

# Radar code documentation

Frank Hermann

May 10, 2022

## 1 Introduction

A novel binary phase radar code is presented which has optimal noise canceling properties for the lag value estimator being considered. Furthermore, it is shown that the estimator is close to the Cramér–Rao bound. The code intrinsically measures the entire range at once, which must be finite.

## 2 Estimator and code requirement for zero bias

We assume that the radar code  $\epsilon$  has length  $L$  and is periodic  $\epsilon_i = \epsilon_{i+L}$ . Furthermore, the radar code is normalized and binary phase, meaning that  $\epsilon_i \in [-1, 1]$ . This implies that  $\epsilon_i^* \epsilon_i = 1$ , where we could drop the complex conjugate symbol, but we will keep it to show that the formalism also applies to poly-phase codes. We further assume that the radar code is transmitted indefinitely.

Figure 1 illustrates the measurement process of the unmodulated code. In the figure, it can be seen that the measured voltage  $m_i$  at time index  $i$  can be calculated by summing over the scattered voltages  $V_i^{(k)}$  which are modulated with the radar code,

$$m_i = \sum_{k=0}^{N-1} \epsilon_{i-k} V_i^{(k)} \quad (1)$$

where  $N$  is the maximum altitude index considered and  $k$  is the altitude index. Note that  $N$  must be finite.

Now, the estimator

$$\hat{R}_{\Delta}^{(h)} = \frac{1}{L} \sum_{j=0}^{L-1} \epsilon_{j+\Delta} m_{j+\Delta+h}^* \epsilon_j^* m_{j+h}. \quad (2)$$

for the autocorrelation function  $R$  at altitude  $h$  and lag  $\Delta > 0$  is introduced. In the following, it will be shown that this estimator is unbiased, if the code fulfills a certain condition.

We calculate the expectation value of the estimator

$$\langle \hat{R}_{\Delta}^{(h)} \rangle = \frac{1}{L} \sum_{j=0}^{L-1} \epsilon_{j+\Delta} \epsilon_j^* \sum_{k=0}^{N-1} \epsilon_{j+\Delta+h-k}^* \epsilon_{j+h-k} \langle (V_{j+\Delta}^{(k)})^* V_j^{(k)} \rangle \quad (3)$$

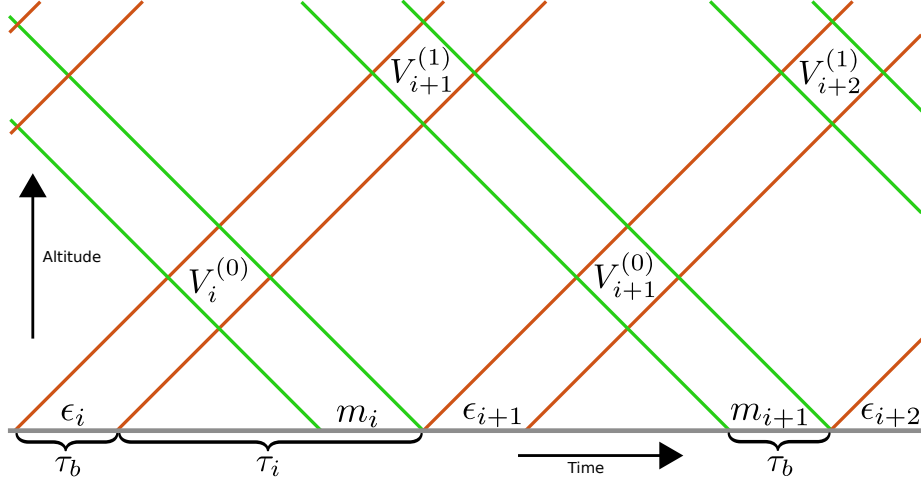


Figure 1: Range time diagram of the simplified version of the radar code.

where the double sum was reduced to a single sum symbol, because scattered voltages of different altitudes do not correlate. Now, we use the definition of the autocorrelation function  $\langle (V_{j+\Delta}^{(k)})^* V_j^{(k)} \rangle = R_{\Delta}^{(k)}$ , where we assume a wide sense stationary process during the time it takes to transmit the code once.

$$\langle \hat{R}_{\Delta}^{(h)} \rangle = R_{\Delta}^{(h)} + \sum_{k=0, k \neq h}^{N-1} R_{\Delta}^{(k)} \left[ \frac{1}{L} \sum_{j=0}^{L-1} \epsilon_{j+\Delta} \epsilon_j^* \epsilon_{j+\Delta+h-k}^* \epsilon_{j+h-k} \right]. \quad (4)$$

The estimator is unbiased if the expression in the square brackets is zero. Let  $w = k - h$  and note that  $w \neq 0$  and  $|w| < N$ . Using the periodicity it can be shown that the condition is equivalent to

$$0 = \frac{1}{L} \sum_{j=0}^{L-1} \epsilon_j \epsilon_{j+w}^* \epsilon_{j+\Delta}^* \epsilon_{j+\Delta+w}. \quad (5)$$

## 2.1 Bias of a random code

For a random code, the condition is not exactly zero

$$r = \frac{1}{L} \sum_{j=0}^{L-1} \epsilon_j \epsilon_{j+w}^* \epsilon_{j+\Delta}^* \epsilon_{j+\Delta+w}. \quad (6)$$

The variance is

$$\text{Var}[r] = \frac{1}{L}, \quad (7)$$

which means that the result will be biased on average by

$$b_{\Delta}^{(h)} = \sum_{k=0, k \neq h}^{N-1} \frac{R_{\Delta}^{(k)}}{\sqrt{L}} \quad (8)$$

However, one would use deconvolution (lag profile inversion) for a random code, this would remove the bias.

### 3 Optimizing the noise cancellation properties

The variance is calculated by

$$\langle |\hat{R}_\Delta^{(h)}|^2 \rangle = \frac{1}{L^2} \sum_{j_1=0}^{L-1} \sum_{j_2=0}^{L-1} \epsilon_{j_1} \epsilon_{j_1+\Delta} \epsilon_{j_2} \epsilon_{j_2+\Delta} \langle m_{j_1+\Delta+h} m_{j_1+h}^* m_{j_2+\Delta+h}^* m_{j_2+h} \rangle \quad (9)$$

using Isserli's theorem on the four product of Gaussian random numbers

$$\text{Var}[\hat{R}_\Delta^{(h)}] = \frac{1}{L^2} \sum_{j_1=0}^{L-1} \sum_{j_2=0}^{L-1} \epsilon_{j_1} \epsilon_{j_1+\Delta} \epsilon_{j_2} \epsilon_{j_2+\Delta} \langle m_{j_1+\Delta+h} m_{j_2+\Delta+h}^* \rangle \langle m_{j_2+h} m_{j_1+h}^* \rangle \quad (10)$$

and using  $\langle m_i m_j \rangle = 0$  yields the desired variance. Calculating the remaining expectation values

$$\langle m_{j_1}^* m_{j_2} \rangle = \sum_{k=0}^{N-1} \epsilon_{j_1-k} \epsilon_{j_2-k} R_{|j_1-j_2|}^{(k)} \quad (11)$$

and substituting it into the previous expression result in

$$\text{Var}[\hat{R}_\Delta^{(h)}] = \frac{1}{L^2} \sum_{j_1=0}^{L-1} \sum_{j_2=0}^{L-1} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} \hat{\epsilon}_{j_1} \hat{\epsilon}_{j_2} R_{|j_1-j_2|}^{(k_1)} R_{|j_1-j_2|}^{(k_2)} \quad (12)$$

where

$$\hat{\epsilon}_j = \epsilon_j \epsilon_{j+\Delta} \epsilon_{j+h-k_2} \epsilon_{j+\Delta+h-k_1} \quad (13)$$

was defined for brevity. By making use of the symmetry in  $j_1 \leftrightarrow j_2$  it can be shown that

$$\text{Var}[\hat{R}_\Delta^{(h)}] = \frac{1}{L} \left( \sum_{k_1=0}^{N-1} R_0^{(k_1)} \right)^2 + \frac{2}{L^2} \sum_{j_2=1}^{L-1} \sum_{j_1=0}^{L-1-j_2} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} \hat{\epsilon}_{j_1} \hat{\epsilon}_{j_1+j_2} R_{|j_2|}^{(k_1)} R_{|j_2|}^{(k_2)}. \quad (14)$$

Now, the unknown product of  $R$  variables does no longer depend  $j_1$ , but the codes are summed over  $j_1$ . For a random code of infinite length, the sum over  $j_1$  will have zero expectation value. However, for a finite code the sum over  $j_1$  can be negative, yielding additional noise suppression. The design of a code with maximal noise suppression appears to depend on the measurement results themselves. All codes which fulfill the bias free condition yield basically the same amount of noise suppression and appear like a random code to the second term. This gives effectively a standard deviation proportional to  $N/L^{3/2}$  because  $R$  drops exponentially with increasing  $j_2$ . So a long code might be better than a short periodic one.

### 4 Covariance with correct calculation of real part - also use this for variance

$$\langle \text{Re}(\hat{R}_{\Delta_1}^{(h)}) \text{Re}(\hat{R}_{\Delta_2}^{(h)}) \rangle = \frac{1}{L^2} \sum_{j_1=0}^{L-1} \sum_{j_2=0}^{L-1} \epsilon_{j_1} \epsilon_{j_1+\Delta_1} \epsilon_{j_2} \epsilon_{j_2+\Delta_2} \langle \text{Re}(m_{j_1+\Delta_1+h} m_{j_1+h}^*) \text{Re}(m_{j_2+\Delta_2+h}^* m_{j_2+h}) \rangle \quad (15)$$

$$2 \langle \text{Re} (m_{j_1+\Delta_1+h} m_{j_1+h}^*) \text{Re} (m_{j_2+\Delta_2+h}^* m_{j_2+h}) \rangle = \text{Re} (\langle m_{j_1+\Delta_1+h} m_{j_1+h}^* m_{j_2+\Delta_2+h}^* m_{j_2+h} \rangle) + \text{Re} (\langle m_{j_1+\Delta_1+h}^* m_{j_1+h} m_{j_2+\Delta_2+h}^* m_{j_2+h} \rangle)$$

plugged into the previous expression this gives

$$\begin{aligned} \langle \text{Re} (\hat{R}_{\Delta_1}^{(h)}) \text{Re} (\hat{R}_{\Delta_2}^{(h)}) \rangle &= \frac{1}{2} \text{Re} (\langle \hat{R}_{\Delta_1}^{(h)} \rangle^* \langle \hat{R}_{\Delta_2}^{(h)} \rangle) + \frac{1}{2} \text{Re} (\langle \hat{R}_{\Delta_1}^{(h)} \rangle \langle \hat{R}_{\Delta_2}^{(h)} \rangle) + \\ &\quad \frac{1}{2L^2} \sum_{j_1=0}^{L-1} \sum_{j_2=0}^{L-1} \epsilon_{j_1} \epsilon_{j_1+\Delta_1} \epsilon_{j_2} \epsilon_{j_2+\Delta_2} A \end{aligned}$$

$$A = \text{Re} (\langle m_{j_2+\Delta_2+h}^* m_{j_1+\Delta_1+h} \rangle \langle m_{j_1+h}^* m_{j_2+h} \rangle) + \text{Re} (\langle m_{j_1+\Delta_1+h}^* m_{j_2+h} \rangle \langle m_{j_2+\Delta_2+h}^* m_{j_1+h} \rangle)$$

$$\text{Cov} [\text{Re} (\hat{R}_{\Delta_1}^{(h)}), \text{Re} (\hat{R}_{\Delta_2}^{(h)})] = \frac{1}{2L^2} \sum_{j_1=0}^{L-1} \sum_{j_2=0}^{L-1} \epsilon_{j_1} \epsilon_{j_1+\Delta_1} \epsilon_{j_2} \epsilon_{j_2+\Delta_2} A$$

$$\hat{\epsilon}_{j,\Delta,l_1,l_2} = \epsilon_j \epsilon_{j+\Delta} \epsilon_{j+h-l_2} \epsilon_{j+\Delta+h-l_1}$$

$$\text{Cov} [\text{Re} (\hat{R}_{\Delta_1}^{(h)}), \text{Re} (\hat{R}_{\Delta_2}^{(h)})] = \frac{1}{2L^2} \sum_{j_1=0}^{L-1} \sum_{j_2=0}^{L-1} \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} (A + B)$$

$$A = \hat{\epsilon}_{j_1,\Delta_1,k_1,k_2} \hat{\epsilon}_{j_2,\Delta_2,k_1,k_2} \text{Re} (R_{j_2-j_1+\Delta_2-\Delta_1}^{(k_1)} R_{j_2-j_1}^{(k_2)}) \quad (16)$$

$$B = \hat{\epsilon}_{j_1,\Delta_1,k_1,k_2} \hat{\epsilon}_{j_2,\Delta_2,k_2,k_1} \text{Re} (R_{j_1-j_2+\Delta_1}^{(k_1)} R_{j_2-j_1+\Delta_2}^{(k_2)}) \quad (17)$$

## 5 Solving the code condition

The condition given by (5) is trivially fulfilled for pseudo random numbers if  $L \rightarrow \infty$ . However, the measurement would be biased and have no (definitive) time resolution in this case. With a brute-force computer search it can be shown that solutions do exist, when a finite  $\Delta \in [1, M]$  is assumed, where  $M$  is the total number of measurement points in the auto correlation function. If a code fulfills the condition it will be referred to as a valid code. Note that an inversion or a (cyclic) shift of any valid code is also a valid code. Thus, only codes which start with a + (or -) must be checked. Further, all codes which do not contain the sequence +- must be a constant sequence. Thus is sufficient to check codes which start with + and end with -. However there are still many codes which are cyclic versions of other codes.

### 5.1 Number of codes

For each  $L$  periodic code, there are  $L$  shifted versions, but how many  $L$  periodic codes are there? If  $L = 6$  there are also  $2^3$  3-periodic codes and  $2^2$  2-periodic codes. However, in both cases we must subtract the amount of 1 periodic codes, yielding the recursive equation

$$N_6 = 2^6 - N_3 - N_2 \quad (18)$$

where  $N_L$  is the number of  $L$  periodic codes. The general equation is

$$N_L = 2^L - \sum_{i \in D(L)} N_i \quad (19)$$

where  $D(L)$  is the set of even divisors of  $L$ .

### 5.2 Building the codes

We will introduce the compressed notation of a code here. For example: 11001110 is 2231 in the compressed notation. We want to generate codes which do not

1. have a periodicity smaller than  $L$ ,
2. exist multiple times as shifted versions and
3. exist another time as inverted version.

If we generate a code from its compressed version we can ensure the last condition by starting always with  $-$  (or  $+$ ). To ensure (2) we must realize that the number of digits in the compressed version must also be even so that the code does not start and end with the same sign. Furthermore, any code has a longest sequence  $\omega$  of ones. In the case that the code contains a sequence of zeros longer than the longest sequence of ones, we consider its inverse. We can always place this sequence at the beginning, which means that in the compressed version the largest digit always comes first. However, a sequence of length  $\omega$  might occur multiple times in the code, which can lead to a violation of (1). The solution is to split the code so that each sub-sequence starts with  $\omega$  and the digits which follow are itself a sequence of the type discussed here.

6 51 42 33 2211 111111

### 5.3 The autocorrelation of a code

Let

$$\gamma_{j,i} = \epsilon_{j+i}^* \epsilon_j \quad (20)$$

so that the autocorrelation function of the code can be written as

$$s_i = \sum_{j=0}^{L-1} \gamma_{j,i}. \quad (21)$$

We want to satisfy conditions of the type  $s_i = 0$  for certain values of  $i$ , which requires solving the inverse problem, which is, calculating  $\epsilon$  given a certain  $\gamma$ . The recursive equation

$$\epsilon_{j+i}^* = \gamma_{j,i} \epsilon_j \quad (22)$$

follows directly from the definition of  $\gamma$ . It can be solved by plugging it into itself till  $\epsilon_j$  is encountered again, which yields the following condition for the existence of a solution

$$\gamma_{k,i} = \prod_{j=1}^{L/d-1} \gamma_{jd+k,i} \quad (23)$$

where  $d = d(i)$  is the greatest common denominator of  $L$  and  $i$  and  $k \in [1, d]$ . Thus, we have  $d$  conditions, which removes  $d$  degrees of freedom from  $\gamma_{j,i}$ , meaning that for each value of  $i$ , a number of  $L-d$  bits can be chosen arbitrarily without violating the condition above.

We have now formally solved the problem for a single, fixed value of  $i$ . In order to satisfy a problem in which two values of  $i$  are allowed, called  $i_1$  and  $i_2$ , we calculate the sets of corresponding solutions  $\gamma_{j,i}$  and  $\gamma'_{j,i'}$  first.

$$s'_i s_i = \sum_{j'=0}^{L-1} \sum_{j=0}^{L-1} \gamma'_{j',i} \gamma_{j,i}. \quad (24)$$

#### 5.4 Connections between $\gamma$

$$\gamma_{j,i_1} \gamma_{j,i_2} = \epsilon_{j+i_2}^* \epsilon_j \epsilon_{j+i_1}^* \epsilon_j = \gamma_{j+i_1, i_2-i_1} \quad (25)$$

$$\gamma_{j+i,i} \gamma_{j,i} = \gamma_{j,2i} \quad (26)$$

in compressed form each digit must be smaller than 3 1111111111121212222211  
because we must fulfill  $\Delta = w = 1$  1X1 contributes +1, 12 and 21  
in a simple compressed form two adjacent 0 symbols are not allowed 101010100  
111111101

no optimization  $2^6$  with double one in the beginning  $2^4$  6 51 42, 411 (411 not valid! (cyclic shift)) 321, 312 (not valid! cyclic shift!), 3111 21111 111111 with cyclic elimination  $2^3$  8 3221, 3122, 3212, 31121  
31122312

$$+ - + - + + - + + +$$

decomposition into products of codes (advantage: stays on ring)  $+++++ = \begin{matrix} + \\ + \\ + \\ + \\ + \end{matrix}$   
 $+++++ - ++++- = \begin{matrix} + \\ + \\ + \\ + \\ - \end{matrix}$   $++++- - +++- = \begin{matrix} + \\ + \\ + \\ - \\ - \end{matrix}$   $+++-- - +-+ = \begin{matrix} + \\ + \\ - \\ - \\ - \end{matrix}$  each row is only allowed to combine (by multiply) with uneven number of rows above it

$$+-+--+, ++-++-, +++++- \text{ (5 periodicity for } L=10)$$

n periodicity times m periodicity code is a  $n*m$  periodicity code

when code has length L, the code can NOT be expressed as a product of codes which lengths are the prime factors of L.

++- (satisfies j=1)

the code  $\pm$ — is already a basis for all codes of the same length

longest run in the beginning followed by an uneven number of runs 511111 5311  
5 3 2 5 4

for s  $\geq$  0: 5:  $-\min(s, 3)$  3:  $-\min(s, 2)$  2:  $-\min(s, 2)$  4:  $-\min(s, 4) + \min(s - 5, 0)$

$$s_j = \sum_{i=1}^L a_i a_{i+j} = -L \quad (28)$$

$$s_j = \sum_{i=1}^L a_i b_i a_{i+j} b_{i+j} = - \sum_{i=1}^L b_i b_{i+j} \quad (29)$$

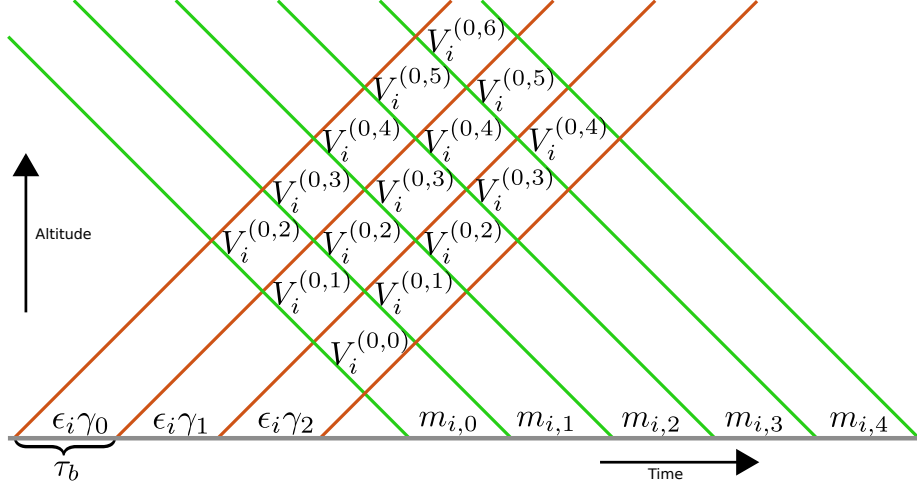


Figure 2: Range time diagram of the modulated version of the radar code with  $S = 3$ .

if  $a$  is  $j$  periodic, thus it is not necessary to consider any  $j$  periodic components if  $s_j = 0$  is demanded.

$$s_j = \sum_{i=1}^L (a_i + a_{i+j})^2 b_i b_{i+j} = - \sum_{i=1}^L b_i b_{i+j} \quad (30)$$

$$s_j = \sum_{i=1}^L \epsilon_i \epsilon_{i+j} = \sum_{i=1}^L |\epsilon_i + \epsilon_{i+j}| - 1 \geq 2W - L \quad (31)$$

$$W = \sum_{i=1}^L \epsilon_i \leq (s_j + L)/2 \quad (32)$$

## 6 Modulating the code

### 6.1 Modulation with fixed code, match filter on pulse, ignoring voltage changes

In the following it is considered that the recieved modulated voltage is demodulated first with an inverse filter and then correlated. This leads to a nice analytical expression, but only if the voltage during one pulse is considered constant, which is wrong...

$$m_{i,w} = \sum_{k=0}^{N-1} \epsilon_{i-k} \sum_{s=0}^{S-1} \gamma_{S-1-s} V_i^{(k,s+w)} \quad (33)$$

$$d_{i,w} = \sum_{s=0}^{S-1} \gamma_s m_{i,s+w} \quad (34)$$

$$w \in [0, K - S] \quad (35)$$



Where  $K$  is the number of bits received.

$$d_{i,w} = \sum_{k=0}^{N-1} \epsilon_{i-k} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} \gamma_{s_2} \gamma_{S-1-s_1} V_i^{(k, s_1+s_2+w)} \quad (36)$$

$$d_{i,w} = \sum_{k=0}^{N-1} \epsilon_{i-k} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} \gamma_{s_1} \gamma_{s_2} V_i^{(k, S-1-s_1+s_2+w)} \quad (37)$$

$$d_{i,w} = \sum_{k=0}^{N-1} \epsilon_{i-k} \left[ S V_i^{(k, S-1+w)} + \sum_{s_1=1}^{S-1} \sum_{s_2=0}^{S-1-s_1} \gamma_{s_2} \gamma_{s_1+s_2} V_i^{(k, S-1+s_1+w)} + \sum_{s_1=1}^{S-1} \sum_{s_2=0}^{S-1-s_1} \gamma_{s_2+s_1} \gamma_{s_2} V_i^{(k, S-1-s_1+w)} \right] \quad (38)$$

$$A_{s_1} = \sum_{s_2=0}^{S-1-s_1} \gamma_{s_2+s_1} \gamma_{s_2} \text{ for } s_1 \in [0, S-1] \quad (39)$$

$$d_{i,w} = \sum_{k=0}^{N-1} \epsilon_{i-k} \sum_{s=1-S}^{S-1} A_{|s|} V_i^{(k, S-1+s+w)} \quad (40)$$

$$\hat{R}_{\Delta}^{(h,w)} = \frac{1}{L} \sum_{j=0}^{L-1} \epsilon_{j+\Delta} d_{j+\Delta+h,w}^* \epsilon_j d_{j+h,w} \quad (41)$$

$$\hat{R}_{\Delta}^{(h,w)} = \frac{1}{L} \sum_{j=0}^{L-1} \epsilon_{j+\Delta} \sum_{k_1=0}^{N-1} \epsilon_{j+\Delta+h-k_1} \sum_{s_1=1-S}^{S-1} A_{|s_1|} \left( V_{j+\Delta+h}^{(k_1, S-1+s_1+w)} \right)^* \epsilon_j \sum_{k_2=0}^{N-1} \epsilon_{j+h-k_2} \sum_{s_2=1-S}^{S-1} A_{|s_2|} V_{j+h}^{(k_2, S-1+s_2+w)} \quad (42)$$

$$\langle \hat{R}_{\Delta}^{(h,w)} \rangle = \frac{1}{L} \sum_{j=0}^{L-1} \epsilon_{j+\Delta} \sum_{k=0}^{N-1} \epsilon_{j+\Delta+h-k} \sum_{s_1=1-S}^{S-1} A_{|s_1|} \epsilon_j \epsilon_{j+h-k} \sum_{s_2=1-S}^{S-1} A_{|s_2|} \left\langle \left( V_{j+\Delta+h}^{(k, S-1+s_1+w)} \right)^* V_{j+h}^{(k, S-1+s_2+w)} \right\rangle \quad (43)$$

$$\left\langle \left( V_{j+\Delta+h}^{(k, S-1+s_1+w)} \right)^* V_{j+h}^{(k, S-1+s_2+w)} \right\rangle = \delta_{s_1, s_2} R_{\Delta}^{(k, S-1+s_1+w)} + \delta_{s_1, s_2+1} R_{\Delta}^{(k, S-1+s_1+w+0.5)} / 2 + \delta_{s_1, s_2-1} R_{\Delta}^{(k, S-1+s_1+w-0.5)} \quad (44)$$

$$\langle \hat{R}_{\Delta}^{(h,w)} \rangle = \sum_{s_1=1-S}^{S-1} A_{|s_1|}^2 R_{\Delta}^{(h, S-1+s_1+w)} + \frac{1}{2} \sum_{s_1=1-S}^{S-2} A_{|s_1|} A_{|s_1+1|} R_{\Delta}^{(h, S-1+s_1+w+0.5)} + \frac{1}{2} \sum_{s_1=2-S}^{S-1} A_{|s_1|} A_{|s_1-1|} R_{\Delta}^{(h, S-1+s_1+w-0.5)} \quad (45)$$

if  $\gamma$  is a Barker code we have  $A_{|i|} A_{|i+1|} = 0$

$$\langle \hat{R}_{\Delta}^{(h,w)} \rangle = \sum_{s_1=1-S}^{S-1} A_{|s_1|}^2 R_{\Delta}^{(h, S-1+s_1+w)} \quad (46)$$

so we find a biased estimator, where the signal is stronger by a factor

$$\frac{S^2}{S-1} \quad (47)$$

compared to the bias. one can also deconvolute, to find an exact result.

## 6.2 Modulation with fixed code, match filter on pulse

$$m_{i,w} = \sum_{k=0}^{N-1} \epsilon_{i-k} \sum_{s=0}^{S-1} \gamma_{S-1-s} V_{i,w}^{(k,s+w)} \quad (48)$$

$$d_{i,w} = \sum_{s=0}^{S-1} \gamma_s m_{i,s+w} \quad (49)$$

$$w \in [0, K - S] \quad (50)$$

Where  $K$  is the number of bits received.

$$d_{i,w} = \sum_{k=0}^{N-1} \epsilon_{i-k} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} \gamma_{s_2} \gamma_{S-1-s_1} V_{i,w+s_2}^{(k,s_1+s_2+w)} \quad (51)$$

$$d_{i,w} = \sum_{k=0}^{N-1} \epsilon_{i-k} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} \gamma_{s_1} \gamma_{s_2} V_{i,w+s_2}^{(k,S-1-s_1+s_2+w)} \quad (52)$$

In order to get rid of half overlapping range gates we use

$$V_{i,w}^{(k,s)} = X_{i,w}^{(L,k,s)} + X_{i,w}^{(U,k,s+1)} \quad (53)$$

with

$$\left\langle \left( X_{i_1,w_1}^{(z_1,k,s_1)} \right)^* X_{i_2,w_2}^{(z_2,k,s_2)} \right\rangle = \delta_{s_1,s_2} G_{\Delta,w_1-w_2}^{(z_1,z_2,k,s_1)} \quad (54)$$

with this we have

$$\left\langle \left( V_{i_1,w_1+s_2}^{(k,S-1-s_1+s_2+w_1)} \right)^* V_{i_2,w_2+s_4}^{(k,S-1-s_3+s_4+w_2)} \right\rangle = \quad (55)$$

$$G_{\Delta,w_1-w_2+s_2-s_4}^{(L,L,S-1-s_1+s_2+w_1)} \delta_{s_2-s_1+w_1,s_4-s_3+w_2} \quad (56)$$

$$+ G_{\Delta,w_1-w_2+s_2-s_4}^{(U,U,S-1-s_1+s_2+w_1)} \delta_{s_2-s_1+w_1,s_4-s_3+w_2} \quad (57)$$

$$+ G_{\Delta,w_1-w_2+s_2-s_4}^{(L,U,S-1-s_1+s_2+w_1)} \delta_{s_2-s_1+w_1,s_4-s_3+w_2+1} \quad (58)$$

$$+ G_{\Delta,w_1-w_2+s_2-s_4}^{(U,L,S-1-s_1+s_2+w_1)} \delta_{s_2-s_1+w_1+1,s_4-s_3+w_2} \quad (59)$$

$$\left\langle \left( V_{i_1,w_1}^{(k,s_1)} \right)^* V_{i_2,w_2}^{(k,s_2)} \right\rangle = \left( G_{\Delta,\delta}^{(L,L,s_1)} + G_{\Delta,\delta}^{(U,U,s_1+1)} \right) \delta_{s_1,s_2} + G_{\Delta,\delta}^{(L,U,s_1)} \delta_{s_1,s_2+1} + G_{\Delta,\delta}^{(U,L,s_1+1)} \delta_{s_1+1,s_2} \quad (60)$$

we see that

$$\left( G_{i,w_1-w_2}^{(z_1,z_2,k,s_1)} \right)^* = G_{i,w_1-w_2}^{(z_2,z_1,k,s_1)} \quad (61)$$

using this we have

$$\left\langle \left( V_{i_1,w_1+s_2}^{(k,S-1-s_1+s_2+w_1)} \right)^* V_{i_2,w_2+s_4}^{(k,S-1-s_3+s_4+w_2)} \right\rangle = \quad (62)$$

$$\left( G_{\Delta,w_1-w_2+s_2-s_4}^{(U,U,S-1-s_1+s_2+w_1)} + G_{\Delta,w_1-w_2+s_2-s_4}^{(L,L,S-1-s_1+s_2+w_1)} \right) \delta_{s_2-s_1+w_1,s_4-s_3+w_2} \quad (63)$$

$$+ G_{\Delta,w_1-w_2+s_2-s_4}^{(L,U,S-1-s_1+s_2+w_1)} \delta_{s_2-s_1+w_1,s_4-s_3+w_2+1} \quad (64)$$

$$+ \left( G_{\Delta,w_1-w_2+s_2-s_4}^{(L,U,S-1-s_1+s_2+w_1)} \right)^* \delta_{s_2-s_1+w_1+1,s_4-s_3+w_2} \quad (65)$$

### 6.3 Modulation with fixed code, decoding first

$$\langle m_{i_1, w_1}^* m_{i_2, w_2} \rangle = \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} \gamma_{s_1} \gamma_{s_2} \left\langle \left( V_{i_1, w_1}^{(k, S-1-s_1+w_1)} \right)^* V_{i_2, w_2}^{(k, S-1-s_2+w_2)} \right\rangle \quad (66)$$

### 6.4 Modulation with alternating code

$$m_{i, w} = \sum_{k=0}^{N-1} \epsilon_{i-k} \sum_{s=0}^{S-1} \gamma_{S-1-s}^{(k)} V_i^{(k, s+w)} \quad (67)$$

$$\hat{R}_{\Delta}^{(h, w)} = \frac{1}{L} \sum_{j=0}^{L-1} \epsilon_{j+\Delta} m_{j+\Delta+h, w}^* \epsilon_j m_{j+h, w} \quad (68)$$

$$\langle \hat{R}_{\Delta}^{(h, w)} \rangle = \frac{1}{L} \sum_{s_1=0}^{S-1} \sum_{s_2=0}^{S-1} \sum_{j=0}^{L-1} \sum_{k=0}^{N-1} \epsilon_{j+\Delta} \epsilon_{j+\Delta+h-k} \gamma_{S-1-s_1}^{(j+\Delta+h-k)} \epsilon_j \epsilon_{i-k} \gamma_{S-1-s_2}^{(j+h-k)} \left\langle \left( V_{j+\Delta+h}^{(k, s_1+w)} \right)^* V_{j+h}^{(k, s_2+w)} \right\rangle \quad (69)$$

## 7 The alternating codes

### 7.1 Short lags expectation value

$$m_j^{(c)} = \sum_{i=0}^{B-1} \epsilon_i^{(c)} V_{i, j}^{(c)} + \xi_j^{(c)} \quad (70)$$

$$\hat{R}_{\Delta}^{(h(j_1-k_1))} = \frac{1}{M} \sum_{c=0}^{M-1} \epsilon_{k_1}^{(c)} \epsilon_{k_2}^{(c)} \left( m_{j_1}^{(c)} \right)^* m_{j_2}^{(c)} \quad (71)$$

$$\langle \hat{R}_{\Delta}^{(j_1-k_1)} \rangle = \frac{1}{M} \sum_{c=0}^{M-1} \epsilon_{k_1}^{(c)} \epsilon_{k_2}^{(c)} \left\langle \left( m_{j_1}^{(c)} \right)^* m_{j_2}^{(c)} \right\rangle = \frac{1}{M} \sum_{i_1=0}^{B-1} \sum_{i_2=0}^{B-1} \sum_{c=0}^{M-1} \epsilon_{k_1}^{(c)} \epsilon_{k_2}^{(c)} \epsilon_{i_1}^{(c)} \epsilon_{i_2}^{(c)} \left\langle \left( V_{i_1, j_1}^{(c)} \right)^* V_{i_2, j_2}^{(c)} \right\rangle \quad (72)$$

$$\Delta = k_1 - k_2 = j_1 - j_2 > 0 \quad (73)$$

### 7.2 Long lags expectation value

$$m_j^{(c)} = \sum_{i=0}^{B-1} \epsilon_i^{(c)} V_{i, j}^{(c)} + \xi_j^{(c)} \quad (74)$$

$$\hat{R}_{\Delta}^{(h(j_1-k_1))} = \frac{1}{M} \sum_{c=0}^{M-1} \epsilon_{k_1}^{(c)} \epsilon_{k_2}^{(c+1)} \left( m_{j_1}^{(c)} \right)^* m_{j_2}^{(c+1)} \quad (75)$$

$$\langle \hat{R}_{\Delta}^{(j_1-k_1)} \rangle = \frac{1}{M} \sum_{c=0}^{M-1} \epsilon_{k_1}^{(c)} \epsilon_{k_2}^{(c+1)} \left\langle \left( m_{j_1}^{(c)} \right)^* m_{j_2}^{(c+1)} \right\rangle = \frac{1}{M} \sum_{i_1=0}^{B-1} \sum_{i_2=0}^{B-1} \sum_{c=0}^{M-1} \epsilon_{k_1}^{(c)} \epsilon_{k_2}^{(c+1)} \epsilon_{i_1}^{(c)} \epsilon_{i_2}^{(c+1)} \left\langle \left( V_{i_1, j_1}^{(c)} \right)^* V_{i_2, j_2}^{(c+1)} \right\rangle \quad (76)$$

$$\Delta = k_1 - k_2 = j_1 - j_2 > 0 \quad (77)$$

### 7.3 Bias

the estimator is biased, due to signals coming from higher altitudes, which are caused by the pulse  $c - 1$

$$\langle \hat{R}_{\Delta}^{(j_1-k_1)} \rangle = \frac{1}{M} \sum_{i_1=0}^{B-1} \sum_{i_2=0}^{B-1} \sum_{c=0}^{M-1} \epsilon_{k_1}^{(c)} \epsilon_{k_2}^{(c)} \epsilon_{i_1}^{(c-1)} \epsilon_{i_2}^{(c-1)} \langle (V_{i_1, j_1}^{(c-1,1)})^* V_{i_2, j_2}^{(c-1,1)} \rangle \quad (78)$$

SPECULATION: as the product of codes is random, we do not expect constant 1 but 0 with standard deviation of  $1/\sqrt{B}$  for all indices for which the expectation value is not zero. the additional (implicit) averaging over  $B$  will contribute another factor  $1/\sqrt{B}$

### 7.4 Variance

$$\langle |\hat{R}_{\Delta}^{(j_1-k_1)}|^2 \rangle = \frac{1}{M^2} \sum_{c_1=0}^{M-1} \sum_{c_2=0}^{M-1} \epsilon_{k_1}^{(c_1)} \epsilon_{k_2}^{(c_1)} \epsilon_{k_1}^{(c_2)} \epsilon_{k_2}^{(c_2)} \langle m_{j_1}^{(c_1)} (m_{j_2}^{(c_1)})^* (m_{j_1}^{(c_2)})^* m_{j_2}^{(c_2)} \rangle \quad (79)$$

$$\text{Var} [\hat{R}_{\Delta}^{(j_1-k_1)}] = \frac{1}{M^2} \sum_{c_1=0}^{M-1} \sum_{c_2=0}^{M-1} \epsilon_{k_1}^{(c_1)} \epsilon_{k_2}^{(c_1)} \epsilon_{k_1}^{(c_2)} \epsilon_{k_2}^{(c_2)} \langle m_{j_1}^{(c_1)} (m_{j_1}^{(c_2)})^* \rangle \langle (m_{j_2}^{(c_1)})^* m_{j_2}^{(c_2)} \rangle \quad (80)$$

$$\text{Var} [\hat{R}_{\Delta}^{(j_1-k_1)}] \approx \frac{1}{M} \langle m_{j_1}^{(c)} (m_{j_1}^{(c)})^* \rangle \langle (m_{j_2}^{(c)})^* m_{j_2}^{(c)} \rangle \quad (81)$$

$$\langle (m_j^{(c)})^* m_j^{(c)} \rangle = \sum_{i_1=0}^{B-1} \sum_{i_2=0}^{B-1} \epsilon_{i_1}^{(c)} \epsilon_{i_2}^{(c)} \langle V_{i_1, j}^{(c)} (V_{i_2, j}^{(c)})^* \rangle + P_n \approx BP + P_n \quad (82)$$

$$\text{Var} [\hat{R}_{\Delta}^{(j_1-k_1)}] = \frac{(BP + P_n)^2}{M} \quad (83)$$

## 8 Hypothesis on code length

## 9 Code with cyclic difference set for $\tau_i$

## 10 Simulation

### 10.1 Monte Carlo Generation

The most straight-forward approach to generate random numbers following a wide-sense stationary process is as follows:

$$R_{jk}^{(h)} = \langle (V_j^{(h)})^* V_k^{(h)} \rangle \quad (84)$$

where  $R_{jk}^{(h)}$  is interpreted (for some fixed  $h$ ) as a correlation matrix for the random vector  $V_j^{(h)}$  which is to be generated. We can satisfy the correlation properties by a linear transformation

$$V_j^{(h)} = \sum_i T_{ji} f_i \quad (85)$$

where  $T_{ji}$  must be of toeplitz type and  $f_i$  are uncorrelated standard normal Gaussian circular-symmetric random variables with  $\langle f_i f_j \rangle = \delta_{ij}$ . This yields

$$R_{jk}^{(h)} = \sum_{i,l} \langle (T_{ji} f_i)^* T_{kl} f_l \rangle = \sum_i T_{ji}^* T_{ki}. \quad (86)$$

Using that the matrix  $T$  is symmetric and seeing that  $R_{jk}^{(h)}$  must be positive semidefinite (as it is a covariance matrix) we can calculate  $T$  with some square root of matrix algorithm.

However all of this can be done more efficiently by using the time stationary of the process. Note that  $T$  is of toeplitz type simply means that we actually need a convolution of  $f$  with  $T$  where  $T$  must be symmetric with a length of at least  $2M+1$  so that the points up to  $M+1$  can be correlated. However, the generated time series would not have the true correlation properties of the ACF if it is truncated after  $M+1$ . The ACF can be either truncated after  $n-1$ , where  $n$  is the total length of the time series, or when the ACF approaches approximately zero. Now, denote the length of the ACF by  $K$ . The convolution is given by

$$V_j^{(h)} = \sum_{i=1-K}^{K-1} T_i f_{j-i}. \quad (87)$$

It follows that,

$$R_{jk}^{(h)} = \langle (V_j^{(h)})^* V_k^{(h)} \rangle = \sum_{l=1-K}^{K-1} \sum_{m=1-K}^{K-1} T_l T_m \langle f_{j-l} f_{k-m} \rangle. \quad (88)$$

Using that  $\langle f_{j-l} f_{k-m} \rangle = \delta_{m,k-j+l}$  and substituting  $\Delta = j - k$  yields

$$R_{\Delta}^{(h)} = \sum_{i=1-K}^{K-1} T_i T_{i-\Delta} \quad (89)$$

where it is assumed that  $T$  is zero padded for all indices which fall out of range. Applying a Fourier transform over  $\Delta$  on both sides results in

$$\text{DCT}[R^{(h)}]_j = \sum_{i=1-K}^{K-1} T_i \text{DTFT}[T_{i-\Delta}]_j. \quad (90)$$

From the definition of the DTFT follows that

$$\text{DCT}[R^{(h)}]_j = \sum_{l=1-K}^{K-1} T_l \exp\left(\frac{i2\pi lj}{2K-1}\right) \text{DCT}[T]_j, \quad (91)$$

using the definition of the Fourier transform again results in

$$\text{DCT}[R^{(h)}]_j = (2K-1) (\text{DCT}[T]_j)^2. \quad (92)$$

Now there are two options: Either one solves for  $T$  and does the convolution in the time domain, or one solves for the Fourier transform of  $T$  and does the convolution in Fourier space. Which one is faster depends on the choice of  $K$ . The first approach is preferred when  $K$  is fixed which results in an asymptotic runtime of  $\mathcal{O}(Kn)$ . In the case  $K = n$  latter approach is preferred as it results in a runtime of  $\mathcal{O}(n \log(n))$ . However, choosing  $K = n$  has not a real practical use case, as the ACF can always be cut off at small value, in the sense that  $M \gtrsim K$ .

## 10.2 Discrete Fourier and Cosine Transform

We define the Discrete Time Fourier Transform as

$$\text{DTFT}[x]_n = \sum_{k=0}^{N-1} x_k \exp\left(-\frac{i2\pi kn}{N}\right) \quad (93)$$

and its inverse by

$$\text{DTFT}^{-1}[x]_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k \exp\left(\frac{i2\pi kn}{N}\right). \quad (94)$$

Sometimes, it can be handy to allow negative indices for a more “natural” representation. In the case that  $n$  is even

$$\text{DTFT}^{-1}[\hat{x}]_n = \frac{1}{N} \sum_{k=-N/2+2}^{N/2+1} \hat{x}_k \exp\left(\frac{i2\pi kn}{N}\right) \quad (95)$$

and in the uneven case

$$\text{DTFT}^{-1}[\hat{x}]_n = \frac{1}{N} \sum_{k=-(N-1)/2}^{(N-1)/2} \hat{x}_k \exp\left(\frac{i2\pi kn}{N}\right). \quad (96)$$

The Discrete Cosine Transform of Type I is defined as

$$\text{DCT}[x']_n = x'_0 + x'_{N'-1}(-1)^n + 2 \sum_{k=0}^{N'-1} x'_k \cos\left(\frac{\pi kn}{N'-1}\right). \quad (97)$$

and its inverse is given by

$$\text{DCT}^{-1}[x']_n = \frac{\text{DCT}[x']_n}{2(N'-1)} \quad (98)$$

The DCT is equivalent to a DTFT for a symmetric input but it is faster and uses less memory. More precisely, it can be shown that for  $N = 2(N' - 1)$  and  $x_i = x'_i$  for  $i \in [0, N' - 1]$  and  $x_i = x'_{N-i}$  for  $i \in [N', N - 1]$  that

$$\text{DTFT}^{-1}[x]_n = \text{DCT}^{-1}[x']_n \quad (99)$$

where  $n \in [0, N' - 1]$ .

## 10.3 From the continuous Fourier transform to the DCT

$$R(\tau) = \int_{-\infty}^{\infty} \exp(-2\pi i \omega \tau) S(\omega) d\omega \quad (100)$$

$$= 2 \int_0^{\infty} \cos(2\pi i \omega \tau) S(\omega) d\omega \quad (101)$$

$$S(j\Delta_\omega) = S_j \quad (102)$$

$$R(j\Delta_\tau) = R_j \quad (103)$$

As we only store a fixed number of samples, the period must be finite. This means, that a periodic continuation beyond the last sample ( $S_{N-1}$ ) must be specified. In our case, even symmetry around  $N-1$  is considered. As the first and last sample's bins shall be centered around  $\omega = 0$  and  $\omega = (N-1)\Delta_\omega$  respectively, they have half width, so the integral becomes

$$R_k = S_0\Delta_\omega + \cos(2\pi(N-1)k\Delta_\omega\Delta_\tau)S_{N-1} + 2 \sum_{j=1}^{j=N-2} \cos(2\pi jk\Delta_\omega\Delta_\tau)S_j\Delta_\omega \quad (104)$$

The maximum frequency (half bandwidth in our case) is  $B = \Delta_\omega(N-1)$ . Now we can use the sampling theorem to find  $\Delta_\tau = 1/(2B)$ .

$$R_k = \left( S_0 + (-1)^k S_{N-1} + 2 \sum_{j=1}^{j=N-2} \cos\left(\frac{\pi jk}{N-1}\right) S_j \right) \Delta_\omega \quad (105)$$

This is a type I DCT (Discrete Cosine Transform) according to scipy definition.

## 11 A lower bound for the variance

Assumptions made: monostatic radar, binary coded transmission, gaussian process A lower bound for the variance of any ACF estimation independent of the estimator is to be calculate, by means of the Cramér–Rao bound. What the radar measures (independent of the estimator) is

$$m_i = \sum_{k=0}^{N-1} \epsilon_{i-k} V_i^{(k)}. \quad (106)$$

The parameters to be estimated are

$$R_\Delta^{(k)} = \langle (V_{j+\Delta}^{(k)})^* V_j^{(k)} \rangle. \quad (107)$$

The individual  $m_i$  are random Gaussian distributed. The information is contained in the correlations. Define the probability density function  $P = P(m_{1..n}, R_{1..M}^{(0..N-1)})$ .

We need to calculate  $P$ . the random variables  $V_i^{(k)}$  are normally distributed with a correlation matrix of toeplitz type given by

$$\Sigma_{ij}^{(k)} = R_{i-j}^{(k)} \quad (108)$$

then  $m_i$  are Gaussian distributed according to

$$\Sigma_{ij} = \sum_{k=0}^{N-1} \epsilon_{i-k} \epsilon_{j-k} R_{|i-j|}^{(k)}. \quad (109)$$

we note that the matrix is periodic with  $\Sigma_{i,j} = \Sigma_{i+L,j+L}$  where we imagine the matrix to be infinite in size as the auto correlation never drops to precisely to zero. The fisher information matrix for any zero mean Gaussian is

$$\mathcal{I}_{ij} = \frac{1}{2} \text{tr} \left( \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_i} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_j} \right) \quad (110)$$

where each  $\theta_i$  identifies exactly one parameter  $\theta_i = R_{\Delta_i}^{(k_i)}$

$$\frac{\partial \Sigma_{ij}}{\partial \theta_p} = \epsilon_{i-k_p} \epsilon_{j-k_p} \delta_{|i-j|, \Delta_p} \quad (111)$$

the fisher information matrix is

$$\mathcal{I}_{ij} = \frac{1}{2} \sum_{k_1, k_2, k_3, k_4} \Sigma_{k_1, k_2}^{-1} \epsilon_{k_2-k_i} \epsilon_{k_3-k_i} \delta_{|k_2-k_3|, \Delta_i} \Sigma_{k_3, k_4}^{-1} \epsilon_{k_4-k_j} \epsilon_{k_1-k_j} \delta_{|k_4-k_1|, \Delta_j} \quad (112)$$

defining

$$\sigma_{\Delta_1, \Delta_2} = \epsilon_{k_2-k_i} \epsilon_{k_2+\Delta_1-k_i} \epsilon_{k_1-k_j} \epsilon_{k_1+\Delta_2-k_j} \Sigma_{k_1, k_2}^{-1} \Sigma_{k_2+\Delta_1, k_1+\Delta_2}^{-1} \quad (113)$$

swapping  $k_1 \leftrightarrow k_2$  and using symmetry of  $\Sigma^{-1}$  it can be shown that

$$\sigma_{\Delta_1, \Delta_2} = \epsilon_{k_1-k_i} \epsilon_{k_1+\Delta_1-k_i} \epsilon_{k_2-k_j} \epsilon_{k_2+\Delta_2-k_j} \Sigma_{k_1, k_2}^{-1} \Sigma_{k_1+\Delta_1, k_2+\Delta_2}^{-1} \quad (114)$$

using this definition results in

$$\mathcal{I}_{ij} = \frac{1}{2} \sum_{k_1, k_2} [\sigma_{\Delta_i, \Delta_j} + \sigma_{-\Delta_i, \Delta_j} + \sigma_{\Delta_i, -\Delta_j} + \sigma_{-\Delta_i, -\Delta_j}] \quad (115)$$

for an infinite length random code with high  $N$  the diagonal elements of  $\Sigma$  grow proportional to  $N$ , while the off-diagonal elements have zero expectation value. They do have a standard deviation of  $\sqrt{N}$  but for sufficient large  $N$  a diagonal matrix is found:

$$\Sigma_{ij} = \delta_{ij} \sum_{k=0}^{N-1} R_0^{(k)} \quad (116)$$

which has the trivial inverse

$$\Sigma_{ij}^{-1} = \delta_{ij} \left( \sum_{k=0}^{N-1} R_0^{(k)} \right)^{-1} \quad (117)$$

when substituting this into  $\sigma$  it can be seen that  $\Delta_i = \Delta_j$  and  $k_1 = k_2$  must hold for  $\sigma \neq 0$ .

$$\sigma_{\Delta_1, \Delta_2} = \delta_{k_1 k_2} \delta_{\Delta_1 \Delta_2} \epsilon_{k_1-k_i} \epsilon_{k_1+\Delta_1-k_i} \epsilon_{k_2-k_j} \epsilon_{k_2+\Delta_2-k_j} \left( \sum_{k=0}^{N-1} R_0^{(k)} \right)^{-2} \quad (118)$$

the epsilons will square away for  $k_i = k_j$  or special cases like  $\Delta_1 = \Delta_2 = k_j, k_i = 0$  but we are only interested in the main diagonal of the fisher matrix, so  $k_i = k_j$ . the fisher matrix is thus approximately given by

$$\mathcal{I}_{ij} = \delta_{ij} L \left( \sum_{k=0}^{N-1} R_0^{(k)} \right)^{-2} \quad (119)$$

where we assumed  $L$  measurements

so the lower bound for the standard deviation is

$$\sigma \geq \frac{1}{\sqrt{L}} \sum_{k=0}^{N-1} R_0^{(k)} \quad (120)$$



## 12 Calculation of codes

Due to technical limitations of the radar we have  $\epsilon_i \in [-1, 1]$  and a naive brute force algorithm finds the following (shortest) codes for a given  $N$  and  $M$

$L$	$N$	$M$	Code
24	5	4	- + - + + + + + + - + + - - + - + + - + + + -
32	6	5	- + + - + + + + + + + - - + + - - - + - - - + + - + - - -
32	8	3	- + - - + + + + + + + - + - - - + - - + + + + - - + + + - - -
40	7	4	- + - - + + + + + + + + - + + + - - + - + + + - - - + - + + + - - +
48	8	5	- - + + + + + + + + - + - + - - + - - + - + + - - + - - - + - - - + + - - - +

Note that we have used that every code which is valid for  $w$  is also valid for  $-w$ , because the value of the sum does not change if the code is shifted by  $w$ , due to the codes periodicity.

## 13 Possibility of using a Quantum SAT-Solver

The condition could be transformed to a boolean SAT problem, which could be solved by Grover's algorithm on a quantum computer. See <https://learn.qiskit.org/course/introduction/grovers-search-algorithm>. However, there are also different SAT-Solvers, which might pose a speedup compared to random guessing. There are also combinations of improved SAT-Solvers and quantum; investigate this.