

Research Project and Seminar

Informatik-Ingenieurwesen

Orthogonal Codes for Acoustic Underwater Localization

by

Sergej Keller

April 2022

Supervised by

Christoph Weyer

Institute of Telematics, Hamburg University of Technology

First Examiner

Prof. Dr.-Ing. Bernd-Christian Renner

Institute of Autonomous Cyber-Physical Systems
Hamburg University of Technology

Second Examiner

Prof. Dr. Volker Turau

Institute of Telematics
Hamburg University of Technology

Acknowledgment

This is the place to thank all the people involved with your thesis / project. Examples would be your family, friends, and of course your supervisor. The acknowledgement will not have any influence on your grade; however, we think it is good style to have an acknowledgement in your thesis.

Abstract

The abstract of your thesis goes here. There may be formal requirements on it that can be found in the corresponding examination guidelines (Prüfungsordnung). If there are none, ask your supervisor. As a rule of thumb, the abstract should be concise and focused. It is not a shortened introduction to your work. We also suggest that—if an abstract is not required—only write one if it is really well done.

Table of Contents

List of Symbols	iii
1 Introduction	1
1.1 Motivation	1
1.2 Setup	2
1.3 Principle	2
2 PN and orthogonal sequences	3
2.1 Pseudo-random codes	3
2.1.1 Gold Codes	4
2.1.2 Kasami Codes	5
2.2 Comparison	5
3 Signal Processing	7
3.1 Base-Band	7
3.2 Transmission Band	7
3.3 Simulation	8
3.4 Peak detection	8
4 Implementation in Python	9
4.1 Documentation - Signal Processing	9
Bibliography	11
A Content of the DVD	13

Table of Contents

List of Symbols

$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	Regular sets of numbers
$\mathcal{NP}, \mathcal{P}$	Complexity classes
$\mathcal{V} = \{v_0, \dots, v_{N-1}\}$	Set of N nodes v_i belonging to a network with sink v_0
ϱ	Node density, i.e., the average number of nodes within another node's communication range
$(v_i, v_j) \in \mathcal{E}$	Set of bidirectional communication links in the network
$G = (\mathcal{V}, \mathcal{E})$	Graph representation of a wireless sensor network
$\mathcal{N}_i = \{v_j \in \mathcal{V} \mid i \neq j \wedge (v_i, v_j) \in \mathcal{E}\}$	The set of bidirectional communication partners of node v_i
$\mathcal{T} \subseteq \mathcal{E}, \quad \mathcal{T} = N - 1$	Routing tree rooted in the sink
\mathcal{T}_i	Subtree rooted in (and including) node v_i
$\mathcal{C}_i, \quad \mathcal{C}_i = \mathcal{C}_i $	The set and number of children of node v_i in \mathcal{T}
$\mathcal{F} = \{v_i \in \mathcal{V} \mid \mathcal{C}_i = \emptyset\}$	Set of leafs in \mathcal{T} and the number of leafs
\mathcal{F}_i	Set of leafs in the subtree \mathcal{T}_i of \mathcal{T}

List of Symbols

Introduction

Acoustic signals are nowadays mostly used for military and research propose. Electromagnetic waves cant perpetuate water, so by the use of acoustic modems water can be used as a medium. Underwater vehicles need to be capable of communicate to either themselves or objects placed at the surface. Even though the throughput of these aucustic signals is way less than its electromagnetic counterpart, it still has the advantage to perpetuate fluids.

1.1 Motivation

Using water as a medium for localization signals can be somewhat sophisticated to implement and are therefore a challenging research topic. Due strong damping of the classic electromagnetic waves general technologies like GPS or GLONASS are not applicable in under water scenarios.

Acoustic transmission comes in handy in this case. Most systems use a vehicle which transmits acoustic beacons. These are then received by hydrophones placed at the surface. By estimating the travel times the distance between hydrophones and the underwater vehicle can be calculated. However, to let it localize itself, the reverse method is needed. Thus, anchors firmly fixed at the surface send their individual signals and the vehicle receives them.

Signals send by the anchors need to be separable but still observable. Therefore we need codes that are orthogonal towards each other but nonetheless posses clear auto-correlation.

This project dives into the mathematical details of pseudo.random maximum length sequence generation and signal processing. By the use of cross-correlation and auto-correlation the separation of signals can be implemented and evaluated.

1 Introduction

1.2 Setup

The setup consists of 4 Anchors fixed at the surface broadcasting different signals produced from an python program. Acoustic Modems use broadcast the transfer band signal into the water. At the beginning these 4 anchors need to be synchronized. A blueROV2 receives these signals under water and saves them. Afterward the saved signals are used in python to calculate the position of the ROV.

1.3 Principle

To get the appropriate signals we need to apply certain signal processing steps. First the pseudo-random codes gets up-sampled and put through an cosine FIR filter to remove high frequencies from the base-band. The resulting signal is then shifted to the transfer band. Hereon either a custom delay is artificially added by zero padding for the simulation or is sent by the acoustic modem.

Because of the advantageous properties of the used codes the different codes can be separated by cross-correlation by the not delayed versions. The cleaner the auto-correlation the higher are the peaks which are used to measure the initial delay. Method of peak detection may be needed to filter out peaks caused from reflections.

PN and orthogonal sequences

There are two main goals need to be pursued for receiving higher localization accuracy.

First the code which is used for the under water localization needs its auto-correlation approaching a Dirac impulse. Thus one get advantageous detection capabilities.

The next factor are cross-correlation properties, which should meet certain criteria for improving the separation from other sequences. These will come in handy if noise, reflections and other artifacts emerge.

2.1 Pseudo-random codes

There are a couple of principles to generate PN sequences. Most of these methods use linear feedback shift registers to generate the codes by an initial condition. In this project I will concertize my research on gold codes, Kasami-Codes and the classical m-sequences which are also used for generating gold codes.

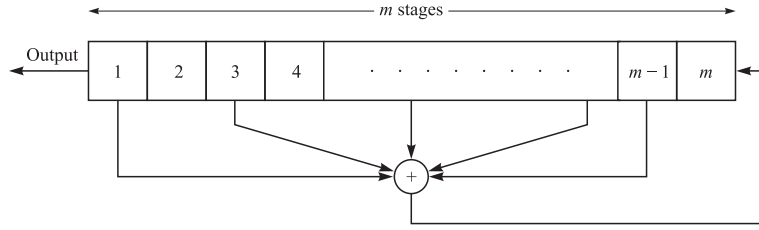
M-sequences are binary PN codes, which are generated by linear shift registers with feedback. The sequences are periodic and contain the same number of zeros and ones [PS08]. M-sequences need to fulfill certain criteria. First its length is defined by $N = 2^n - 1$ where n is the maximum degree of the generator polynomial $f(X)$ [SP80].

$$I. |u| = 2^n - 1 = N, \text{ from polynomial } h(x) \text{ of degree } n \quad (2.1)$$

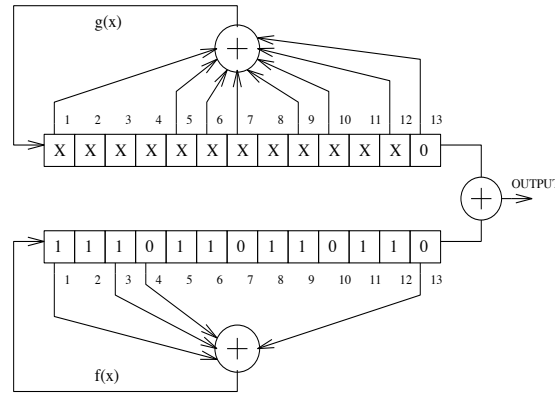
$$II. \frac{N}{gcd(N, q) = N'}, \text{ from decimation polynomials } \widetilde{h(x)} \quad (2.2)$$

Second the cross-correlation between m-sequences needs to take three values only, which are $-1, -t(n), t(n) - 2$. With it $t(n)$ is defined by $1 + 2^{\lfloor 0.5(n+2) \rfloor}$ [SP80]. If every pair of m-sequences is a preferred pair, they form a maximal connected set and these sets have a

2 PN and orthogonal sequences



■ **Figure 2.1:** Basic structure of an LFSR (Linear Feedback Register). [PS08]



■ **Figure 2.2:** LFSR structure of preferred generator polynomial of degree 13). [MD]

limited cardinality. Experiments from Gold and Koptizke showed that the number of such connected pairs is limited. Between degrees

[GK65]. To get an m-sequence we need a primitive which needs to fulfill certain criteria.

2.1.1 Gold Codes

Because of not optimal cross-correlation properties m-sequences alone are not applicable for the project. But if these type of codes are combined their correlation qualities can change. Gold Codes are m-sequences where two of them with same length are modulo-2 summed. [PS08]

The gold code which has the highest similarity to Gaussian random variable has a degree of 13. Thus, a generator polynomial pair of $x^{13} + x^4 + x^3 + x + 1$ and $x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^6 + x^5 + x + 1$ is chosen [MD] . (@TODO: remove this example)

$$Gold(u, v) = \{u, v, u \oplus v, u \oplus (v \ll 1), \dots, u \oplus (v \ll N - 1)\} \quad (2.3)$$

2.1.2 Kasami Codes

Kasami sequences are constructed in the same fashion by using m-sequences with the exception that a second sequence, which is used in the modulo sum, is formed by decimating the default m-sequence by $2^{m/2}$ [PS08] [SP80] [PPWW72].

$$w = u[2^{N/2} + 1] = \{u_1, \dots, u_i, \dots, u_N | \text{take every } i\text{-th bit of } u\} \quad (2.4)$$

$$\text{Kasami}(u) = \{u, u \oplus w, u \oplus (w \ll 1), \dots, u \oplus (w \ll 2^{N/2} - 2)\} \quad (2.5)$$

2.2 Comparison

For the localization process by orthogonal codes certain criteria needs to be met, which were named in the first chapter. To compare the before explained code types three measures are introduced.

The first one is the peak to side-lobe ratio (PSR) 2.6. This measure is defined by subtracting the mean from the peak of the auto-correlation. Then this value get divided by the standard deviation of the same auto-correlation. A higher PSR value signifies a smaller error between the auto correlation and the perfect Dirac resulting in better detection capability. The second one is the ratio between the auto-correlation peak and the maximum of the cross-correlation (ACR) 2.7. Here a higher value indicates good code separation qualities.

$$PSR = \frac{\max\{x_{ac}\} - \overline{x_{ac}}}{\sigma_{ac}} \quad (2.6)$$

$$ACR = \frac{\max\{x_{ac}\}}{\max\{x_{cc}\}} \quad (2.7)$$

From preferred polynomial all possible maximum length sequences, gold sequences and kasami sequences are generated. Then both measures are applied on the cross-correlation and auto-correlation functions of the random codes. The PSR and ACR measures are plotted against the used polynomials. Also the best case of PSR and ACR are plotted by their given correlation function.

Maximum length sequences hold the best auto-correlation properties in comparison to its competitors. But it shows peaks in its cross-correlation, making it a rather bad option for orthogonal separation. The Kasami sequence has a way better cross-correlation but still a small peak. The clear winner are gold codes because of the good auto-correlation and very good cross-correlation properties 2.3. Its auto-correlations lags a bit behind its competitors but orthogonality is as much as important.

2 PN and orthogonal sequences

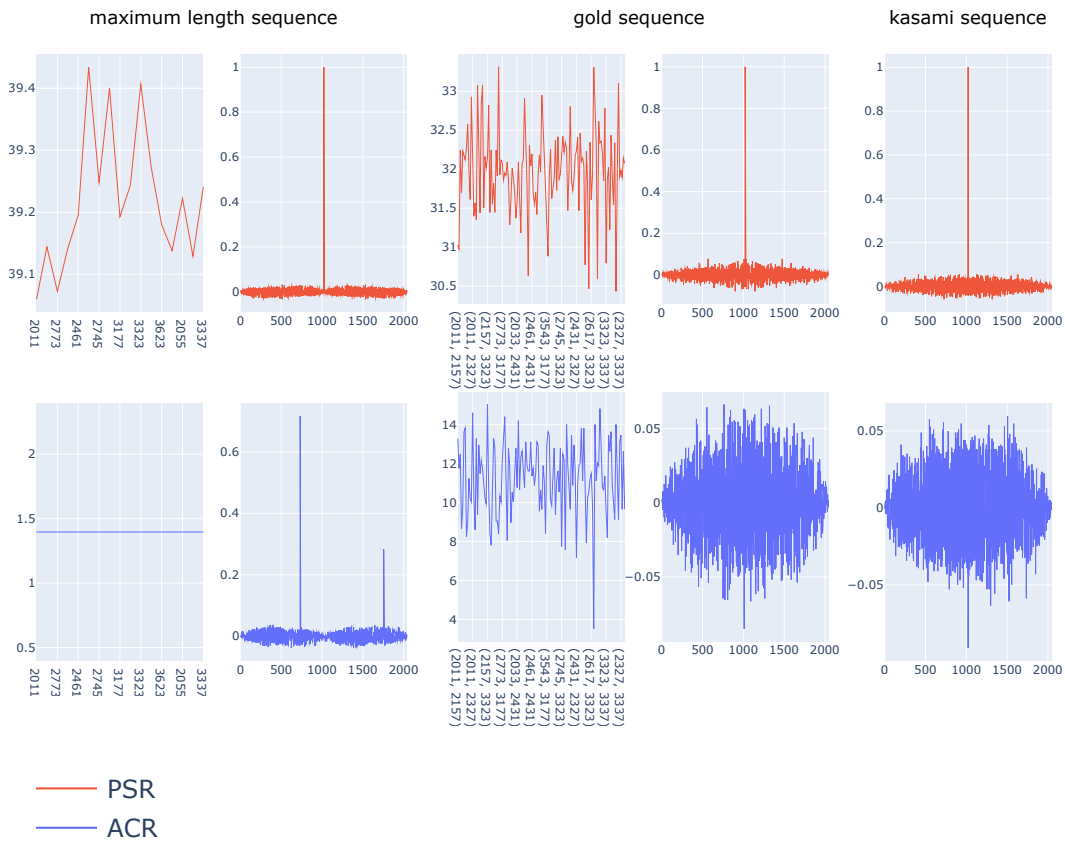


Figure 2.3: Random Sequence Evaluation

Signal Processing

The signal processing is separated into two sections. First the code needs to be transformed into the base-band. After that a spectrum shift to a specific transmission band is realized. Afterwards the exported signal is put through an simulator which adds reflection noise. The result is than again imported and gets reverse spectrum shifted. At the end a peak detection method is used to ignore reflection peaks.

3.1 Base-Band

Before a spectrum shift is applied to the signal, the bandwidth needs to be bounded. Otherwise absolute code bits would result in theoretically infinite frequencies which are impossible to implement for transmission. A raised cosine filter is therefore applied to remove all unwanted frequencies above a certain threshold. The base-band for our application is $20kHz$. Thus, our symbol length is set to $1/20kHz$. A appropriate roll-off coefficient of 0.125 is picked. The resolution of cosine needs to be high enough to include at least a couple of periods. A whole cosine is not tangible because its periodic and therefore infinite in time.

3.2 Transmission Band

Now the spectrum is ready to be shifted by the given transmission frequency f_c . The resulting signal could hold imaginary parts, hence we only move the real part further in processing.

$$x_{TB}(t) = Re\{x_{BB}(t) \cdot e^{-2\pi j f_c t}\} \quad (3.1)$$

3 Signal Processing

3.3 Simulation

The simulation consists of a watermark benchmark [vWOJ] and a additive GWN generated by a desired SNR between $-20dB$ and $20dB$ in steps of $5dB$. From the general equation of the Signal to Noise Ratio we derive our noise standard deviation by transforming this ratio.

$$SNR = \frac{P_{Signal}}{P_{Noise}} = \frac{\mathbb{E}(S^2)}{\mathbb{E}(N^2)} = \frac{\sum S^2}{N \cdot \sigma_N^2} \Leftrightarrow \sigma_N = \sqrt{\frac{N}{SNR \cdot \sum S^2}} \quad (3.2)$$

3.4 Peak detection

The received signal, consisting of summed delayed signals, cross-correlated by every anchor. If the signal was not reflected the peak in cross-correlation would be obvious. But because of the introduction by noise and water reflections more peaks could appear. To suppress these effects a CA-FAR Algorithm [Roh11] is applied to only detect the first reflected peak resulting in true results.

CA-FAR works by using multiple values intervals. The most outer one could be described as a train bin and is used to get an estimation of the signals noise. Especially CA-FAR uses averaging to estimate the noise by the measured cells. The bordering bin, defined as the guard cells, is used to reduce self-interference of the peaks. Thus, increasing window sizes N result in better noise estimating but is limited to the sample rate 3.3 [Roh11].

The calculated threshold is than scaled by a formula depending on the false alarm rate. The higher the false alarm rate, the more weak amplitude peaks gets included by the threshold 3.4.

Because Cell Averaging shows not satisfactory results in sensitive multi targets examples, the noise estimation could be enhanced by a sorting average. That principle is defined as CO-FAR [Roh11].

candidate saelstermple	i
guard interval (half)	\mathcal{G}
train interval (half)	\mathcal{T}
false alarm rate	η

$$Threshold(i)_x = \frac{\alpha}{2\mathcal{T}} \left[\sum_{j=i-(\mathcal{G}+\mathcal{T})}^{i+(\mathcal{G}+\mathcal{T})} x(j) - \sum_{j=i-\mathcal{G}}^{i+\mathcal{G}} x(j) \right] \quad (3.3)$$

$$\text{scale factor } \alpha = 2\mathcal{T} \left(\eta^{-1/2\mathcal{T}} - 1 \right) \quad (3.4)$$

Implementation in Python

For the Implementation python 3.10 is used. The signal generation and processing is based on the use of the libraries compy, scipy and numpy. Graphical plotting is realized by the use of the plotly library, which generates plots as a interactive html file.

4.1 Documentation - Signal Processing

4 Implementation in Python

Bibliography

- [GK65] R. Gold and E. Kopitzke. Study of correlation properties of binary sequences. *Interim Tech. Rep.*, 1–4(1), 1965.
- [MD] Steven J. Merrifield and John C. Devlin. The implementation of a multiplexing gold codegenerator using a xilinx logic cell array. *School of Electronic Engineering, La Trobe University*.
- [PPWW72] William Wesley Peterson, Wesley Peterson, Edward J Weldon, and Edward J Weldon. *Error-correcting codes*. MIT press, 1972.
- [PS08] John G. Proakis and Masoud Salehi. *Digital Communications*. McGraw-Hill Higher Education, New York, 2008.
- [Roh11] Hermann Rohling. In *2011 12th International Radar Symposium (IRS)*, pages 631–638, 2011.
- [SP80] Dilip V Sarwate and Michael B Pursley. Crosscorrelation properties of pseudorandom and related sequences. *Proceedings of the IEEE*, 68(5):593–619, 1980.
- [vWOJ] Paul van Walree, Roald Otnes, and Trond Jenserud. Watermark.

Bibliography

Content of the DVD

In this chapter, you should explain the content of your DVD.