

# Radar code documentation

Frank Hermann

March 3, 2022

## 1 Introduction

A binary phase radar code is presented which has optimal noise canceling properties, intrinsically measures the entire range at once (assuming that it is finite) and supports arbitrary radar duty cycles.

## 2 Mathematical description

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,

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

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  $\Delta$  is considered. 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

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

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

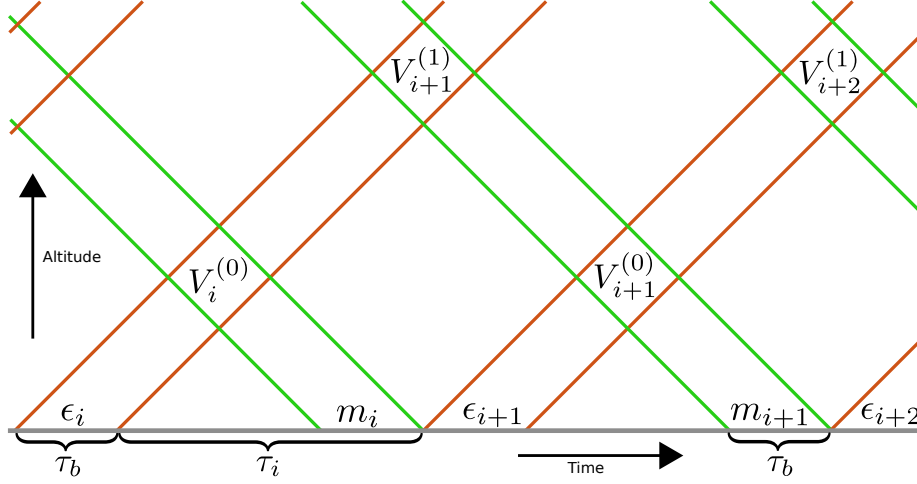


Figure 1: Range time diagram of the simplified version of the radar code.

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

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

In the next section, solutions of this equation are discussed.

### 3 Solving the code condition

The condition given by (5) is trivially fulfilled for pseudo random numbers if  $L \rightarrow \infty$ . However, the measurement would 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.

### 3.1 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

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

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

$$N_L = 2^L - \sum_{i \in D(L)} N_i \quad (7)$$

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

### 3.2 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

### 3.3 The autocorrelation of a code

Let

$$\gamma_{j,i} = \epsilon_{j+i}^* \epsilon_j \quad (8)$$

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

$$s_i = \sum_{j=0}^{L-1} \gamma_{j,i}. \quad (9)$$

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

$$\epsilon_{j+i}^* = \gamma_{j,i} \epsilon_j \quad (10)$$

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

$$\gamma_{k,i} = \prod_{j=1}^{L/d-1} \gamma_{jd+k,i} \quad (11)$$

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.

$$s'_i s_i = \sum_{j'=0}^{L-1} \sum_{j=0}^{L-1} \gamma'_{j',i} \gamma_{j,i}. \quad (12)$$

### 3.4 Connections between $\gamma$

$$\gamma_{j,i_1} \gamma_{j,i_2} = \epsilon_{j+i_2}^* \epsilon_j \epsilon_{j+i_1}^* \epsilon_j = \gamma_{j+i_1, i_2-i_1} \quad (13)$$

$$\gamma_{j+i,i} \gamma_{j,i} = \gamma_{j,2i} \quad (14)$$



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

$$s_j = \sum_{i=1}^L (a_i + a_{i+j})^2 b_i b_{i+j} = - \sum_{i=1}^L b_i b_{i+j} \quad (18)$$

$$s_j = \sum_{i=1}^L \epsilon_i \epsilon_{i+j} = \sum_{i=1}^L |\epsilon_i + \epsilon_{i+j}| - 1 \geq 2W - L \quad (19)$$

$$W = \sum_{i=1}^L \epsilon_i \leq (s_j + L)/2 \quad (20)$$

## 4 Modulating the code

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

## 6 Properties of the code

## 7 Simulation

### 7.1 Monte Carlo Generation

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

$$R_{jk}^{(h)} = \langle (V_j^{(h)})^* V_k^{(h)} \rangle \quad (21)$$

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

$$V_j^{(h)} = \sum_i T_{ji} f_i \quad (22)$$

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

$$R_{jk}^{(h)} = \sum_{i,l} \langle (T_{ji} f_i)^* T_{kl} f_l \rangle = \sum_i T_{ji}^* T_{ki}. \quad (23)$$

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

$$V_j^{(h)} = \sum_{i=1-K}^{K-1} T_i f_{j-i}. \quad (24)$$

It follows that,

$$R_{jk}^{(h)} = \langle (V_j^{(h)})^* V_k^{(h)} \rangle = \sum_{l=1-K}^{K-1} \sum_{m=1-K}^{K-1} T_l T_m \langle f_{j-l} f_{k-m} \rangle. \quad (25)$$

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

$$R_{\Delta}^{(h)} = \sum_{i=1-K}^{K-1} T_i T_{i-\Delta} \quad (26)$$

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

$$\text{DCT}[R^{(h)}]_j = \sum_{i=1-K}^{K-1} T_i \text{DTFT}[T_{i-\Delta}]_j. \quad (27)$$

From the definition of the DTFT follows that

$$\text{DCT}[R^{(h)}]_j = \sum_{l=1-K}^{K-1} T_l \exp\left(\frac{i2\pi lj}{2K-1}\right) \text{DCT}[T]_j, \quad (28)$$

using the definition of the Fourier transform again results in

$$\text{DCT}[R^{(h)}]_j = (2K-1) (\text{DCT}[T]_j)^2. \quad (29)$$

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  $M \gtrsim K$ .

## 7.2 Discrete Fourier and Cosine Transform

We define the Discrete Time Fourier Transform as

$$\text{DTFT}[x]_n = \sum_{k=0}^{N-1} x_k \exp\left(-\frac{i2\pi kn}{N}\right) \quad (30)$$

and its inverse by

$$\text{DTFT}^{-1}[x]_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k \exp\left(\frac{i2\pi kn}{N}\right). \quad (31)$$

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

$$\text{DTFT}^{-1}[\hat{x}]_n = \frac{1}{N} \sum_{k=-N/2+2}^{N/2+1} \hat{x}_k \exp\left(\frac{i2\pi kn}{N}\right) \quad (32)$$

and in the uneven case

$$\text{DTFT}^{-1}[x]_n = \frac{1}{N} \sum_{k=-(N-1)/2}^{(N-1)/2} x_k \exp\left(\frac{i2\pi kn}{N}\right). \quad (33)$$

The Discrete Cosine Transform of Type I is defined as

$$\text{DCT}[x']_n = x'_0 + x'_{N'-1}(-1)^n + 2 \sum_{k=0}^{N'-1} x'_k \cos\left(\frac{\pi kn}{N'-1}\right). \quad (34)$$

and its inverse is given by

$$\text{DCT}^{-1}[x']_n = \frac{\text{DCT}[x']_n}{2(N'-1)} \quad (35)$$

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

$$\text{DTFT}^{-1}[x]_n = \text{DCT}^{-1}[x']_n \quad (36)$$

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

### 7.3 From the continuous Fourier transform to the DCT

$$R(\tau) = \int_{-\infty}^{\infty} \exp(-2\pi i \omega \tau) S(\omega) d\omega \quad (37)$$

$$= 2 \int_0^{\infty} \cos(2\pi i \omega \tau) S(\omega) d\omega \quad (38)$$

$$S(j\Delta_\omega) = S_j \quad (39)$$

$$R(j\Delta_\tau) = R_j \quad (40)$$

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

$$R_k = S_0\Delta_\omega + \cos(2\pi(N-1)k\Delta_\omega\Delta_\tau)S_{N-1} + 2 \sum_{j=1}^{j=N-2} \cos(2\pi jk\Delta_\omega\Delta_\tau)S_j\Delta_\omega \quad (41)$$



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

$$R_k = \left( S_0 + (-1)^k S_{N-1} + 2 \sum_{j=1}^{j=N-2} \cos\left(\frac{\pi j k}{N-1}\right) S_j \right) \Delta_\omega \quad (42)$$

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

## 8 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.

## 9 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.