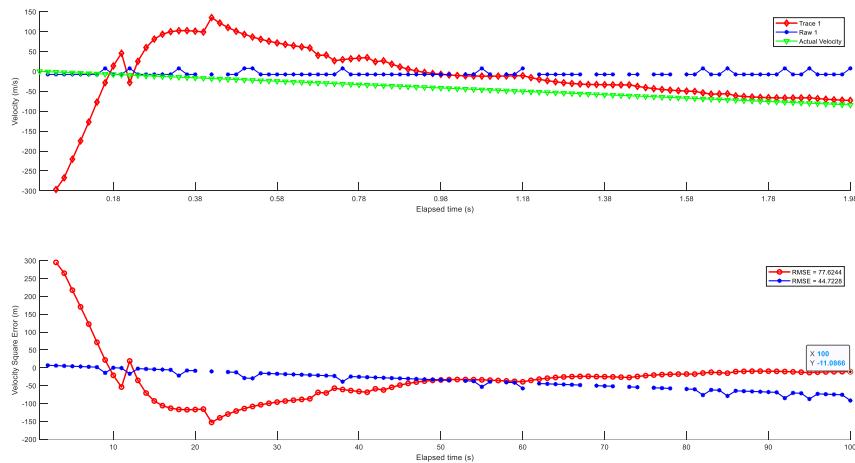


Figure 34: Loitering Drone with NCA (velocity plot).

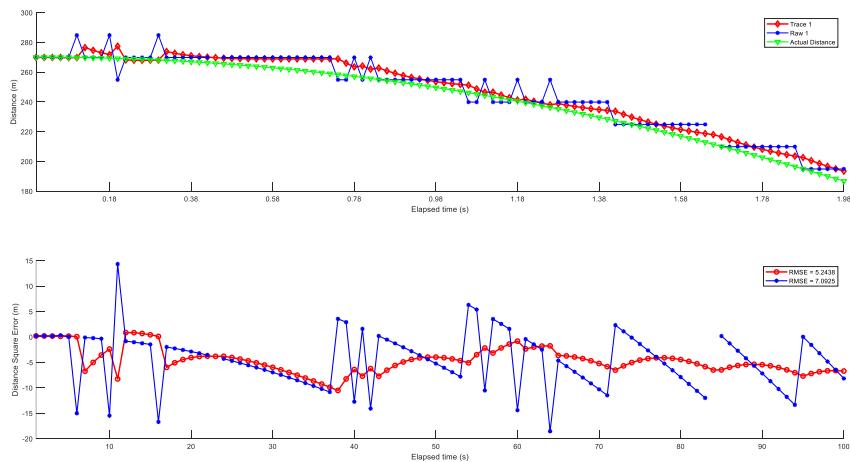


Figure 35: Loitering Drone with NCA (distance plot) 30 meters closer than the original case.

5.3.5. MANOEUVRING TARGET

In this case we assume the polynomial motion mentioned in section 5.2.4 with constants $b = 20$, $c = b \cdot 2$ and $d = -4.5 \cdot b$. For a radar capable of detecting a target with **$SNR = -33 dB$** at **$2.6 km$** , and having the case of a target **detected at 1000 m** coming at us with **initial velocity of 300 m/s** and **initial acceleration of $b=20 m/s^2$** , we obtain

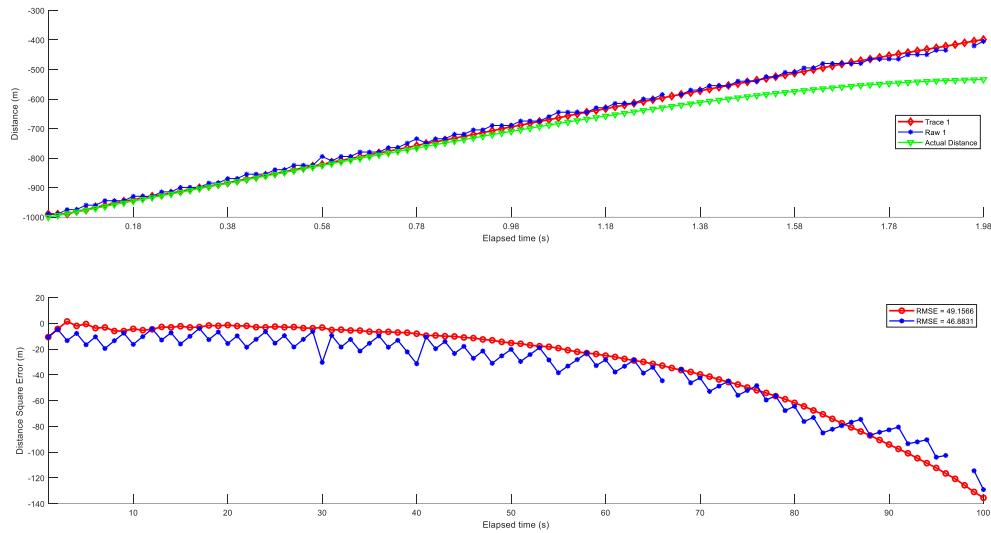


Figure 36: Maneuvering target with NCV (distance plot).

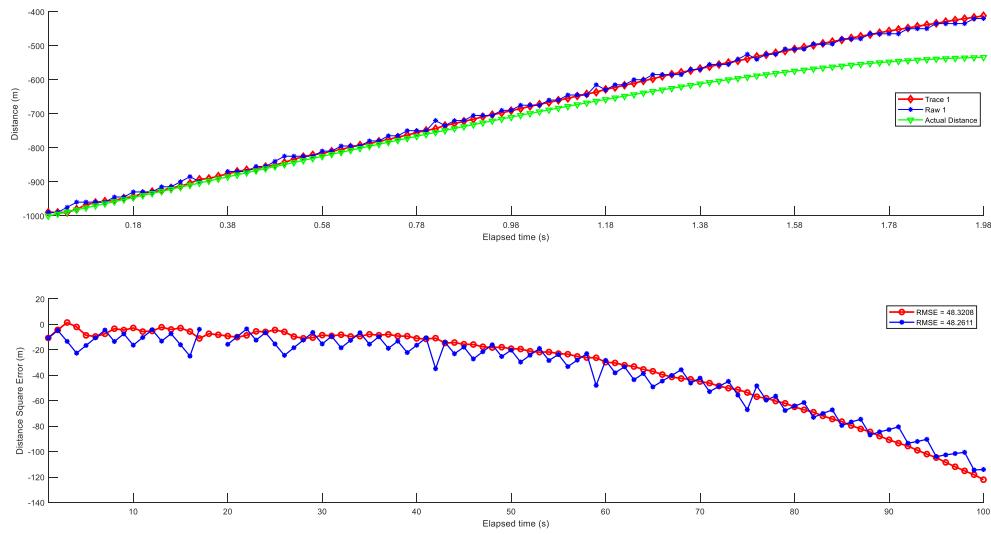


Figure 37: Maneuvering target with NCA (distance plot).

The results of this simulation are no surprise. In these figures it is shown why following maneuvering targets it is a huge challenge. The target is well followed while its velocity is relatively constant. When it abruptly changes its velocity, both filters have already converged and they lose the target. In Figure 39, since in the NCA the acceleration is taken into account, we see that the filter tries to follow the target but the change on velocity is too steep (*nearly constant acceleration*).

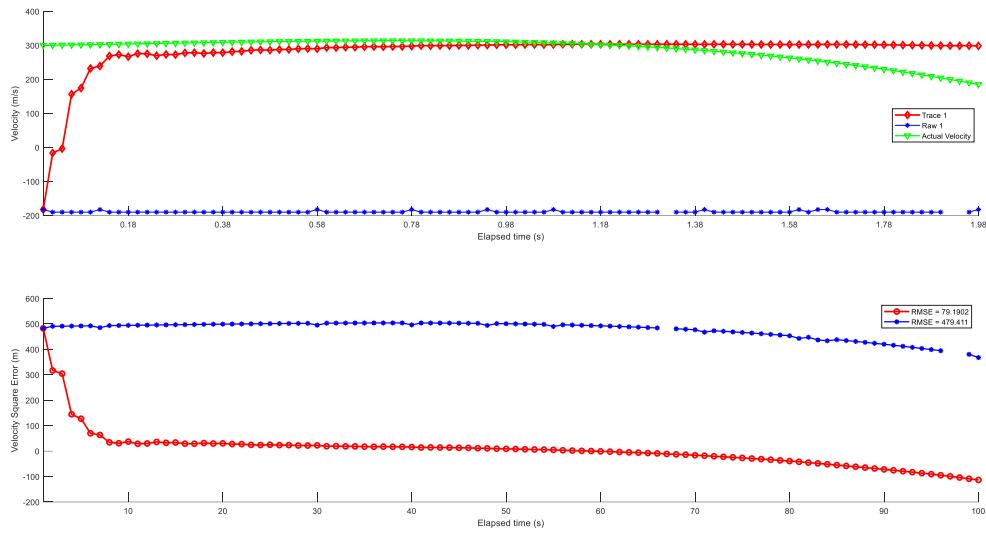


Figure 38: Maneuvering target with NCV (velocity plot).

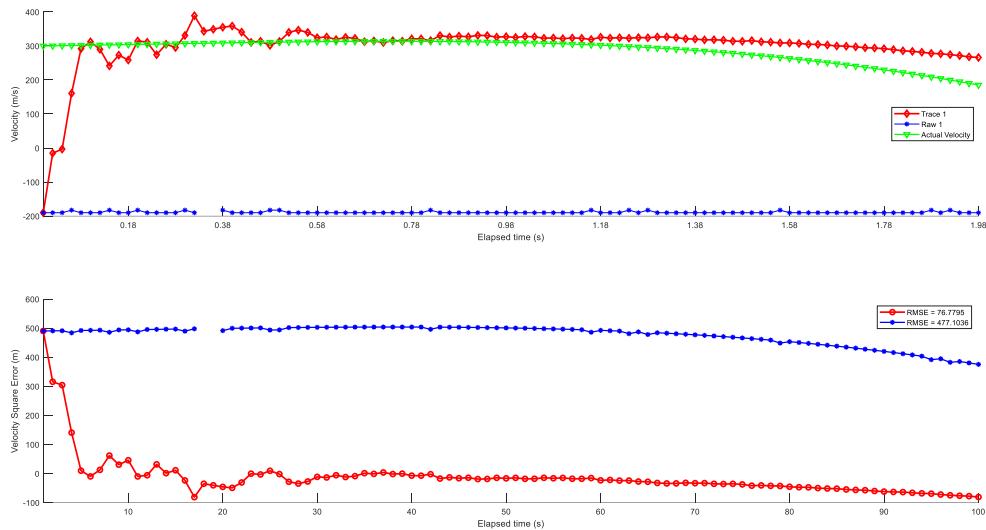


Figure 39: Maneuvering target with NCA (velocity plot).

5.3.6. ARTILLERY TARGET

In this case, as we are assuming direct fire, we define $m = 18.6$ Kg as the mass of the projectile. For a radar capable of detecting a target with **$SNR = -33 dB$ at 10 km**, and having the case of a target **detected at 4000 m** coming at us with **initial velocity of 1470 m/s**, we obtain

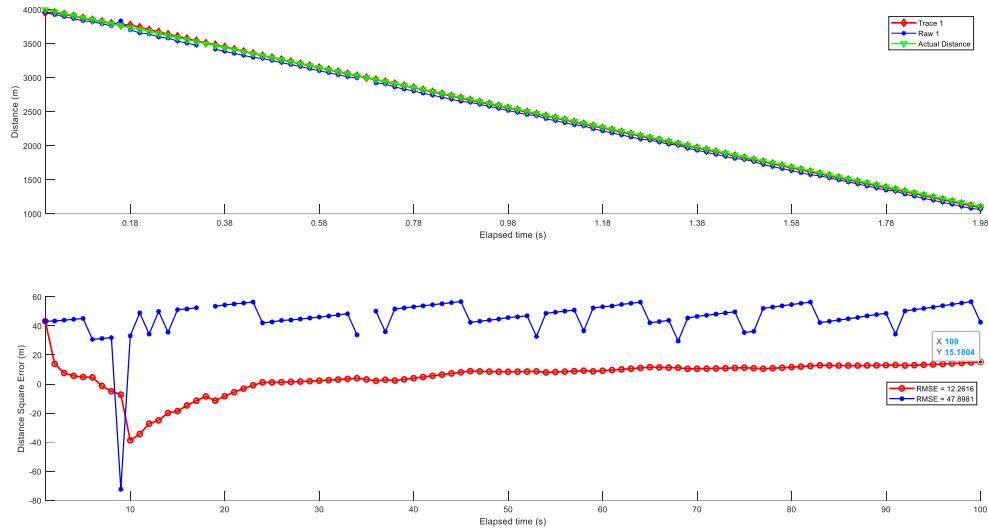


Figure 40: Artillery with NCV (distance plot).

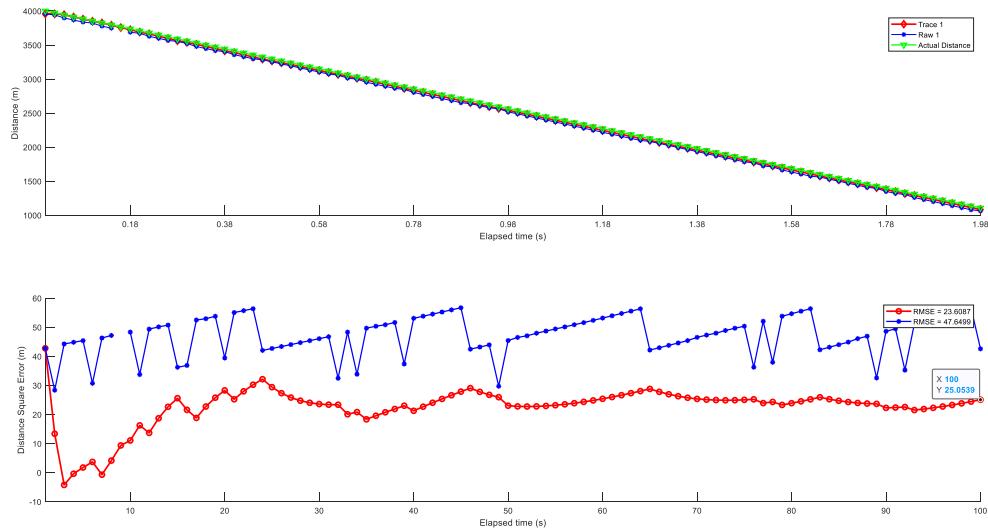


Figure 41: Artillery with NCA (distance plot).

By looking at these figures, it is easy to confirm that the data processing has been optimized for these high velocities. Even though this study case has acceleration (deceleration), with NCV filtering we are obtaining better performance. This can be explained by two different things:

- 1- The fact that filter optimization has not been fine-tuned.
- 2- Target's deceleration is so tiny that Nearly Constant Velocity defines better the target's motion than Nearly Constant Acceleration.

(Note that the initial velocity differs on 18 m/s with the final value, which means that the velocity changes only 1.22% from beginning to end).

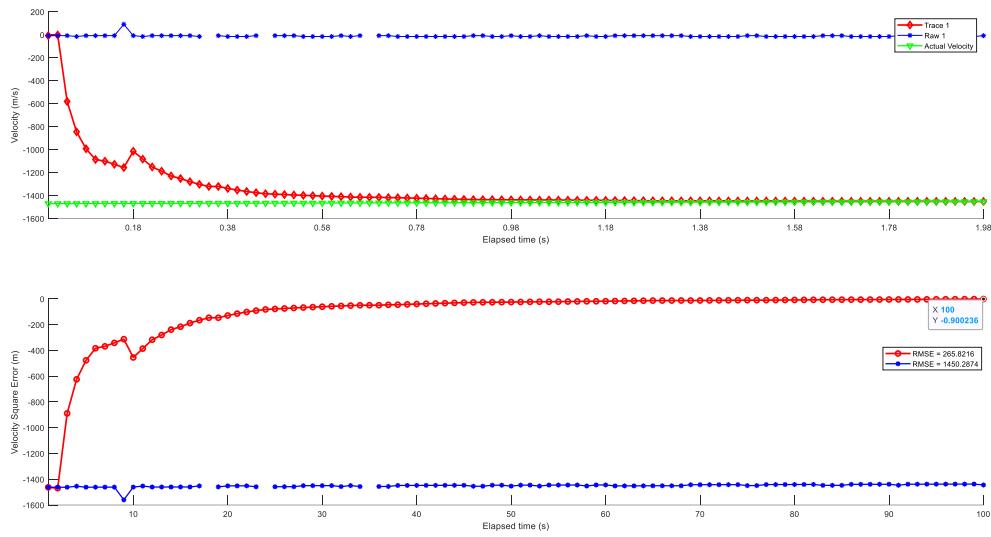


Figure 42: Artillery with NCV (velocity plot).

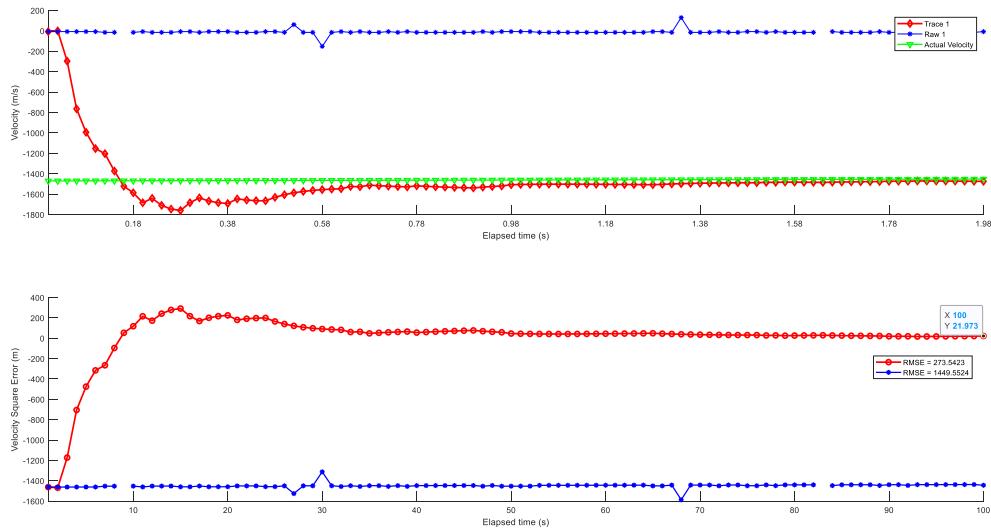


Figure 43: Artillery with NCA (velocity plot).

5.3.7. LOITERING DRONE AND RPG-7

Joining both 5.3.3 and 5.2.4 cases we obtain

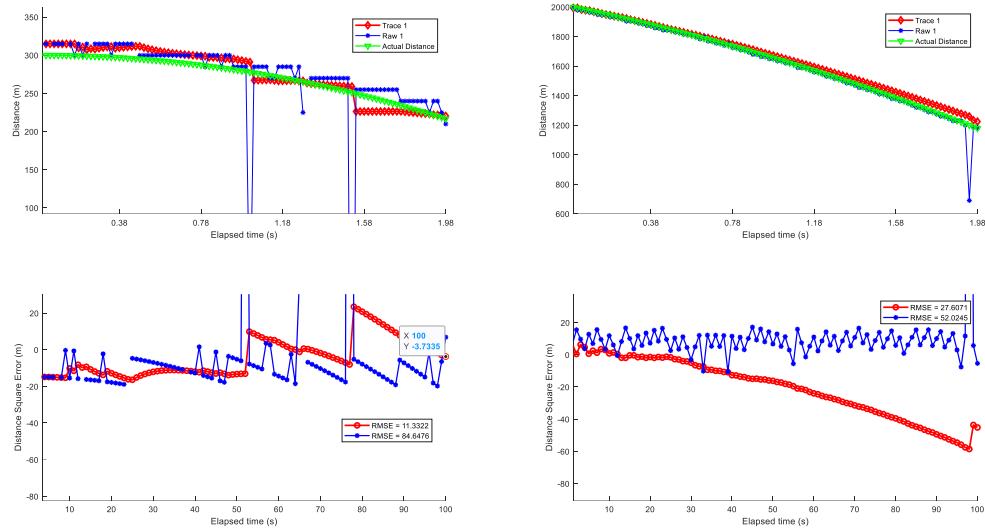


Figure 44: Loitering (left) + RPG (right) with NCV (distance plot).

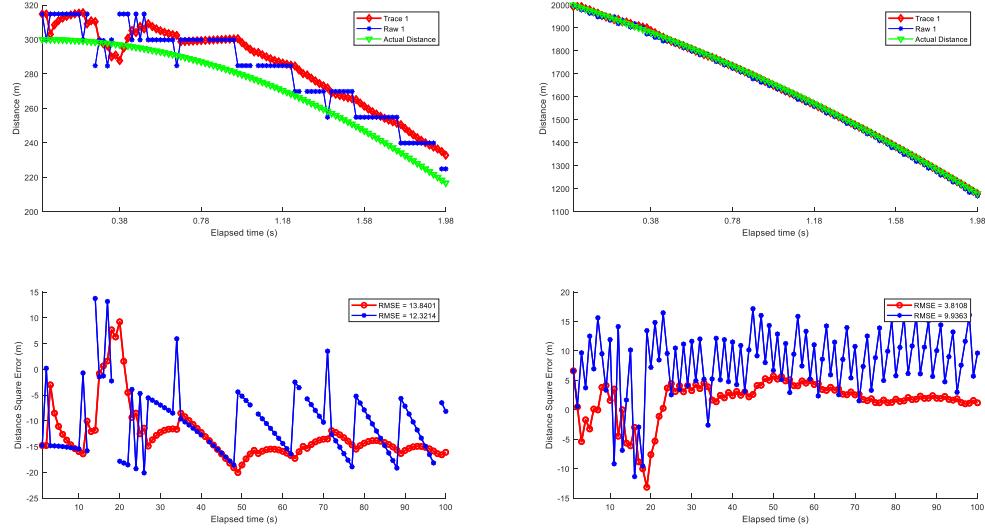


Figure 45: Loitering (left) + RPG (right) with NCA (distance plot).

The only thing that changes with respect the individual cases is the increase in the noise received by the radar. Consequently, the overall performance worsens.

Ironically, the fake associations done with NCV for the loitering drone is making the RMSE lower than for the NCA case. However, this is a punctual situation and in a real-life situation it won't work: just as the fake associations have improved the performance the opposite could have happened.

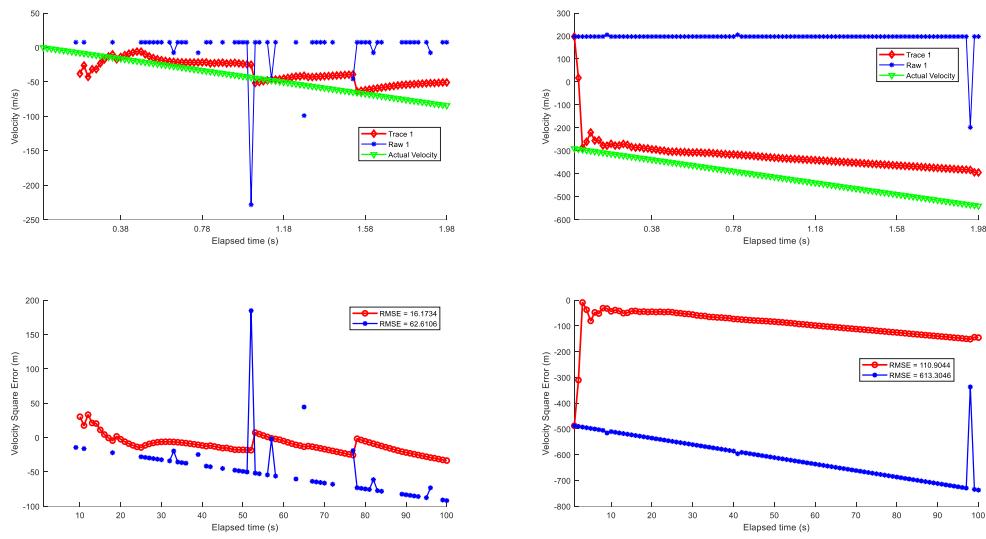


Figure 46: Loitering (left) + RPG (right) with NCV (velocity plot).

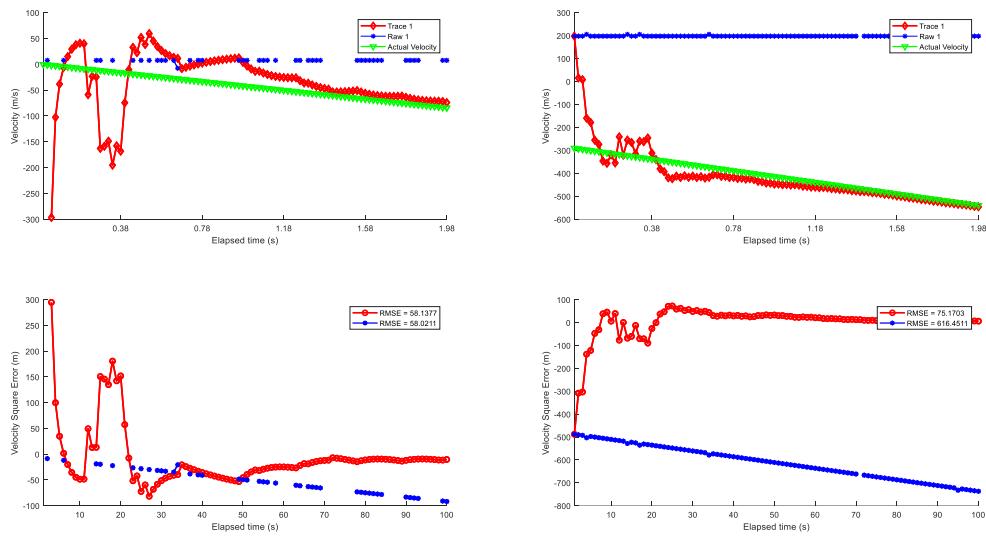


Figure 47: Loitering (left) + RPG (right) with NCA (velocity plot).

6. CONCLUSIONS & FUTURE LINES OF ACTION

6.1. CONCLUSIONS

After analyzing these results, we can confirm that *Nearly Constant Acceleration* Kalman filtering has better performance than *Nearly Constant Velocity* in most cases when the target presents acceleration. However, if the acceleration is sufficiently low so the velocity remains nearly constant, NCV seems to have better tracking performance than NCA.

Distance resolution is key for target tracking, especially for low velocity targets, since they will remain during a longer time in the same range cell: the longer time the target remains in the same cell, the higher underperformance of the filter will be suffered, giving the case of worse error with filter than without it.

This kind of radar is normally used accompanied by a countermeasure system (e.g., antimissile weapon). Hence, the distance error of the track must be very low so the target can be intercepted. For this reason, optimizing our tracking for highly ambiguous velocities might not be ideal in a scenario with ambiguous and unambiguous velocities. What's more, with the distance resolution used in these simulations we won't be able to provide enough accuracy to take down most of the threats in any case (ambiguous/unambiguous).

As a final point, we have proven that linear filters are of no use when the time of tracking a maneuvering target comes.

6.2. FUTURE LINES OF ACTION

First of all, Monte Carlo simulations should be carried out for each study scenario to do a fairer comparison between algorithms. The use of these roughly optimized Kalman parameters might have introduced a considerable bias. After doing this, the gating threshold could be set at a more realistic value, which will make more relevant plotting a tracking process more similar to a real-life case (traces are lost and reinitialized).

In addition, it would be interesting to study the effects of **increasing distance resolution** or maximum unambiguous velocity. Furthermore, the case of one target *slowly* passing another one would be useful to study the effect of target masking.

7. REFERENCES

- [1] "History of Radar." Wikipedia. Wikimedia Foundation, October 20, 2021. https://en.wikipedia.org/wiki/History_of_radar#World_War_II_radar. Accessed January 2022.
- [2] "Heinrich Hertz." Wikipedia. Wikimedia Foundation, December 20, 2021. https://en.wikipedia.org/wiki/Heinrich_Hertz. Accessed January 2022.
- [3] "Radar during World War II." Radar during World War II - Engineering and Technology History Wiki. https://ethw.org/Radar_during_World_War_II. Accessed January 2022.
- [4] "Cold War Era." RAF Air Defence Radar Museum, March 11, 2021. <https://www.radarmuseum.co.uk/history/cold-war-era/>. Accessed January 2022.
- [5] Urrea, Claudio, and Rayko Agramonte. *Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation*. Journal of Sensors. Hindawi, September 3, 2021. <https://www.hindawi.com/journals/js/2021/9674015/>. Accessed January 2022.
- [6] González Partida, José Tomás. *NUEVAS TÉCNICAS PARA SENSORES RADAR EMBARCADOS EN VEHÍCULOS AÉREOS NO TRIPULADOS*. Tesis (Doctoral), E.T.S.I. Telecomunicación (UPM), 2009.
- [7] Parker, Michael. *Digital Signal Processing 101: Everything You Need to Know to Get Started*. Seconded. Newnes, 2017.
- [8] Oppenheim, Alan V., Alan S. Willsky, and Syed Hamid Nawab. *Signals & Systems*. Noida, India: Pearson, 2015.
- [9] Hyun, Eugin, Young-Seok Jin, and Jong-Hun Lee. 2016. *A Pedestrian Detection Scheme Using a Coherent Phase Difference Method Based on 2D Range-Doppler FMCW Radar Sensors* 16, no. 1: 124. <https://doi.org/10.3390/s16010124>
- [10] Richards, M. A. *Fundamentals of Radar Signal Processing*. Seconded. New York: McGraw-Hill, 2014.
- [11] Richards, Mark A., James A. Scheer, and William A. Holm. *Principles of Modern Radar*. Raleigh, NC: Scitech Publishing Inc, 2015.
- [12] Gasior, Marek & Gonzalez, Jose. (2004). *Improving FFT Frequency Measurement Resolution by Parabolic and Gaussian Spectrum Interpolation*. 732. 10.1063/1.1831158.
- [13] Skolnik, Merrill I. *Radar Handbook*. New York: McGraw-Hill, 2008.
- [14] Mahafza, Bassem R. *Radar Systems Analysis and Design Using MATLAB*. Boca Raton (Fla.): Chapman & Hall, 2005.
- [15] Wehner, Donald R. *High-Resolution Radar*. Boston: Artech House, 1995.
- [16] Oppenheim, Alan V., and Li Lee. *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Prentice Hall, 1999.
- [17] Siouris, George M. *Missile Guidance and Control Systems*. New York: Springer, 2011.

ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES

A.1 INTRODUCCIÓN

Este proyecto pretende rebajar los costes de diseño del procesamiento de datos de un radar LFM-CW. Con el auge de los drones y la “moda” de los ataques de tipo *loitering*, la detección de amenazas aéreas es crítica en ciertos lugares del mundo.

A.2 IMPACTO AMBIENTAL

El impacto ambiental no va más allá de los recursos utilizados: el ordenador y sus respectivos periféricos. El impacto ambiental más directo sería quizás en los momentos de uso intensivo del ordenador durante las simulaciones y el consecuente incremento en la electricidad consumida.

A.3 IMPACTO ECONÓMICO

Este simulador permitirá ahorrar tiempo a los ingenieros de INDRA y mejorar sus diseños, con lo que es complicado estimar un impacto económico concreto. Lo que sí podemos decir es que el impacto será menor, aunque sí tiene cierto potencial.

A.4 IMPACTO SOCIAL

El impacto social es elevado por el contexto: la defensa aérea es fundamental para la seguridad de un país. Este proyecto ayuda indirectamente a que la población civil de zonas conflictivas pueda vivir con mayor seguridad y se pueda desarrollar como sociedad sin tanta incertidumbre.

A.5 CONCLUSIONES

Podemos concluir que este proyecto no supone a priori un gran impacto económico ni tampoco ambiental. Por otro lado, debido al contexto en el que se usa esta tecnología, se podría decir que el proyecto tiene un impacto social moderado o incluso elevado, aunque, eso sí, de forma indirecta.

ANEXO B: PRESUPUESTO ECONÓMICO

COSTE DE MANO DE OBRA (coste directo)

Horas	Precio/hora	Total
500	20 €	10.000 €

COSTE DE RECURSOS MATERIALES (coste directo)

	Precio de compra	Uso en meses	Amortización (en años)	Total
Ordenador personal (sobremesa+portátil+periféricos)	1.500,00 €	6	5	150,00 €
Licencia de MATLAB	1.000,00€	9	1	500,00€

COSTE TOTAL DE RECURSOS MATERIALES

10.650,00 €

GASTOS GENERALES (costes indirectos)

15%

sobre CD

1.597,50 €

BENEFICIO INDUSTRIAL

6%

sobre CD+CI

639,00 €

SUBTOTAL PRESUPUESTO

12.886,50 €

IVA APPLICABLE

21%

2.706,17 €

TOTAL PRESUPUESTO

15.592,67 €

ANEXO C: SIMULADOR

El código del simulador desarrollado puede obtenerse en el siguiente enlace https://github.com/pbielzalm/TFG_LFMCW