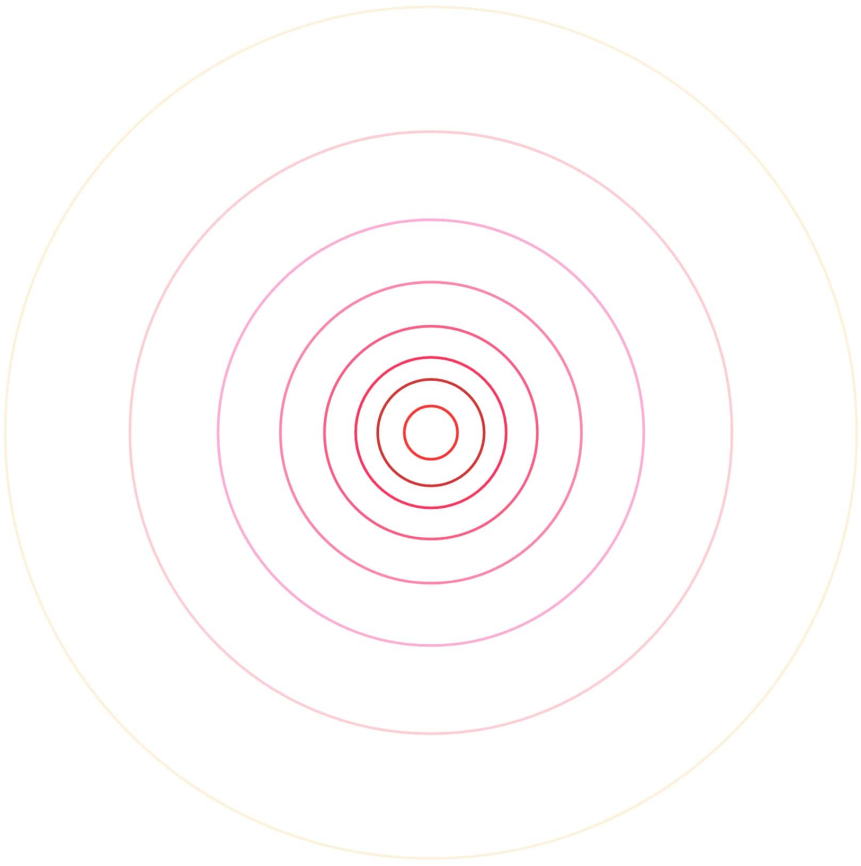
Total Least Squares

Comparing Least Squares Methods for Signal Reconstruction

Jort Houben

Bachelor Thesis
Applied Mathematics



TU Delft

Total Least Squares

Comparing Least Squares Methods for Signal Reconstruction

by

Jort Houben

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended publicly on the 12th of July, 2023 at lecture hall G in EWI.

Student number:	5132509	
Project duration:	March 7, 2023 – July 12, 2023	
Thesis committee:	Prof. dr. ir. M.B. van Gijzen,	TU Delft, supervisor
	Dr. H.N. Kekkonen,	TU Delft

TU Delft

Preface

Before you lies my bachelor's thesis "*Total Least Squares: Comparing Least Squares Methods for Signal Reconstruction*". My supervisor was Professor Martin van Gijzen from the Numerical Analysis Section. I would like to sincerely thank Professor Van Gijzen for not only supervising my thesis, but also for guiding me through the provided code and papers, the quick responses to e-mails, and the easter eggs we ate and jokes we made during our meetings.

I stumbled across the method of least squares–a method that will be used throughout this entire thesis– in a course on numerical linear algebra during my exchange at the TU Berlin. The ingenuity of the method fascinated me, and once I read that there was a bachelor end project that investigated and applied these kinds of methods, there was no doubt about my thesis choice.

I worked with much enthusiasm and interest on this project. I needed to extract much of the information and techniques for the signal reconstruction out of Professor Van Gijzen's MATLAB code. This could be challenging at times, but ultimately this was all part of the process.

Jort Houben
Delft, July 2023

Abstract

A common problem in wireless communication is the existence of multipath propagation. This means that a transmitted signal is received multiple times because of reflections caused by the environment. We present two ways of modeling multipath propagation of an acoustic underwater signal. We discretise these models to solve them numerically. During the solving process, we are presented with inconsistent, overdetermined systems of linear equations. We investigate two methods to go about these systems: the ordinary least squares method and the total least squares method. We reconstruct a signal using both of these methods and compare their results. The method of least squares reconstructs the signal moderately well. For the total least squares method this is not the case. It turns out that it is not straightforward to formulate a total least squares problem in the corresponding model. We suspect that, in part, this is why the signal reconstruction does not work well for the total least squares method.

Contents

Introduction

With the continual advancement of technology and the advent of the Information Age, digital communication has become an indispensable part of our lives. We can roughly categorise two kinds of digital communication: those which use wires, and those that do not.

Communication

Communicating without wires is significantly more challenging than with wires, simply because a signal transmitted through a wire doesn't encounter any obstacles on the way, and knows exactly where to go. When communicating without wires, a transmitter sends a signal in all possible directions in its environment. If this takes place in a closed environment–for example, underwater in the sea–there exist structures which can reflect the signal. These reflections could also reach the receiver. This causes multiple overlapping copies of the transmitted signal to be received. This phenomenon is known as multipath propagation. A depiction of underwater multipath propagation can be seen in Figure 1.

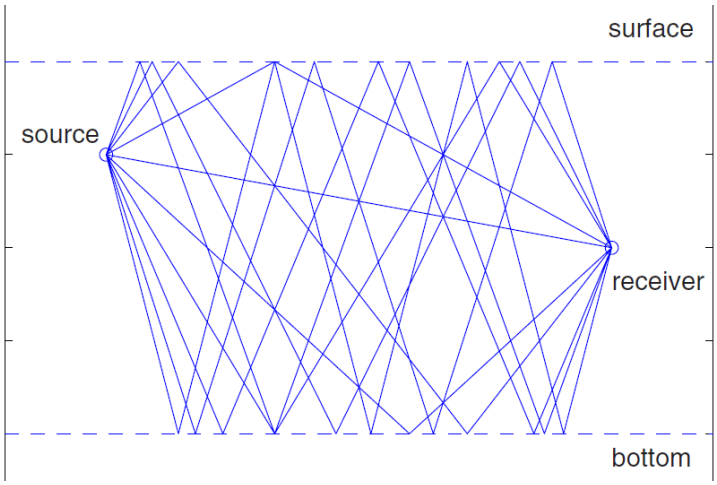


Figure 1: Example of underwater multipath propagation, taken from Zeng et al., 2010.

Since the reflected signals travel longer routes to get to the receiver, these signals are delayed and attenuated. So, even though only one signal is being sent, multiple transformed copies of this signal are being received. To reconstruct the transmitted signal, we need to somehow extract the original transmitted signal from all these transformed copies.

The received signal is usually modeled as the convolution of the transmitted signal and an impulse. In this thesis, we make a discretisation of this model to approximate the transmitted signal as closely as possible. During this process, we encounter inconsistent systems of linear equations. That is, systems without solutions. Over the centuries, many techniques have been developed to compute "solutions" to these systems with the least amount of error.

Historical Context

The first complete formulation of such a technique was developed by the German mathematician Carl Friedrich Gauss (1777-1855) at age 16, when he was studying the distribution of prime numbers. He later used the same technique to study the orbits of asteroids (Fraleigh and Beauregard, 1987, p. 375). However, Gauss didn't publish his findings for a long time–a trend that he would continue throughout

his whole life. The French mathematician Adrien-Marie Legendre (1752-1833) independently developed this method, and published it in 1805. We now call this technique the method of least squares. The famous dispute about who first discovered the method of least squares has gone down in history as the “priority dispute over the discovery of the method of least squares”. It it likely (but not certain) that Gauss developed the method much earlier than Legendre, but somehow failed to communicate it to the scientific community (Stigler, 1981).

In the ages to come, the least squares method has been studied and generalised extensively. One of its generalisations is the so-called method of total least squares. This method minimises a different kind of error, which is more reasonable in many real-world applications.

Research

We will study both methods when reconstructing our signal. We are particularly interested in the reconstruction of an acoustic signal sent underwater. Our research question is therefore:

> Does the method of total least squares give a better result than the method of least squares when reconstructing an acoustic underwater signal?

We investigate acoustic rather than electromagnetic signals because the latter attenuates very quickly under water, and hence is not well-suited for long-distance communication.

On top of that, we investigate a common application of least squares in regression analysis, namely in polynomial regression. We can also solve regression problems using total least squares, and we will investigate their differences.

Structure

This thesis begins with a treatise on the required prerequisite theory. This chapter mainly describes the least squares and total least squares method. It will also cover certain structured matrices and the convolution operation. All of the prerequisite theory is linear algebra.

We then apply these methods to polynomial regression. A polynomial can be estimated using the least squares as well as the total least squares method. We will consider both estimations and compare their results.

We conclude with the main objective of this thesis. We apply both methods to the reconstruction of a sea signal. We first describe a model of multipath propagation. We then perform a discretisation to obtain a numerical model which helps us approximate the transmitted signal. This approximation process involves “solving” an inconsistent system of linear equations. The way that this system is constructed suggests the usage of least squares to estimate certain parameters. We then slightly modify this model to obtain another model which suggests the usage of total least squares. For both models, we reconstruct the signal and compare their results.

We close this thesis with the conclusion and a small idea for further research, followed by a small appendix.

1 Preliminaries

In this chapter, we mainly discuss two methods to compute "solutions" with the least amount of error in unsolvable linear systems of equations. These are the methods of (ordinary) least squares and total least squares. This will turn out useful for the signal processing that we will discuss in Chapter 3. Both methods have a different approach for solving the problem, and therefore different results.

Before we explain these two methods, we will recall some linear algebra terminology that will be used repeatedly throughout this thesis. After that, the two methods and their derivation will be discussed, including a short comparison. This is followed by a description of two kinds of structured matrices, which will turn out to be useful in combination with the two methods. To finish things off, we will define the convolution and deconvolution operation, and see how this relates to one of the structures matrices.

1.1. Basic Linear Algebra Terminology

Let $Ax = b$ be a system of linear equations[1], where $A \in \mathbb{R}^{n\times m}$, $x \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$. We can distinguish three kinds of systems depending on the size of A. If A is square, so when $n = m$, then the system is called a *square system*. In this case, the number of equations is equal to the number of unknowns. If there are less equations than unknowns, so when $n < m$, the system is called *underdetermined*. If there are more equations than unknowns, so when $n > m$, the system is called *overdetermined*. The methods that we will study in this thesis will mostly apply to overdetermined systems of equations.

A system of linear equations has either one, zero, or infinitely many solutions. In the case that it has one or infinitely many solutions, the system is called *consistent*. If it has no solutions, it is called *inconsistent*.

A square matrix $A \in \mathbb{R}^{n\times n}$ is *positive semi-definite* if the number $x^\intercal Ax$ is non-negative for every $x \in \mathbb{R}^n$.

1.2. Ordinary Least Squares

When considering an overdetermined, inconsistent system $Ax = b$ for $A \in \mathbb{R}^{n\times m}$, $x \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$, a well-known way to approximate the "best" solution vector is using the (ordinary) least squares (LS) method:

$$x_{\text{ls}} := \underset{x\in\mathbb{R}^m}{\operatorname{argmin}} ||Ax - b||_2. \tag{1.1}$$

In other words, the LS method finds a linear combination of the columns of A such that the difference between this linear combination and b is minimal in the Euclidean norm.

The solution to the LS problem is derived as follows. First, we define the residual of a vector x in the system $Ax = b$ as $r(x) = b - Ax$. That is, the residual is the vector which tells us how "far away" x is from the solution. When we look at the LS problem now, we see that LS seeks to find an x for which

[1] This is often abbreviated to "system of equations" or simply "system".

the Euclidean norm of its residual is minimised.

The residual of a solution to the LS problem has an interesting property. A vector x_{ls} solves the LS problem if and only if its residual is orthogonal to the range of A–or equivalently, orthogonal to its column space. In other words, we have

$$A^{\intercal}(b - Ax_{\text{ls}}) = 0. \tag{1.2}$$

It follows that the solution to

$$A^{\intercal}Ax = A^{\intercal}b, \tag{1.3}$$

solves the LS problem (Fraleigh and Beauregard, 1987, pp. 360-362, 369–374).

The system in Equation (1.3) is known as the *normal equations* (in linear algebra, normality is synonymous with orthogonality). We have now derived the following result.

Theorem 1. *For a given matrix $A \in \mathbb{R}^{n\times m}$ and $b \in \mathbb{R}^n$, the following are equivalent.*

1. *The vector $x_{\text{ls}} \in \mathbb{R}^m$ solves the LS problem $\underset{x\in\mathbb{R}^m}{\operatorname{argmin}} ||Ax - b||_2$.*
2. *The vector $x_{\text{ls}} \in \mathbb{R}^m$ solves the normal equations $A^{\intercal}Ax = A^{\intercal}b$.*
3. *The residual $r(x_{\text{ls}})$ and $\operatorname{ran}(A)$ are orthogonal.*

It can be shown that $A^{\intercal}A$ is non-singular whenever A has full rank (Fraleigh and Beauregard, 1987, p. 361). Assuming A has full rank, the consistency of the normal equations ensures that $x_{\text{ls}} = (A^{\intercal}A)^{-1}A^{\intercal}b$ is the unique[2] solution to the LS problem (Golub and Van Loan, 1996, pp. 237-239).

We can use the singular value decomposition (SVD) to formulate the solution of a LS problem differently. For any $A \in \mathbb{R}^{n\times m}$, there exist orthogonal matrices $U = [u_1 \cdots u_n] \in \mathbb{R}^{n\times n}$ and $V = [v_1 \cdots v_m] \in \mathbb{R}^{m\times m}$ and a diagonal matrix $\Sigma_r = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$ such that for

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times m}, \tag{1.4}$$

we have

$$A = U\Sigma V^{\intercal}. \tag{1.5}$$

The values σ_i for $i = 1, \ldots, r$ are the so-called singular values of A, which are the square roots[3] of the eigenvalues of $A^{\intercal}A$ or $AA^{\intercal}$. Here we have $r = \operatorname{rank}(A)$, and we sorted the singular values in Σ_r in descending order. Under the restriction of this sorting, the SVD is unique. The vectors $u_1, \ldots u_n$ and $v_1, \ldots v_m$ are the normalised eigenvectors of $AA^{\intercal}$ and $A^{\intercal}A$ respectively, and they are called the left and right-singular vectors (respectively) (Golub and Van Loan, 1996, pp. 70-72).

In addition, we could split up U and $V^{\intercal}$ in the following block matrices:

$$A = [U_1 U_2] \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^{\intercal} \\ V_2^{\intercal} \end{bmatrix} = U_1 \Sigma_r V_1^{\intercal}, \tag{1.6}$$

where $U_1 := [u_1 \cdots u_r] \in \mathbb{R}^{n\times r}$ and $V_1 := [v_1 \cdots v_r] \in \mathbb{R}^{m\times r}$ only contain the first r left and right-singular vectors. This simplifies to

$$A = U_1 \Sigma_r V_1^{\intercal}. \tag{1.7}$$

Calculating the SVD this way is computationally less expensive. This formulation is known as the compact or condensed SVD. Moreover, we can write

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^{\intercal}, \tag{1.8}$$

[2] In general, if A is not full rank then x_{ls} is not unique. However, if we take the minimal vector in terms of the Euclidean norm out of all solutions, then this extra constraint causes the solution to be unique again (Golub and Van Loan, 1996, p. 257).

[3] Note that the square roots of these eigenvalues can be taken since the eigenvalues of $A^{\intercal}A$ are always non-negative real numbers. It can be shown that if a matrix is symmetric positive definite, its eigenvalues are non-negative real numbers. Now, $A^{\intercal}A$ is symmetric, and we have $x^{\intercal}A^{\intercal}Ax = ||Ax||_2^2 \geq 0$.

which is a useful expression for certain kinds of computations (Zhang, 2015).

We can use the compact SVD to give a different formulation to the solution of a LS problem. Previously, we saw that $x_{\text{ls}} = (A^\intercal A)^{-1}A^\intercal b$ is the solution to a LS problem if A has full rank.

Now, let $A = U\Sigma V^\intercal$ with $n \geq m$ have (full) rank $m = r$. The compact SVD yields

$$A = [U_1 U_2]\begin{bmatrix}\Sigma_r \\ 0\end{bmatrix}V^\intercal = U_1\Sigma_r V^\intercal. \tag{1.9}$$

It follows that $A^\intercal = V\Sigma_r U_1^\intercal$ and $A^\intercal A = V\Sigma_r^2 V^\intercal$. We then get $(A^\intercal A)^{-1} = V\Sigma_r^{-2}V^\intercal$. We can now write $x_{\text{ls}} = V\Sigma_r^{-1}U_1^\intercal b$.

Let us note one important aspect of the LS method by formulating it differently. Equivalently, we could formulate it as

$$x_{\text{ls}} := \underset{x\in\mathbb{R}^m}{\operatorname{argmin}}||\Delta b||_2, \quad \text{subject to} \quad Ax = b + \Delta b. \tag{1.10}$$

We now observe that in the LS method, only the vector b is corrected, while A remains unchanged (Markovsky and Van Huffel, 2007). We also note that $\Delta b = -r(x)$.

1.3. Total Least Squares

An alternative to the "ordinary" least squares method has been developed because of the method's asymmetry: the vector b is corrected whereas the matrix A is not. It could happen that the data in both A and b are perturbed, and in this case it is more reasonable to try to correct them both rather than just b.

Since we will need a measure of a matrix instead of a vector, we will need to define a norm on a matrix. A well-known generalisation of the Euclidean norm for vectors to a norm on matrices is the Frobenius norm, which is defined as

$$||A||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}|a_{ij}|^2}, \quad \text{for} \quad A \in \mathbb{R}^{n\times m}. \tag{1.11}$$

This generalises the Euclidean norm since for $m = 1$, we get

$$||A||_F = ||A||_2. \tag{1.12}$$

For an inconsistent, overdetermined system $Ax = b$, the so-called total least squares (TLS) method looks for the minimal corrections that need to be applied to both A and b to make the linear system $\hat{A}x = \hat{b}$ consistent, where $\hat{A} = A + \Delta A$ and $\hat{b} = b + \Delta b$. Specifically, the formulation of this method is stated as follows. For $A \in \mathbb{R}^{n\times m}$ with $n > m$ and $b \in \mathbb{R}^n$, we seek to find

$$x_{\text{tls}} := \underset{x\in\mathbb{R}^m}{\operatorname{argmin}}\,||[\Delta A \mid \Delta b]||_F, \quad \text{subject to} \quad (A + \Delta A)x = b + \Delta b. \tag{1.13}$$

That is, the TLS problem looks for the minimal augmented matrix $[\Delta A \mid \Delta b]$ such that $\hat{A}x = \hat{b}$ is a consistent linear system, and uses the Frobenius norm to measure minimality.

The solution to the TLS problem is derived as follows. First we rewrite the system as

$$[A \mid b]\begin{bmatrix}x \\ -1\end{bmatrix} = 0, \tag{1.14}$$

and let

$$[A \mid b] = U\Sigma V^\intercal = \sum_{k=1}^{r}\sigma_k u_k v_k^\intercal \tag{1.15}$$

be the SVD of $[A \mid b]$ with rank r. We note that if $r \leq m$, the rank-nullity theorem[4] implies $\text{nullity}([A \mid b]) \geq 1$, so that the system in Equation (1.14) is consistent. Hence we only consider systems where $r = m+1$. Also note that the inconsistency of $Ax = b$ implies that the rank of $[A \mid b]$ must always be one greater than the rank of A, since $b \notin \text{ran}(A)$.

Now, the goal of TLS is to make the system $[A \mid b][x \ -1]^\intercal = 0$ consistent by altering $[A \mid b]$ as little as possible. The only way to make this system consistent is to decrease the rank of $[A \mid b]$. That is, we need to find a so-called a low-rank approximation of this matrix.

The low-rank approximation problem for $C \in \mathbb{R}^{n \times m}$ with $n \geq m$ and rank r reads

$$\hat{C}_* := \underset{\hat{C} \in \mathbb{R}^{n \times m}}{\text{argmin}} ||C - \hat{C}||_F, \quad \text{subject to} \quad \text{rank}(\hat{C}) \leq k, \tag{1.16}$$

with $r \leq m$ and $k \leq r$, with k the maximum desired rank. The solution to this problem is a famous result in numerical linear algebra, and goes by the name of the Eckart–Young theorem. It states that the best solution to this low-rank approximation problem is given by

$$\hat{C} = U\Sigma_k V^\intercal. \tag{1.17}$$

That is, we simply equate the singular values $\sigma_{k+1}, \ldots, \sigma_r$ of C to zero in Σ, and the "remaining part" of the SVD yields the low-rank approximation. Moreover, this solution is unique if and only if $\sigma_k \neq \sigma_{k+1}$ (Eckart and Young, 1936).

Now, the low-rank approximation problem for TLS reads

$$[\hat{A} \mid \hat{b}]_* = \underset{[\hat{A} \mid \hat{b}] \in \mathbb{R}^{n \times (m+1)}}{\text{argmin}} ||[A \mid b] - [\hat{A} \mid \hat{b}]||_F, \quad \text{subject to} \quad \text{rank}\left([\hat{A} \mid \hat{b}]\right) \leq m. \tag{1.18}$$

If we express $[A \mid b]$ and $[\hat{A} \mid \hat{b}]$ as

$$[A \mid b] := \sum_{i=1}^{m+1} \sigma_i u_i v_i^\intercal \quad \text{and} \quad [\hat{A} \mid \hat{b}] = \sum_{i=1}^{m} \sigma_i u_i v_i^\intercal, \tag{1.19}$$

the minimiser is

$$[A \mid b] - [\hat{A} \mid \hat{b}] = \sigma_{m+1} u_{m+1} v_{m+1}^\intercal, \tag{1.20}$$

which is a rank-one matrix. The null space of $[\hat{A} \mid \hat{b}]_*$ now contains v_{m+1} as its only non-trivial element (ignoring its scalar multiples). Indeed, we have

$$[\hat{A} \mid \hat{b}]_* v_{m+1} = \sum_{i=1}^{m} \sigma_i u_i v_i^\intercal v_{m+1} = 0, \tag{1.21}$$

because of the orthogonality of V. The null space must be one-dimensional by the rank-nullity theorem. Scaling v_{m+1} such that its last entry equals -1, yields

$$x_{\text{tls}} = \frac{-1}{v_{m+1,n}} v_{m+1}, \tag{1.22}$$

where $v_{m+1,n}$ is the last entry of v_{m+1}. This is the unique solution of the system $[\hat{A} \mid \hat{b}]_*[x \ -1]^\intercal = 0$, and consequently the unique solution to the TLS problem.

In summary, for an inconsistent, overdetermined system $Ax = b$ whose augmented matrix $[A \mid b]$ has full rank, there exists a unique TLS solution if and only if $v_{m+1,n} \neq 0$. In this case, the solution is given by Equation (1.22) (Van Huffel and Vandewalle, 1991, pp. 33-35).

To bring the TLS matter to a close, we make the following observation about the TLS problem. Since ΔA and Δb are concatenated into the augmented matrix C to compute x_{tls}, this method implicitly assumes that the perturbations in A and b follow the same distribution (Markovsky and Van Huffel, 2007).

[4] The rank-nullity theorem is a famous result in linear algebra and states that for a matrix $X \in \mathbb{R}^{n \times m}$, we have $\text{rank}(X) + \text{nullity}(X) = m$, where the nullity is the dimension of the null space.

1.4. Difference Between Methods

The difference between LS and TLS has an intuitive interpretation when these methods are applied in linear regression. When trying to fit a line through the origin to a given set of points, LS minimises over the squared vertical distances, whereas TLS minimises over the squared shortest distances. See Figure 1.1 for an illustration.

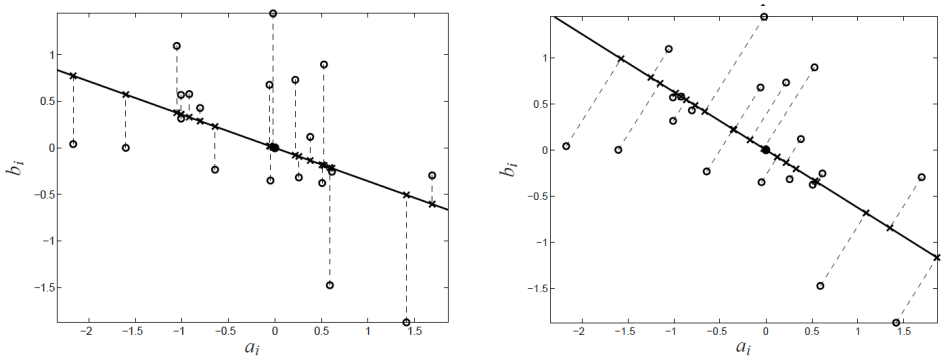


Figure 1.1: Comparison of a LS and TLS fit of a line for the same set of points. On the left, LS is used, and the vertical distances are illustrated by the dashed lines. On the right, TLS is used, and now the shortest distances are illustrated by the dashed lines. Naturally, the dashed lines here are orthogonal to the fitted line. Figure taken from Markovsky and Van Huffel, 2007.

Figure 1.1 illustrates the two observations that we made about LS and TLS earlier: LS only corrects in the y-direction (which corresponds to the vector Δb in Equation (1.1)), whereas TLS corrects in both the x- and y-direction[5] (which corresponds to ΔA and Δb respectively in Equation (3.16)). Depending on whether one or both coordinates are perturbed, we can choose either LS or TLS accordingly.

1.5. Structured Matrices

There are two kinds of matrices with a certain structure that we will use in this thesis.

A matrix is called Toeplitz (named after the German mathematician Otto Toeplitz (1881-1940)) if each of its diagonals is constant. That is, it has the form

$$A = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-(m-1)} \\ a_1 & a_0 & a_{-1} & & \vdots \\ a_2 & a_1 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \\ a_{n-1} & \cdots & & & a_0 \end{bmatrix} \in \mathbb{R}^{n \times m}. \tag{1.23}$$

We will see in Section 1.6 that Toeplitz matrices turn out to be fruitful when computing convolutions.

A Vandermonde matrix (named after the French mathematician Alexandre-Théophile Vandermonde (1735-1796)) is a matrix of the form

$$A = \begin{bmatrix} 1 & a_0 & a_0^2 & \cdots & a_0^{m-1} \\ 1 & a_1 & a_1^2 & \cdots & a_1^{m-1} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{m-1} \\ \vdots & & \vdots & \ddots & \\ 1 & a_{n-1} & & & a_{n-1}^{m-1} \end{bmatrix} \in \mathbb{R}^{n \times m}. \tag{1.24}$$

That is, given a vector $a = [a_0 \cdots a_{n-1}]^\intercal$, for each entry of a a Vandermonde matrix has a row consisting of the first m terms of the geometric series of this entry. We will use Vandermonde matrices for polynomial regression in Section 2.1.1. Note that if all entries of a are distinct, its Vandermonde matrix has full rank.

[5] Technically, TLS could also correct in one direction only, if this direction turns out to be optimal.

1.6. Deconvolutions

The convolution operation is ubiquitous in signal processing. We will define what a convolution and its inverse operation is. Furthermore, we will discuss how a convolution can be expressed as a matrix-vector product, and how this will help us to compute a deconvolution.

1.6.1. Convolution

The (discrete) convolution of two infinite sequences $(\dots, a_{-1}, a_0, a_1, \dots)$ and $(\dots, x_{-1}, x_0, x_1, \dots)$ is the sequence $(\dots, b_{-1}, b_0, b_1, \dots)$, whose kth entry is defined as

$$b_k = \sum_{i=-\infty}^{\infty} a_{k-i} x_i. \tag{1.25}$$

Likewise, we can define the discrete convolution of two vectors $a \in \mathbb{R}^n$ and $x \in \mathbb{R}^m$ as a vector $b \in \mathbb{R}^{n+m-1}$, whose kth entry is given by

$$b_k = \sum_{i=0}^{n+m-1} a_{k-i} x_i. \tag{1.26}$$

In the case that an entry of a has an "illegal" index–that is, the index is non-positive or larger than the length of the vector–we take this entry to be zero. Note that this convolution is commutative.

1.6.2. Toeplitz Matrices

We can describe the convolution of two vectors as a matrix-vector product using a Toeplitz matrix, whose structure we discussed in Section 1.5. Given two vectors $a \in \mathbb{R}^n, x \in \mathbb{R}^m$, if we define

$$A := \begin{bmatrix} a_0 & 0 & \cdots & 0 \\ a_1 & a_0 & \cdots & 0 \\ \vdots & a_1 & \ddots & \\ a_{n-1} & \vdots & \ddots & a_0 \\ 0 & a_{n-1} & & a_1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{n-1} \end{bmatrix} \in \mathbb{R}^{(n+m-1)\times m}, \tag{1.27}$$

we have the following equality:

$$a \otimes x = Ax, \tag{1.28}$$

where $\otimes$ denotes convolution. Thus, Toeplitz matrices can be used to translate a discrete convolution of two vectors into a matrix-vector product (Wintermantel and Luder, 1998).

1.6.3. Deconvolution

Instead of computing the vector b as a convolution of a and x, we might be interested in the inverse operation. That is, given $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^{n+m-1}$, we want to compute $x \in \mathbb{R}^m$ such that $a \otimes x = b$. This is known as a deconvolution. Observe that using the theory described in Section 1.6.2, computing the deconvolution of $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^{n+m-1}$ is equivalent to solving the system

$$Ax = b, \tag{1.29}$$

where A has the structure described in Equation (1.27).

Note that since we have $n + m - 1 > m$, the system $Ax = b$ is always overdetermined. Furthermore, observe that any column of A cannot be a linear combination of the other columns, and so this form of a Toeplitz matrix always has full rank. Moreover, this system is likely to be inconsistent when we work with real-world data. This suggests that we can solve the system using LS or TLS.

In conclusion, we can translate a deconvolution problem to a linear system of equations, and then use either LS or TLS to compute or estimate the deconvolutions.

2

Regression Analysis

A common application of LS can be found in regression analysis. In particular, fitting a polynomials to a set of points can be formulated as an optimisation problem which we can solve using LS. We could also give this problem a TLS formulation and solve it using this method.

We will start this chapter by formulating polynomial regression as an overdetermined system of equations. We will then study the approximation that LS and TLS provide for a given set of data points. Subsequently, we will note an important observation about the way that TLS goes about approximating its polynomial. This is followed by a study of the results of LS and TLS when a slight adjustment in the computation of approximations is made.

2.1. Polynomial Regression

Both the LS and TLS method can be used for polynomial regression. For either technique, a matrix A and vectors x and b are constructed in such a way that the problem of polynomial fitting is translated to a system of equations, which in turn can be solved using LS or TLS.

2.1.1. Ordinary Least Squares

Suppose we have n (real) data points (α_i, β_i) for $i = 0, \ldots, n-1$ and we want to find a real polynomial $p(t) = c_0 + c_1 t + \ldots + c_{m-1} t^{m-1}$ of degree $m-1$ which fits the data points best in the "least squares sense". We assume $n \geq m+1$. That is, we are looking for a $p \in \mathcal{P}_{m-1}$[1] such that

$$\sum_{i=0}^{n-1} (p(\alpha_i) - \beta_i)^2 \leq \sum_{i=0}^{n-1} (q(\alpha_i) - \beta_i)^2, \quad \text{for all } q \in \mathcal{P}_{m-1}. \tag{2.1}$$

So, we say that $p(t)$ fits the data points best if the sum of the squared distances between the polynomial and the data points is minimal.

We can compute the coefficients of p using the LS method. If we define

$$b := \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_{n-1} \end{bmatrix} \in \mathbb{R}^n, \quad A := \begin{bmatrix} 1 & \alpha_0 & \cdots & \alpha_0^{m-1} \\ 1 & \alpha_1 & \cdots & \alpha_1^{m-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha_{n-1} & \cdots & \alpha_{n-1}^{m-1} \end{bmatrix} \in \mathbb{R}^{n \times m}, \quad \text{and} \quad x := \begin{bmatrix} c_0 \\ \vdots \\ c_{m-1} \end{bmatrix} \in \mathbb{R}^m, \tag{2.2}$$

we can associate the coefficients of q with the vector x, so that we get

$$\sum_{i=0}^{n-1} (q(\alpha_i) - \beta_i)^2 = ||Ax - b||_2^2. \tag{2.3}$$

[1] Here $\mathcal{P}_k$ denotes the set of all real polynomials of degree at most k.

If we minimise the RHS of Equation (2.3), we have formulated polynomial regression as a LS problem (Boyd and Vandenberghe, 2018, pp. 255-258; Bingham and Fry, 2010, pp. 66-68).

Hence, applying the LS method to the overdetermined[2] system $Ax = b$ is equivalent to finding a polynomial that fits the given data best. For uniqueness of p, we note the following. The matrix A is a Vandermonde matrix, whose structure and properties we discussed in Section 1.5. Recall that if $\alpha_0, \dots, \alpha_{n-1}$ are distinct, then A has full rank, and consequently its LS formulation has a unique solution. So, if all x-coordinates of the data points are distinct, there is a unique polynomial which fits this data best in the least squares sense.

In the case of an underdetermined or square system, there is no need to apply LS since the system that we need to solve becomes consistent. In particular, for a square system there exists a unique polynomial that lies on all data points, and for an underdetermined system there is an infinite amount of polynomials that lie on the data points. A proof of these results is given in Appendix A.1.

2.1.2. Total Least Squares

For the TLS method, we use the same A, x and b, which we defined in Equation (2.2). As explained in Section 1.3, we solve

$$x_{\text{tls}} := \underset{x \in \mathbb{R}^m}{\operatorname{argmin}} ||[\Delta A \mid \Delta b]||_F \quad \text{subject to} \quad (A + \Delta A)x = b + \Delta b, \tag{2.4}$$

which yields the coefficients of p in the TLS sense.

2.1.3. Results

We test both methods on the same data. In Figure 2.1, a comparison between the LS and TLS method is given. The data points are constructed in the following way. Ten x-coordinates are randomly chosen on the interval. Then p is evaluated at these x-coordinates, so that we have obtained 10 random points on p on the interval. These points are then perturbed by adding Gaussian noise to both the x- and y-coordinate separately with parameters $\mu = 0$ and $\sigma^2 = 0.2$. These perturbed points are then taken to be the data points.

Since the data in both the x- and y-direction is perturbed, we might expect that TLS yields a better fit. However, visually it is not entirely clear which method gives a better result. To give a quantitative measure on this matter, we have listed two kinds of errors and the two minimised norms in Table 2.1.

The residual error is defined to be $||Ax-b||_2$ and the Frobenius error $||[\Delta A \mid \Delta b]||_F$, for which $(A+\Delta A)x = b + \Delta b$ holds, which we compute for both $x = x_{\text{ls}}$ and $x = x_{\text{tls}}$. In other words, these errors are the quantities minimised by LS and TLS. On top of that, we have computed the orthogonal error, which is the square root of the sum of the squared shortest distances from the data points to the polynomials.

	LS	TLS
Residual Error	2.18	2.87
Frobenius Error	2.18	0.13
Orthogonal Error	0.46	0.51

Table 2.1: Comparing the residual, Frobenius, and orthogonal error.

It is unsurprising that the residual error is smaller for LS than for TLS, since this error is minimised in LS. Likewise, it is unsurprising that the Frobenius norm is smaller than the residual norm for LS, since this norm is minimised in TLS. Moreover, note that since $||[0 \mid \Delta b]||_F = ||\Delta b||_2$, the residual error for LS always equals its Frobenius error.

[2] Note that the system is overdetermined because of our assumption $n \geq m + 1$.

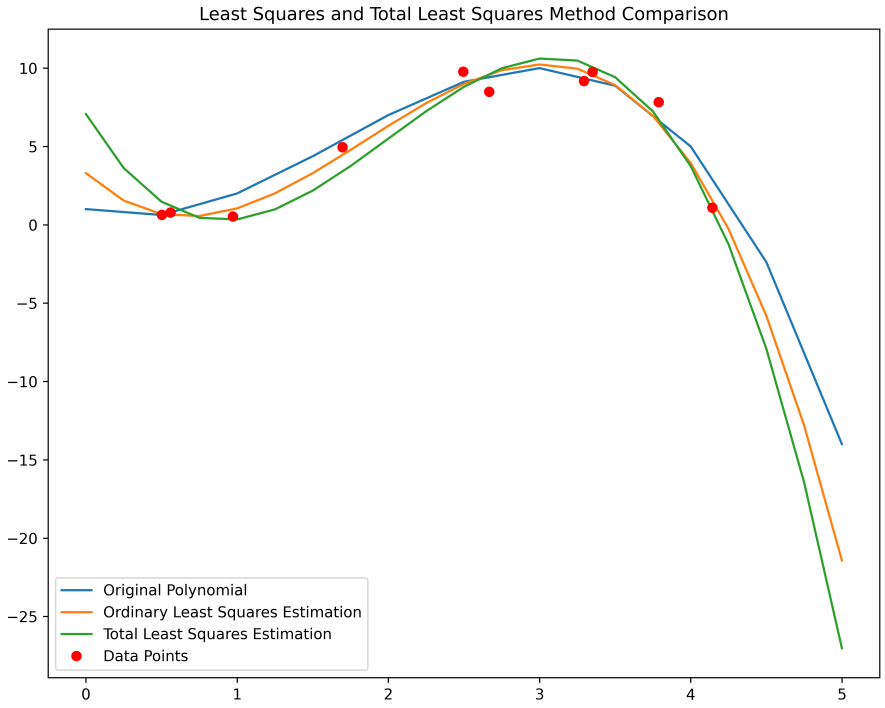


Figure 2.1: Comparing polynomial regression results using LS and TLS to fit 10 data points. The original polynomial is $p(x) = 1 - 3x + 5x^2 - x^4$, plotted on the interval $[0, 5]$.

What can be surprising is that the orthogonal error is greater for TLS than for LS. This is because the TLS formulation of polynomial regression in Equation (2.4) does not necessarily minimise the orthogonal error. This formulation only minimises the orthogonal error if the polynomial is a linear function and it passes through the origin.

2.2. Difference Between Methods

At the end of Section 1.3, we alluded to an implicit assumption that TLS makes. It assumes that the perturbation in every column of the matrix $[\Delta A \mid \Delta b]$ has the same distribution.

By studying the structure of A, we can see that this assumption does not hold. Recall that A was defined as follows:

$$A = \begin{bmatrix} 1 & \alpha_0 & \cdots & \alpha_0^{m-1} \\ 1 & \alpha_1 & \cdots & \alpha_1^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_{n-1} & \cdots & \alpha_{n-1}^{m-1} \end{bmatrix} \in \mathbb{R}^{n \times m}, \tag{2.5}$$

where α_i are the x-coordinates of the data points for $i = 0, \ldots, n-1$.

The first column of A is completely independent from the data points, the second column is composed of the x-coordinates of the data points, and the remaining columns are powers of these x-coordinates. The data points (α_i, β_i) are constructed by adding Gaussian noise with parameters $\mu = 0$ and $\sigma^2 = 0.2$ to both α_i and β_i. When powers are taken of α_i, this is generally not the same as taking the same power of an original (unperturbed) x-coordinate and then adding Gaussian noise. This causes only the second and last column in $[\Delta A \mid \Delta b]$ to be perturbed the same way, whereas all other columns are perturbed differently.

Thus, none of the entries of A except the ones from the second column are generated by adding Gaussian noise to the original data. This makes the perturbation of these entries different from the ones in b, even though this is assumed by the TLS method.

2.3. Different Result

In order to investigate whether TLS gives a different result for polynomial regression when all entries of $[\Delta A \mid \Delta b]$ are perturbed the same way, we conduct the following experiment. Instead of taking random points on the original polynomial, perturbing these, and then formulating A and b, we can construct A and b first using the unperturbed points, and then add Gaussian noise to every entry of A and b to cause them to be perturbed in the same way.

Reversing the order of perturbing the points on the polynomial and constructing A and b yields the result in Figure 2.2.

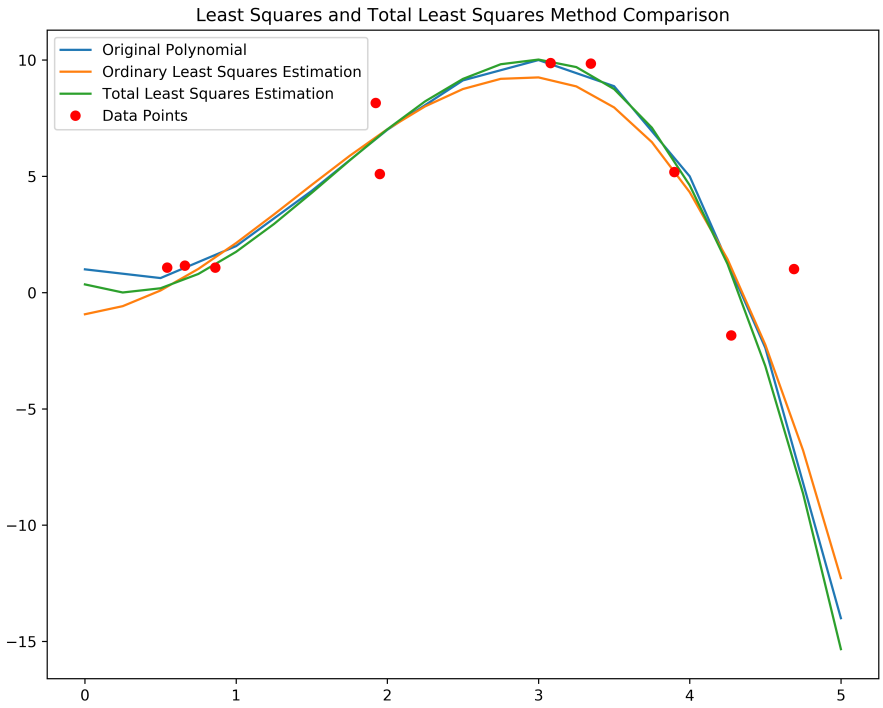


Figure 2.2: Comparing the two polynomials computed with the LS and TLS method. The original polynomial is again $p(x) = 1 - 3x + 5x^2 - 1x^3$ on the interval $[0, 5]$.

The error measures and norms are given in Table 2.2. We see again that the Frobenius norm of the correction matrix is smaller for TLS then for LS. This time, the orthogonal error is smaller for TLS than for LS.

	LS	TLS
Residual Error	2.25	3.32
Frobenius Error	2.25	0.57
Orthogonal Error	1.17	0.73

Table 2.2: Comparing the residual, Frobenius, and orthogonal error for the reversed order of perturbation.

In the first example, we saw that the orthogonal error was not minimised by TLS. This has to do with how the Vandermonde matrix is constructed. For any row i in A, we have the following equality:

$$\begin{bmatrix}1 \ \alpha_i \ \cdots \ \alpha_i^{m-1}\end{bmatrix}\begin{bmatrix}c_0 \\ \vdots \\ c_{m-1}\end{bmatrix} = \beta_i. \tag{2.6}$$

What TLS then does to this row is

$$\begin{bmatrix}1+\Delta a_0 & \alpha_i+\Delta a_1 & \cdots & \alpha_i^{m-1}+\Delta a_{m-1}\end{bmatrix}\begin{bmatrix}c_0 \\ \vdots \\ c_{m-1}\end{bmatrix} = \beta_i + \Delta b, \tag{2.7}$$

and then minimises $\Delta a_0, \ldots, \Delta a_{m-1}, \Delta\beta$ using the Frobenius norm. Computing this inner product yields

$$(1+\Delta a_0)\,c_0 + (\alpha_i+\Delta a_1)\,c_1 + \cdots + \left(\alpha_i^{m-1}+\Delta a_{m-1}\right)c_{m-1} = \beta_i + \Delta b. \tag{2.8}$$

Rewriting gives

$$\sum_{k=0}^{m-1} c_k \alpha_i^k + \left(\sum_{k=0}^{m-1} c_k \Delta a_k - \Delta b\right) = \beta_i. \tag{2.9}$$

We see that TLS really just adds a constant to the polynomial. If the squared shortest distances would be minimised, TLS would be able to make the polynomial “move” over the shortest line connecting the polynomial and the corresponding point. So, TLS would shift in both the x and y-direction to make it fit the point. For a polynomial $p(x)$, shifting Δx in the x-direction and Δy in the y-direction would result in $p(x-\Delta x)+\Delta y$. However, since merely a constant is added to the polynomial in Equation (2.9), TLS does not necessarily minimise the squared orthogonal distances.

3

Sea Signal Reconstruction

A common problem in wireless communication is the phenomenon of so-called multipath propagation. That is, when a transmitter sends a signal to a receiver, the signal radiates from the transmitter in every direction. Given that certain structures in the environment are present that can reflect the signal, this causes the receiver to receive echos of the signal through these reflections. Furthermore, since the reflected signals travel longer distances to the receiver, these signals will delay and lessen in amplitude. The phenomenon of sending one signal but receiving multiple signals through different routes is known as multipath propagation. An illustration is given in Figure 3.1.

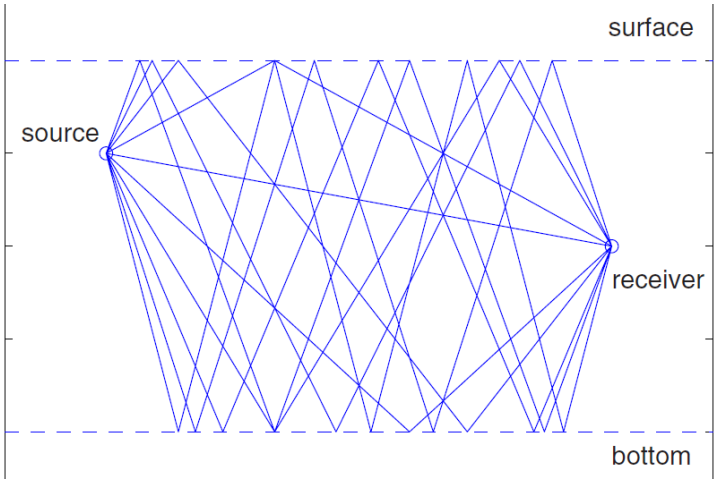


Figure 3.1: Example of multipath propagation for an acoustic signal under water for a given point in time, taken from Zeng et al., 2010.

In short, when a signal is transmitted, not just the original signal is received, but an indefinite amount of attenuated delayed signals as well. Our objective is to reconstruct the unknown transmitted signal using the received signal.

Multipath propagation is caused by the signal being reflected by certain structures within the environment. These structures could be a sea surface, seabed or a wall. In this thesis, we are particularly interested in transmitting acoustic signals under water, and thus we are faced with its environmental obstacles.

The data from the sea signals that we are using in this thesis were gathered by a team of researchers–including Prof. Van Gijzen–when they were conducting communication experiments in the North Sea in 1999. More information on these experiments will be given in Section 3.3.1.

In this chapter, we construct a model for the multipath propagation of an underwater acoustic signal. This model will help us to retrieve the transmitted signal from the received signal using different techniques. There are two key parameters to be estimated in this model which allows one to fully reconstruct the transmitted signal. We study two methods for this reconstruction.

The first method is treated in Section 3.3.2 and is the most common technique. This method uses LS and a deconvolution to reconstruct the transmitted signal. The second method is treated in Section 3.3.4 and can use both LS or TLS, on top of a convolution to reconstruct the signal. However, this method is based on an assumption which does not have much support, but since we can apply TLS to it, we study it nonetheless.

3.1. Multipath Propagation

When sending acoustic communication signals under water, we are faced with the consequent multipath propagation of its environment. Figure 3.1 gives an example of what such a propagation might look like.

We see that the transmitted signal is received directly through the straight line that connects the source and the receiver. On top of that, numerous copies of the same signal are received through reflections from the sea surface and bottom. These signal travel a longer distance, causing them to be attenuated and delayed. Hence, on top of the directly received signal, there is an indefinite amount of attenuated, delayed signals being received, whereas only one (unattenuated and undelayed) signal is sent.

3.2. Model

Underwater acoustic communication has been studied extensively. We first present the usual model for multipath propagation. We then give a simplified overview of the structure of the transmitted signal. This structure is designed according to computational needs that will become clear later on. Since we work with discrete signals (in bits), we end this section with a discretisation of the continuous model.

3.2.1. Continuous Model

Multipath propagation is commonly modeled the following way. For the received signal $r(t)$ and transmitted signal $s(t)$, we have the following relation:

$$r(t) = s(t) \otimes h(t) + v(t), \tag{3.1}$$

where $h(t)$ is the impulse response, $v(t)$ is Gaussian noise, and $\otimes$ denotes convolution. The impulse response gives a brief input, and helps us study the effects its effects on the output. Since we look for a minimal error in the LS sense, we define $h(t)$ as follows:

$$h(t) := \min_{f(t)} \int \left|\left| r(t) - s(t) \otimes f(t) \right|\right|^2 \mathrm{d}t. \tag{3.2}$$

We model $h(t)$ for m multipaths as

$$h(t) = \sum_{k=1}^{m} a_k \delta(t - \tau_k), \tag{3.3}$$

where a_k and τ_k are the attenuation factors and time delays respectively for $k = 1, ..., m$, and $\delta(\cdot)$ is the delta function. So, $h(t)$ gives an impulse at certain times (and equals zero otherwise), whose attenuation depends on that particular point in time. Intuitively, we would expect that $a_k \leq 1$ for all k. We will see later that this intuition often corresponds to how well the signal is reconstructed.

Equating the sum of attenuated multipaths to the received signal gives

$$r(t) = \sum_{k=1}^{m} a_k s(t - \tau_k) + v(t), \tag{3.4}$$

so that we see that the received signal is indeed a sum of attenuated, delayed transmitted signals. The exact derivation can be found in Appendix A.2. The amount of multipaths m is usually determined by trial and error.

Thus, in order to reconstruct the transmitted signal from the received one, we ought to estimate the two sequences of parameters a_k and τ_k for $k = 1, \ldots, m$ (Zeng et al., 2010).

3.2.2. Structure of the Signal

The information (in bits) of the signal are transmitted on a so-called carrier wave. In our case, this is an acoustic wave that allows the signal to move through the water. The process of encoding the information on the carrier wave is called modulation. The inverse process of attaining the information from the carrier wave is called demodulation.

Every transmitted signal consists of two parts: the learning signal and the communication signal. The latter part holds the actual information of the signal. The former is a standard signal, whose received signal is known (Van Gijzen and Van Walree, 2000).

The standard transmitted signal is a part of a so-called maximum length sequence (MLS). This is a binary sequence of period $2^n - 1$, for some $n \in \mathbb{N}$, and contains roughly as many 1s as 0s. These and other properties are desirable when estimating attenuation factors (Birdsall et al., 1971).

The learning signal is equal to the standard signal after modulation. When the learning signal has been fully sent, there is a period of silence so that the receiver can distinguish between the learning and communication signal. An illustration of the whole structure of a transmitted signal is given in Figure 3.2.

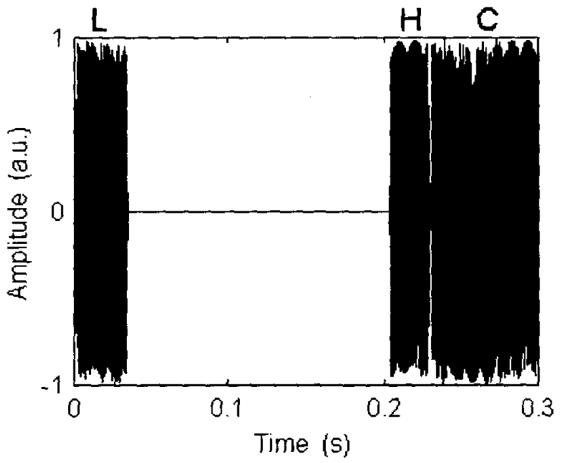


Figure 3.2: The structure of a transmitted signal. In particular, the amplitude of the signal as a function of time is given. Taken from Van Gijzen and Van Walree, 2000.

Note the learning signal (L) at the start, the period of silence depicted by zero amplitude, and the communication signal (composed of H and C)[1].

In Section 3.2.3, we will explain how the structure of the transmitted can be exploited to estimate the attenuation factors and time delays.

[1]The header of the communication signal contains certain kinds of information, for example about the modulation type. Since this part of we communication signal is not relevant for this thesis, we will only consider the (C) part of the communication signal.

3.2.3. Discrete Model

Whereas the model in Equation (3.1) assumes a continuous signal, all digital signals are discrete, and hence the information is encoded in bits[2]. Therefore we discretise the model to compute the attenuation factors and time delays. This discretisation reads

$$r = s \otimes h + v, \tag{3.5}$$

where $\otimes$ now denotes a discrete convolution[3]. We note that since this discretisation is taken over a finite-time period, the received signal is taken from a certain sample period. In Section 1.6 we saw that a discrete convolution of two vectors is equal to a particular kind of matrix-vector product. This yields the new discretisation

$$r = Sh + v, \tag{3.6}$$

where S is a Toeplitz matrix constructed using the transmitted signal, h is the (unknown) vector of attenuation factors, r the vector of the received signal, and v the Gaussian noise.

Similar to how we arrived at Equation (3.2), we assume v to be minimal, so that we need to solve the system

$$Sh = r. \tag{3.7}$$

To determine h, both S and r need be known. Clearly, r is known. As we discussed in Section 3.2.2, the signal is structured in such a way that we know s during the reception of the learning signal. Since the matrix S is fully determined by the known vector s, we can now exploit the structure of the signal to solve the system.

The matrix S is structured the following way:

$$S = \begin{bmatrix} s & 0 & \cdots & 0 \\ -1 & s & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ -1 & -1 & \cdots & s \end{bmatrix}. \tag{3.8}$$

Keep in mind that s is a column vector (so not a number) composed of the bits which encode the signal. In the first column of the matrix S, the vector s is placed at the top, and the rest of the column below s is padded with -1s. For every other column, the entries of the vector s are all put one entry lower than in the previous column, the entries below s are padded with -1s again, and the entries above s are zero-padded.

The columns of the matrix S are zero-padded above the vector s because before the signal is sent, there is no signal, and therefore also no carrier wave being transmitted. The lack of the carrier wave is encoded with 0s. After the signal has been sent, the previously mentioned period of silence takes place. However, since the whole signal (so, the learning signal, the communication signal, and the period of silence) is transmitted through one carrier wave, this period is encoded by a string of -1s. Hence the entries below the vector s are padded with -1s.

So, solving the (inconsistent) system in Equation (3.7) yields the attenuation factors. Note that the structure of S causes it to have full rank, as we discussed in Section 1.6. Consequently, the corresponding LS problem has a unique solution. That is, there is only one choice of the “best” attenuation factor for each considered multipath.

As we discussed in Section 3.2.1, we also need to estimate the time delays in order to reconstruct the signal. The signals are send through shallow water, which causes there to be so many multipaths that we take a sample. Clearly, the sample size is the amount of multipaths, and we assume these to be

[2] To be precise, the signal used in this thesis is not composed of 1s and 0s, but 1s and -1s instead. The replacement of 0 with -1 has to do with how the signal is transmitted using a carrier wave through the water. The exact technical details of this procedure are outside of the scope of this thesis. We refer to Van Gijzen and Van Walree, 2000 for more information on this topic.

[3] Note that the same symbol is used for continuous as well as discrete convolutions. The context should make clear which one is used.

uniformly distributed over the sampling period[4].

One might observe that this way of structuring S causes a problem. Indeed, because of the -1 padding instead of zero-padding below the vectors s, technically we have $Sh \neq s \otimes h$ since S is not equal to the Toeplitz matrix we defined in Equation (1.27). However, the -1s only start to appear after n entries, where n is the length of s (so, n is the amount of bits in the transmitted signal). After these amount of bits, the received signal is not important anymore.

Since S maps h to r, this matrix needs to have m columns (recall that m is the amount of multipaths). For the amount of rows, observe that the vector s is shifted $m-1$ times, so that there are $n+m-1$ rows, where n is the amount of bits in s. So, in short, S is a $(n+m-1)$-by-m matrix composed of 1s, 0s, and -1s.

The learning signal is lengthened to have length $n+m-1$, so that the computation can be performed[5].

3.3. Signal Reconstruction

In this section, we will apply the discrete model for signal reconstruction. We will use real-world data. We start this section by explaining where this data comes from and how it is gathered.

There are two ways we can go about reconstructing the signal. The first uses LS and a deconvolution. The second can use both LS and TLS and then uses a convolution to reconstruct the transmitted signal. The two methods and their results will be discussed in this order. To keep things simple, we will only perform the reconstruction of the learning signal.

3.3.1. Sea Experiments

The data used for performing the reconstruction of the transmitted signal was provided by supervisor Prof. M. Van Gijzen. In April and May 1999, he conducted research at TNO with seven other scientists on underwater acoustic signals in the North Sea, approximately 10 km off the coast near the Dutch village Noordwijk.

For two weeks, they conducted experiments on acoustic underwater communication. The frequency band of the signals was 1-15 kHz, the signalling distance was tested for a range of 1-10 km, and the signal transmitter was placed at depths of 18-22 m (Van Gijzen et al., 2000, pp. 555-560). A depiction of the set-up used for these experiments is given in Figure 3.3.

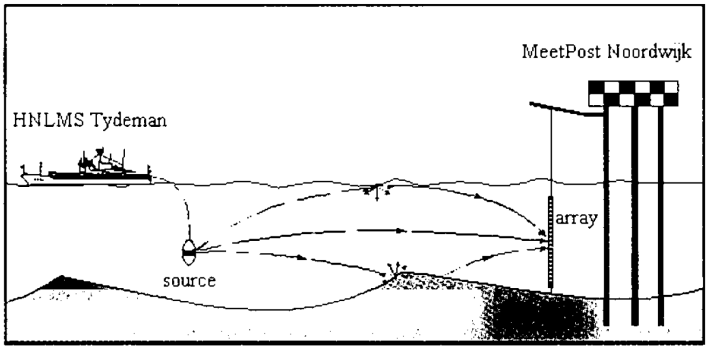


Figure 3.3: Set-up for the underwater acoustic signal experiments. Taken from Van Gijzen et al., 2000, p. 557.

The MLS and one audio file containing a received acoustic signal from these experiments is used in this thesis.

[4]For techniques to estimate the time delays, we refer the reader to Zeng et al., 2010.
[5]The exact lengthening procedure is outside the scope of this thesis.

3.3.2. Deconvolution

The way S is constructed gives rise to a Toeplitz matrix. Moreover, since the system in Equation (3.7) is overdetermined and is very likely to be inconsistent, solving for h is equivalent to solving a deconvolution problem discussed in Section 1.6. Thus, estimating the attenuation factors and time delays amounts to performing a deconvolution.

By the definition of $h(t)$ in Equation (3.2), we get

$$h_{\text{ls}} = \underset{h \in \mathbb{R}^m}{\arg\min} \, ||Sh - r||_2 \tag{3.9}$$

using LS. After the attenuation factors are computed, we can reconstruct the transmitted signal. Since we know the equality

$$r = s \otimes h, \tag{3.10}$$

with known vectors r and h, by deconvolving this equation we can reconstruct the transmitted signal.

It is important to observe that the transmitted signal s in Equation (3.10) may differ from the previous times we used s as the MLS. In Equation (3.10), s is the deconvolution of r and h, and it is supposed to simulate the transmitted signal as well as possible.

3.3.3. Results and Interpretation

Computing h with the LS method and thereby reconstructing the signal using the data described in Section 3.3.1 yields Figure 3.4. Here we consider 120 multipaths and a transmitted signal of 381 bits. This means that S is a $(381 + 120 - 1)$-by-120 matrix. It follows that the learning signal is a vector of 600 entries.

The vectors are plotted as follows. For each entry, the entry's value is taken to be the y-coordinate, and the entry's index is taken to be the x-coordinate. Furthermore, the lines between the points of two consecutive entries is drawn. This means that the transmitted signal is not a square wave, even though it seems like it. So, technically only the vertices portray the signals, whereas the lines do not. We plot the lines nonetheless because it makes studying the different signals easier.

As expected, we see that the transmitted signal is entirely composed of 1s and -1s. The objective is for the reconstructed signal to imitate the transmitted signal as closely as possible. This seems to have happened moderately well, in the sense that for each bit from the transmitted signal, the reconstructed signal "zigzags" around this bit.

This suggests that if we were to round the reconstructed signal to integers, the reconstructed signal might imitate the transmitted signal better. This turns out to improve the reconstruction. If we indeed round every entry of r to integers (so this could include 0), then 62 of the 381 bits get rounded to the wrong bit, which amounts to an error of approximately 16%. Moreover, only 1 out of 120 attenuation factors is greater than 1, and the residual error of h_{ls} equals 3.19.

3.3.4. Convolution

In this method, we reconstruct the transmitted signal in another way. This method relies on an assumption that is dubious. We investigate this method nonetheless because we can apply the TLS method during the procedure.

Instead of writing the received signal as a convolution of the transmitted signal and the impulse response as in Equation (3.1), we could write the transmitted signal as a convolution of the received signal and the impulse response:

$$s(t) = r(t) \otimes h(t) + w(t). \tag{3.11}$$

Similar to the derivation of Equation (3.4), this translates to the equality

$$s(t) = \sum_{k=1}^{m} b_k r(t - \mu_k) + v(t), \tag{3.12}$$

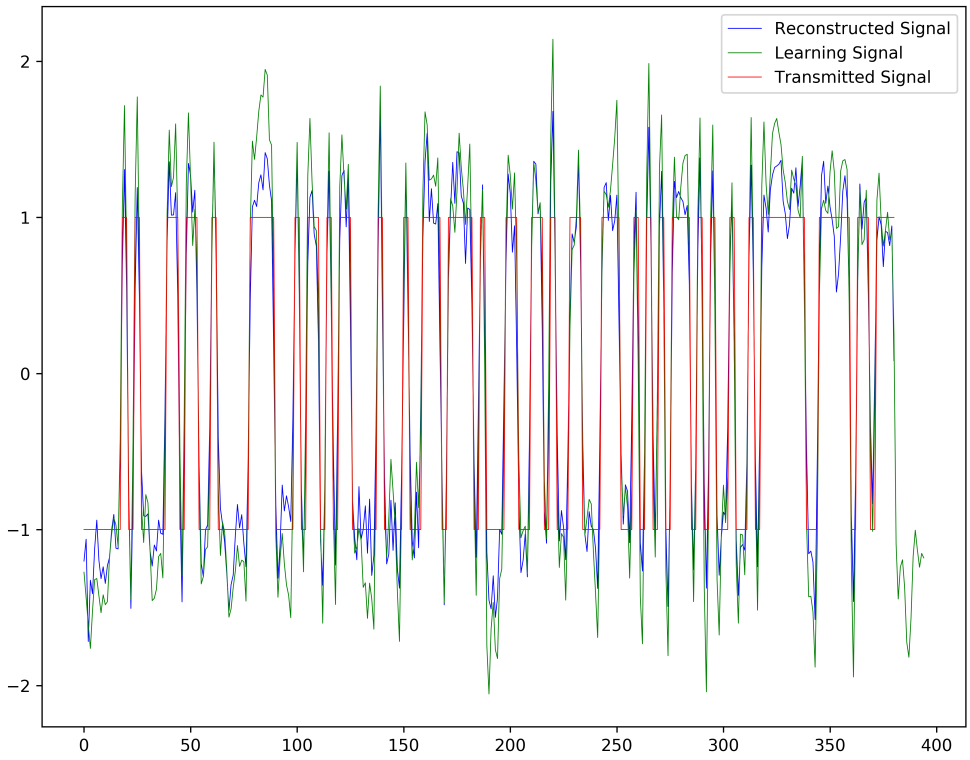


Figure 3.4: Plot of the reconstructed, learning, and transmitted signal. The LS method and a deconvolution is used to reconstruct the transmitted signal.

where b_k and μ_k are the attenuation factors and delays respectively for $k = 1, ..., m$, and $w(t)$ the Gaussian noise. This is essentially the same as Equation (3.4) except that $s(t)$ and $r(t)$ are reversed.

Computing the attenuation factors b_k again amounts to performing a deconvolution on the inconsistent system of equations

$$Rh = s, \tag{3.13}$$

which is the discretisation of Equation (3.12). The matrix R is constructed in the exact same way as S, that is

$$R = \begin{bmatrix} r & 0 & \cdots & 0 \\ -1 & r & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ -1 & -1 & \cdots & r \end{bmatrix}. \tag{3.14}$$

However, R does not have the same size as S, since the learning signal is 600 bits, whereas the transmitted signal is 381 bits. This causes R to be a $(600 + 120 - 1)$-by-120 matrix. This also means that we need to take a longer part of the MLS consisting of 719 bits.

Just like for the matrix S, the structure of R causes the matrix to have full rank and the LS problem to have a unique solution.

Reconstructing the transmitted signal from these attenuation factors and time delays are estimated is now done by a convolution instead of a deconvolution. So, instead of deconvolving $r = s \otimes h$, we now

convolve $s = r \otimes h$.

Whereas previously in the discretisation in Equation (3.7) the perturbations were only in the vector on the RHS of the equation–in which case it makes sense to use LS, in the discretisation in Equation (3.13) the perturbations are only in the matrix on the LHS.

Because of the perturbations only being in the matrix, it makes more sense to use TLS rather than LS to estimate the attenuation factors and time delays. Now, TLS would normally correct for both R and s, but by trying to modify it we could try and let it only correct R and leave s unchanged, so that no unperturbed data will be corrected.

3.3.5. Results and Interpretation Least Squares

Even though it makes more sense to only use TLS, we will still study the results of the signal reconstruction using LS because it seems to still improve the learning signal. This is formulated as

$$h_{\text{ls}} = \underset{h \in \mathbb{R}^m}{\arg\min} \, ||Rh - s||_2 \,. \tag{3.15}$$

The results for reconstructing the transmitted signal using LS and then applying a convolution for this reconstruction are given in Figure 3.5.

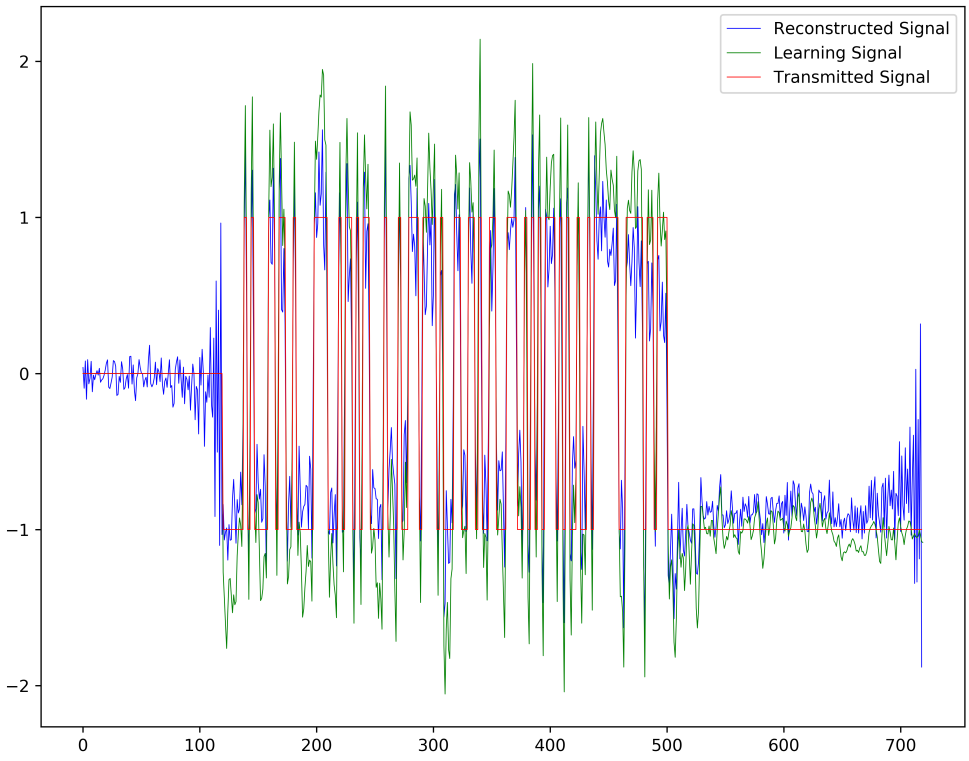


Figure 3.5: Plot of the reconstructed, learning, and transmitted signal. The LS method and a convolution to reconstruct the transmitted signal.

We note that even though LS corrects for non-existent perturbations and does not correct existent perturbations, it still seems to moderately reconstruct the transmitted signal. When we round the re-

constructed signal to integers again, 17% of the bits are incorrect. Furthermore, 3 attenuation factors are greater than 1, and the residual error of h_{ls} is 6.90.

We note that the reconstructed signal is now much longer than when we use deconvolution. Recall that the convolution of two vectors of length n and m is a vector of length $n + m - 1$, whereas for a deconvolution we seek the vector of length n, as we saw in Section 1.6. So, a convolution is always longer than the two convolving components, while deconvolution yields a vector shorter than the longest convolved vector.

3.3.6. Results and Interpretation Total Least Squares

Recall from Section 1.3 that the TLS problem for estimating the solution to an inconsistent, overdetermined system of equations is formulated as

$$x_{\text{tls}} := \underset{x \in \mathbb{R}^m}{\operatorname{argmin}} \, ||[\Delta A \mid \Delta b]||_F, \quad \text{subject to} \quad (A + \Delta A)x = b + \Delta b, \tag{3.16}$$

for $A \in \mathbb{R}^{n \times m}, b \in \mathbb{R}^n$, and that the solution exists if and only if $V_{22} \neq 0$. If it does, then it equals $x_{\text{tls}} = -V_{22}^{-1} V_{12}$, for

$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}, \quad \text{where} \quad V \in \mathbb{R}^{(m+1)\times(m+1)} \text{ and } V_{22} \in \mathbb{R}, \tag{3.17}$$

where the columns of V are the right-singular vectors of A.

For our problem, this formulation translates into

$$h_{\text{tls}} := \underset{x \in \mathbb{R}^m}{\operatorname{argmin}} \, ||[\Delta R \mid \Delta s]||_F, \quad \text{subject to} \quad (R + \Delta R)h = s + \Delta s. \tag{3.18}$$

Using the SVD of $[\Delta R \mid \Delta s]$, we get that $V_{22} \neq 0$ and so a unique solution exists. The reconstructed signal is shown in Figure 3.6. The amplitude of the reconstructed signal is huge. In fact, it is so large that the transmitted and learning signal are roughly a straight line relative to the reconstructed signal. Using this formulation, 47 attenuation factors are greater than 1. Also, the Frobenius norm of the correction matrix is 453.52.

It makes sense that this formulation does not give good results, since both the matrix R and the vector s are corrected, even though s is unperturbed. W will now try to formulate the TLS problem differently to avoid correcting for s.

We consider two different formulations. Ideally, we would like to have

$$h_{\text{tls}} := \underset{x \in \mathbb{R}^m}{\operatorname{argmin}} \, ||\Delta R||_F, \quad \text{subject to} \quad (R + \Delta R)h = s, \tag{3.19}$$

as a formulation, since in this case we only correct for the errors in the matrix R. However, if we look at how the solution $x_{\text{tls}} = -V_{22}^{-1} V_{12}$ is expressed, we see that this formulation is not a valid formulation since V_{12} and V_{22} do not exist, because there is no augmented matrix to take a SVD from. So, this formulation is not solvable as a TLS problem.

The second and last formulation we will study is

$$h_{\text{tls}} := \underset{x \in \mathbb{R}^m}{\operatorname{argmin}} \, ||[\Delta R \mid 0]||_F, \quad \text{subject to} \quad (R + \Delta R)h = s. \tag{3.20}$$

This is a proper formulation of a TLS problem. On top of that, no correction will be applied tot s. Unfortunately, this formulation will always return $h = 0$ as the solution. A proof of why this formulation always yields $h = 0$ as the solution is discussed in Appendix A.3. Moreover, $[\Delta R \mid 0]\,[x \; -1]^{\mathsf{T}}$ forms a consistent system since $\operatorname{rank}([\Delta R \mid 0]) = m$.

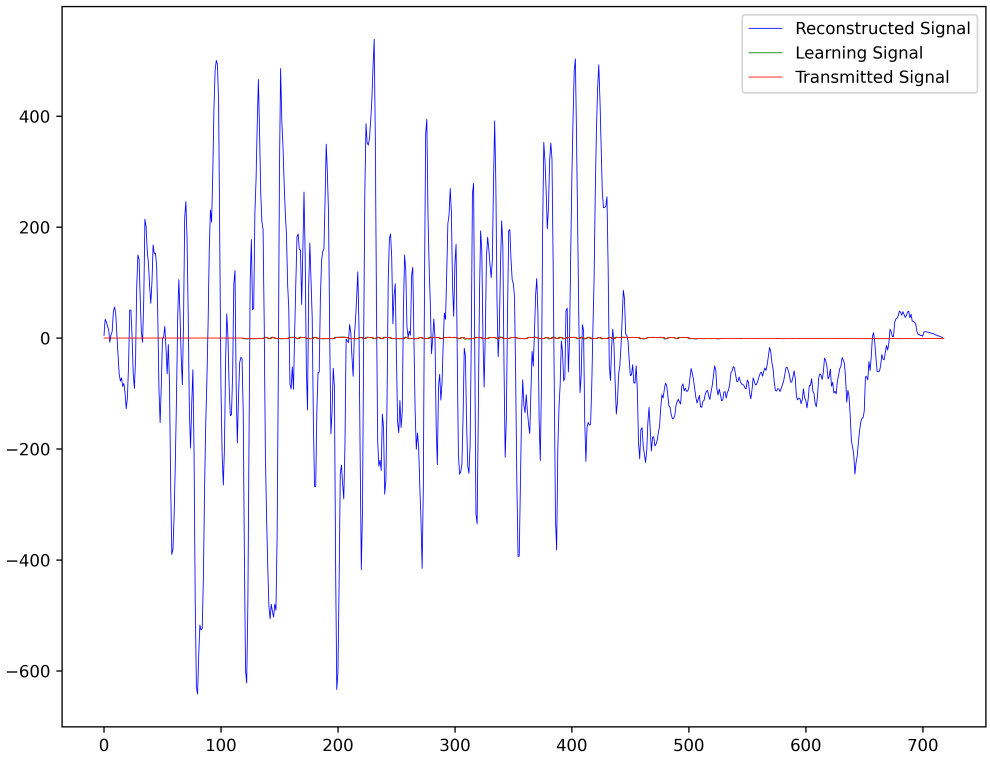


Figure 3.6: Plot of the reconstructed, learning, and transmitted signal. The TLS method and a convolution is used to reconstruct the transmitted signal.

3.4. Conclusion

We have investigated the reconstruction of an acoustic underwater signal. We considered two ways of modeling the multipath propagation of the signal. The first and most common model required a LS solution to estimate the attenuation factors and time delays. The second model suggests using TLS for this estimation, although we tested both TLS and LS on this model.

For the first model, LS seems to work moderately well to reconstruct the signal. Specifically, it corrects 84% of the bits of the signal, it leaves 119 out of 120 attenuation factors under 1, and the residual error of h_{ls} is 3.19.

For the second model, LS works moderately well too (which is surprising, given that the perturbations are in the matrix and not in the vector on the RHS). Specifically, it corrects 83% of the bits of the signal, it leaves 117 out of 120 attenuation factors under 1, and the residual error of h_{ls} is 6.90.

TLS turned out not to work well at all on this model. Not only is it hard to properly formulate a TLS problem without correcting for the RHS, but the only proper TLS formulation that we did manage to make corrects for the RHS. We suspect that the performance of TLS is caused by this formulation, as well as the dubious assumption on which this model is based.

3.5. Further Research

The main question for further research arose when we failed to give a proper TLS formulation of the problem described in Equation (3.19), while studying the second multipath propagation model. The particular question of interest is whether there exists a method on how to compute a minimal correction matrix to make an inconsistent system consistent, whilst not correcting the vector on the RHS. The formulation of this problem is

$$x_{\text{tls}} := \underset{x \in \mathbb{R}^m}{\operatorname{argmin}} ||\Delta A||_F, \quad \text{subject to} \quad (A + \Delta A)x = b. \tag{3.21}$$

Since TLS is not designed to solve such problems, a different method needs to be devised.

The whole approach taken by TLS does not work. If we write $Rh = s$ as either $[R \mid s]\,[h \; -1]^\intercal$ or $[R \mid -1]\,[h \; s]^\intercal$ and we try to find a low-rank approximation of either augmented matrix, we always end up adjusting s.

Let us go back to LS to consider an idea for solving the problem in Equation (3.21). The LS method essentially projects b onto $\operatorname{ran}(A)$ such that $Ax_{\text{ls}} - b$ is orthogonal to $\operatorname{ran}(A)$. Instead of "moving" b to the vector space spanned by the columns of A, we could instead try to move this vector space so that b lies upon it.

Consider the following two ways of going about this. If we pick a different basis for $\operatorname{ran}(A)$ such that it includes $Ax_{\text{ls}} - b$, applying a rotation matrix to each vector in this basis could "rotate" the vector space such that b is in its span. Instead of rotating every basis vector, we could also add vectors to these basis vectors such that b is in its span. In both ways, we adjust the matrix instead of the RHS. How one should go about this exactly, and whether any of these methods correspond to a minimal correction matrix would need to be investigated.

Bibliography

Bingham, N., & Fry, J. (2010). *Regression: Linear models in statistics*. Springer London. https://books.google.nl/books?id=YJkY2LGLbI8C

Birdsall, T. G., Heitmeyer, R. M., & Metzger, K. (1971). *Modulation by linear maximal shift register sequences: Amplitude, biphase and complement-phase modulation* (tech. rep.). MICHIGAN UNIV ANN ARBOR COOLEY ELECTRONICS LAB.

Boyd, S., & Vandenberghe, L. (2018). *Introduction to applied linear algebra: Vectors, matrices, and least squares*. Cambridge university press.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, *1*(3), 211–218.

Fraleigh, J., & Beauregard, R. (1987). *Linear algebra*. Addison-Wesley. https://books.google.nl/books?id=Os5UAAAAYAAJ

Golub, G., & Van Loan, C. (1996). *Matrix computations*. Johns Hopkins University Press. https://books.google.nl/books?id=mlOa7wPX6OYC

Markovsky, I., & Van Huffel, S. (2007). Overview of total least-squares methods [Special Section: Total Least Squares and Errors-in-Variables Modeling]. *Signal Processing*, *87*(10), 2283–2302. https://doi.org/https://doi.org/10.1016/j.sigpro.2007.04.004

Stigler, S. M. (1981). Gauss and the Invention of Least Squares. *The Annals of Statistics*, *9*(3), 465–474. https://doi.org/10.1214/aos/1176345451

Van Gijzen, M., & Van Walree, P. (2000). Shallow-water acoustic communication with high bit rate bpsk signals. *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No.00CH37158)*, *3*, 1621–1624 vol.3. https://doi.org/10.1109/OCEANS.2000.882172

Van Gijzen, M., Van Walree, P., Cano, D., Passerieux, J., Waldhorst, A., Weber, R., & Maillard, C. (2000). *Proceedings of the fifth european conference on underwater acoustics : Ecua 2000. volume i* (M. Zakharia, P. Dubail, & P. Chevret, Eds.). European Commission; Directorate-General for Research; Innovation.

Van Huffel, S., & Vandewalle, J. (1991). *The total least squares problem: Computational aspects and analysis*. Society for Industrial; Applied Mathematics. https://books.google.nl/books?id=VLkv9qZhN%5C_UC

Wintermantel, M., & Luder, E. (1998). Reducing the complexity of discrete convolutions by a linear transformation and modulo arithmetic. *ICSP'98. 1998 Fourth International Conference on Signal Processing (Cat. No. 98TH8344)*, 122–125.

Zeng, W.-J., Jiang, X., Li, X.-L., & Zhang, X.-D. (2010). Deconvolution of sparse underwater acoustic multipath channel with a large time-delay spread. *The Journal of the Acoustical Society of America*, *127*(2), 909–919. https://doi.org/10.1121/1.3278604

Zhang, Z. (2015). The singular value decomposition, applications and beyond. *CoRR*, *abs/1510.08532*. http://arxiv.org/abs/1510.08532

A

Appendix

In the appendix we present two short proofs and one derivation which were omitted in the text.

A.1.

Theorem 2. *Given $n+1$ points $(\alpha_0, \beta_0), \dots, (\alpha_n, \beta_n) \in \mathbb{R}^2$ with distinct x-coordinates*[1]*, there exists a unique polynomial of degree n which lies on all these points. If the degree is strictly greater than n, there exists an infinite amount of such polynomials exists.*

Proof. Let $p(x) = c_0 + c_1 x + \cdots + c_n x^n$ be an nth degree polynomial. For every point (α_i, β_i), we want to have $p(\alpha_i) - \beta_i = 0$. Showing that this holds for all points and that p is unique is equivalent to showing that the system $Ax = 0$ has exactly one solution, where

$$A = \begin{bmatrix} 1 & \alpha_0 & \cdots & \alpha_0^n & \beta_0 \\ 1 & \alpha_1 & \cdots & \alpha_1^n & \beta_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \alpha_n & \cdots & \alpha_n^n & \beta_n \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+2)}, \quad \text{and} \quad x = \begin{bmatrix} c_0 \\ \vdots \\ c_n \\ -1 \end{bmatrix} \in \mathbb{R}^{n+2}. \tag{A.1}$$

So, we need to prove that the null space of A is one-dimensional. The rank-nullity theorem yields

$$\operatorname{rank}(A) + \operatorname{nullity}(A) = n + 2. \tag{A.2}$$

Since the x-coordinates are assumed to be distinct and A is a Vandermonde matrix augmented with one column, we know that $\operatorname{rank}(A) = n + 1$. Consequently, $\operatorname{nullity}(A) = 1$.

Let $q(x) = c_0 + c_1 x + \cdots + c_k x^k$ be a polynomial of degree $k > n$. Then we need to show that

$$A' = \begin{bmatrix} 1 & \alpha_0 & \cdots & \alpha_0^k & \beta_0 \\ 1 & \alpha_1 & \cdots & \alpha_1^k & \beta_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \alpha_n & \cdots & \alpha_n^k & \beta_n \end{bmatrix} \in \mathbb{R}^{(n+1)\times(k+2)} \tag{A.3}$$

has non-trivial null space. We again have $\operatorname{rank}(A') = n + 1$, but now A' maps to $\mathbb{R}^{k+2}$. Since $k + 2 > n + 2$, we have $\operatorname{nullity}(A') > 1$. □

A.2.

The (continuous) convolution of two functions $f(x)$ and $g(x)$ is defined as

$$(f \otimes g)(x) = \int f(t) g(x - t) \mathrm{d}t. \tag{A.4}$$

[1] By definition, a function cannot map one x-coordinate to two distinct y-coordinates. Therefore there can never exist a polynomial which lies on points with equal x-coordinates.

The fundamental property of the delta function for a function $f(x)$ is

$$\int f(x)\delta(x - x_0)\mathrm{d}x = f(x_0). \tag{A.5}$$

So, for the convolution

$$r(t) = s(t) \otimes h(t) + v(t), \tag{A.6}$$

with

$$h(t) = \sum_{k=1}^{m} a_k \delta(t - \tau_k), \tag{A.7}$$

we get

$$r(t) = s(t) \otimes \sum_{k=1}^{m} a_k \delta(t - \tau_k) \tag{A.8}$$

$$= \int s(x) \sum_{k=1}^{m} a_k \delta(t - \tau_k - x)\mathrm{d}x \tag{A.9}$$

$$= \sum_{k=1}^{m} a_k \int s(x)\delta(t - \tau_k - x)\mathrm{d}x \tag{A.10}$$

$$= \sum_{k=1}^{m} a_k s(t - \tau_k). \tag{A.11}$$

A.3.

Theorem 3. *Let $A \in \mathbb{R}^{n\times m}$ and let $v_1, \ldots, v_m$ be its right-singular vectors. Then the right-singular vectors of the augmented matrix $[A \mid 0]$ are $[v_1^\intercal\ 0]^\intercal, \ldots, [v_m^\intercal\ 0]^\intercal$ and $[0 \cdots 0\ 1]^\intercal$.*

Proof. The right singular vectors are the eigenvectors of the matrix $[A \mid 0]^\intercal[A \mid 0]$. Observe

$$\left[\begin{array}{c} A^\intercal \\ \hline 0 \end{array}\right]\left[\begin{array}{c|c} A & 0 \end{array}\right] = \left[\begin{array}{c|c} A^\intercal A & 0 \\ \hline 0 & 0 \end{array}\right]. \tag{A.12}$$

Let v be any right-singular vector of $A^\intercal A$, so that we have $A^\intercal A v = \lambda v$ for some $\lambda \in \mathbb{C}$ and $v \neq 0$. Then $[v^\intercal\ 0]^\intercal$ is an eigenvector of the matrix in Equation (A.12) with eigenvalue λ. Indeed,

$$\left[\begin{array}{c|c} A^\intercal A & 0 \\ \hline 0 & 0 \end{array}\right]\begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda v \\ 0 \end{bmatrix} = \lambda \begin{bmatrix} v \\ 0 \end{bmatrix}. \tag{A.13}$$

Moreover, $[0 \cdots 0\ 1]^\intercal$ is an eigenvector of the matrix in Equation (A.12) with eigenvalue 0, since

$$\left[\begin{array}{c|c} A^\intercal A & 0 \\ \hline 0 & 0 \end{array}\right]\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = 0 \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{A.14}$$

□