Multichannel Sampling With Unknown Offsets Using Groebner Bases

Natacha Ranariboana (natacha@aims.ac.za)
African Institute Mathematical Sciences (AIMS)

Supervised by Barry Green
University of Stellenbosch

June 8, 2007

Abstract

The motivation of this work is to show an application of Groebner Bases to signal processing. We present a new method for signal reconstruction from multiple sets of samples with unknown offsets. First, we rewrite the reconstruction problem as a set of polynomial equations in the unknown signal parameters and the offsets between the sets of samples. Then we construct a Groebner basis for the corresponding affine variety. The signal parameters can then easily be derived from this Groebner basis. This derivation provides an elegant solution method for the initial nonlinear problem. Two examples are shown for the reconstruction of polynomial signals and Fourier series.

Contents

Abstract **i**

Introduction **1**

1 Multichannel Sampling with Unknown Offsets **2**

1.1 Definitions 2

1.2 Groebner bases 4

1.2.1 Zero sets or Affine Varieties and Ideals 4

1.2.2 Groebner bases 6

1.2.3 Groebner bases used to solve polynomial equations 8

1.3 Problem setup 8

1.3.1 The simple case when $\mathcal{B}$ is a polynomial basis 9

1.3.2 Other bases 10

2 Solving the Problem using Groebner bases **12**

2.1 Multichannel sampling without noise 12

2.2 Multichannel sampling under noisy conditions 16

3 Complexity and Optimisation **20**

3.1 Computational cost 20

3.1.1 Time complexity of Buchberger's algorithm 20

3.1.2 High memory of Buchberger' s algorithm 21

3.2 Optimization 21

3.2.1 For Fourier series 21

3.2.2 For Polynomial signals 22

3.3 Computer Algebra software 25

Conclusion **26**

Acknowledgment **27**

Bibliography **28**

Introduction

If a bandlimited signal is uniformly sampled at a frequency lower than twice its maximum frequency, the sampled signal is aliased, and perfect reconstruction is generally not possible. In this paper, we present a method [SVV06] to reconstruct a signal from multiple sets of samples using Groebner bases, which contradicts this well-known result known as the Shannon-Nyquist sampling theorem, that can be found in Mathworld [Wei07].

In fact, the offsets between the different sets of samples are unknown, and can take any real value. Each of the individual sets of samples is sampled uniformly, at a rate below the Nyquist rate. This is the typical setup that is used in super-resolution imaging, where a high-resolution, aliasing-free image is reconstructed from a set of low resolution, aliased images with small relative shifts as explained in [VSVV06].

Such a setup is often called Multichannel Sampling. A first contribution of this paper is to show that, in many cases, the multichannel sampling problem with unknown offsets can be written as a set of polynomial equations in both the unknown signal coefficients and the offsets. The solution can then be computed by using Groebner bases. In any practical setting, the samples are corrupted by noise, and then there is no algebraic solution. Thus a second contribution is to address this noisy version of the problem, and to show how a good approximation can be obtained from multiple Groebner bases for subsets of samples.

Our paper is structured as follows. In chapter one, we give some useful background results related to our Multichannel Sampling using Groebner bases, i.e the main ideas used for our reconstruction problem and an overview of Groebner basis theory. In the second chapter, we will formulate mathematically the Multichannel Sampling problem with unknown offsets as a set of polynomial equations. Groebner bases are then applied to our problem in the case of accurate measurements first. We will present an extended algorithm for solving the problem where the measurements are noisy. The complexity of such an algorithm is discussed in the last chapter, and some optimizations are presented that take advantage of the particular structure of the polynomials before we conclude the paper.

1. Multichannel Sampling with Unknown Offsets

1.1 Definitions

The aim in this section is to set up our main problem. Let us begin by giving some brief definitions and useful terms:

Signals

The signals are functions only of time. Other relevant independent variables do not vary with time so they are reduced to constant in our specific calculation. Signals may be scalar or vector valued.

The concept is broad, and so it is hard to give a precise definition.

Example 1.1.

In everyday life people speaking to one another exchange signals by using hand gestures.

In information theory a signal is a codified message that is the sequence of states in a communication channel that encodes a message.

Samples and sampling

By looking at the dictionary, we have many definitions of **sample**, like for example:

1. if we look at [CIDoE], it is a part of anything presented for inspection, or shown as evidence of equality of the whole.
2. if we take the definition from [WD], it is said as a small part of something intended as representative of the whole.

In signal processing, **sampling** is the reduction of a continuous signal to a discrete signal.

Example 1.2. The conversion of a sound wave (a continuous time signal) to a sequence of samples (a discrete time signal).

Figure 1.1: In the following figure, we illustrate the signal $f(t)$ and we take three offsets (t_0, t_1, t_2). Then it gives us three sets of N samples which are $y_0(n)$, $y_1(n)$ and $y_2(n)$ for $0 \leq n < N, N \in \mathbf{N}$.

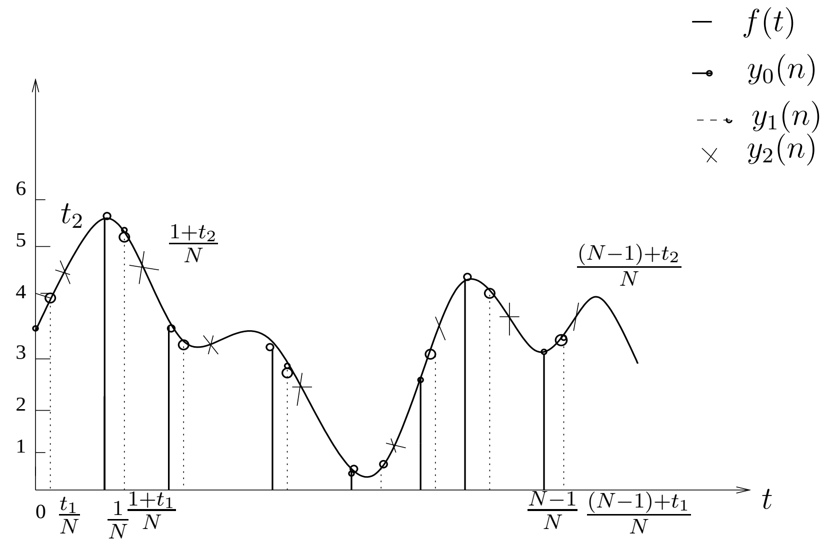

Figure 1.1: Sampling a signal with three offsets and N samples

In [Van06], we will see that there are different categories of sampling methods. For our study we will use the uniform sampling. This means if $X(t)$ is a continuous signal which is to be sampled, then we just have to measure the value of the continuous signal every τ seconds. Then we get a discrete sequence of values $X[n]$ from the signal $X(\tau)$ defined as:

$$X[n] := X(n\tau), \tag{1.1}$$

with $n = 0, 1, 2, 3, \ldots$

Offsets

Taking the definition in the Glossary of terms [GOT].

- The offsets are small jumps, or tares, in a signal due to either instrumental effects or rapid changes

In our case we sample a signal $X(t)$ using M sets of N regularly spaced samples, by an offset we mean the sequence of values t_m $(0 \leq m \leq M$, with $t_0 \ = \ 0)$. The m^{th} set of samples are given by taking $\tau = t_0 + t_m$ in (1.1).

Multichannel Sampling

By Multichannel sampling we mean taking sets of samples with small relative (to τ) offsets in order to construct the original signal by combining the set of all samples.

1.2 Groebner bases

The concept of Groebner basis was introduced by Bruno Buchberger in 1965. This has found applications in various fields of Mathematics as well as in Science and Engineering as we can see in [BW98]. A comprehensive treatment of the theory of Groebner basis is far beyond what can be done in one section. However, there are only a few key ideas behind Groebner basis theory. It is the objective of this section to explain these ideas as simply as possible and to give an overview of the immediate applications for solving our multichannel sampling problem.

Let us consider polynomials in the n complex variables $X_0, \cdots, X_{n-1}$. A polynomial p can be written as:

$$p = \sum_{d} a_d X^d, \quad a_d \in \mathbb{C} \tag{1.2}$$

where the sum is over a finite number of n-tuples $d = (d_0, \ldots, d_{n-1})$, and $X^d = X_0^{d_0} X_1^{d_1} \ldots X_{n-1}^{d_{n-1}}$ is called a monomial.

In the following, we will denote by $\mathbb{C}[X_0, ..., X_{n-1}]$ the ring of (complex) polynomials in the variables $X_0, \cdots, X_{n-1}$.

1.2.1 Zero sets or Affine Varieties and Ideals

Definition 1.1. (Affine Varieties) Let $p_0, \cdots, p_{s-1} \in \mathbb{C}[X_0, ..., X_{n-1}]$. Then:

$$V(p_0, \cdots, p_{s-1}) = \{(c_0, \ldots, c_{n-1}) \in \mathbb{C}^n \,|p_i(c_0, \ldots, c_{n-1}) = 0, \ \forall\ 0 \leq i < s\} \tag{1.3}$$

is called the affine variety defined by $p_0, \cdots, p_{s-1}$.

In the linear case, to find the points of the variety $V(p_0, \cdots, p_{s-1})$ with:

$$p_i(X_0, \cdots, X_{n-1}) = a_{i0}X_0 + \ldots + a_{i(n-1)}X_{n-1} + b_i, \ i = 0, \ldots, s-1 \tag{1.4}$$

is equivalent to solving the system

$$AX + b = 0, \tag{1.5}$$

with $A = (a_{ij})$ and $b = (b_0, \ldots, b_{s-1})^T$.

We usually solve this system (1.5) using Gaussian Elimination. To perform Gaussian Elimination starting with:

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0n-1} \\ \vdots & & & \vdots \\ a_{(s-1)0} & a_{(s-1)0} & & a_{(s-1)(n-1)} \end{bmatrix} \begin{bmatrix} X_0 \\ \vdots \\ X_{n-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ \vdots \\ b_{s-1} \end{bmatrix}$$

Compose the "augmented matrix equation"

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0n-1} & b_0 \\ \vdots & & & \vdots & \vdots \\ a_{(s-1)0} & a_{(s-1)0} & & a_{(s-1)(n-1)} & b_{s-1} \end{bmatrix} \tag{1.6}$$

Here the column vector in the variable X is carried along for labelling the matrix rows. Now, perform elementary row operations to put the augmented matrix into the upper triangular form:

$$\begin{bmatrix} a'_{00} & a'_{01} & \cdots & a'_{0n-1} & b'_0 \\ 0 & a'_{11} & \cdots & a'_{1n-1} & b'_1 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & a'_{(s-1)(n-1)} & b'_s - 1 \end{bmatrix} \tag{1.7}$$

The solution of the system is obtained by back substitution.

Then by Buchberger's algorithm, we extend the Gaussian elimination to the case of polynomial equations. The set of polynomials obtained after applying Buchberger's algorithm is called a Groebner basis. Similarly with the linear case, when we perform elementary row operations, we define an ordering of the terms of (1.4), i.e. the monomials of $X_0, \cdots, X_{n-1}$. In the polynomial case, there are different ways to order monomials according to the variables and exponents. In our case we will take the lex ordering as in [SVV06].

Definition 1.2. (Lex ordering) Let $d = (d_0, \cdots, d_{n-1})$ and $d' = (d'_0, \cdots, d'_{n-1})$ be two $n-$tuples representing positive integer exponents of the monomial $X^d, X^{d'}$.

We say that $d \succ_{lex} d'$, if $d - d' \ \in \mathbb{Z}^n$, and in this difference the left most non zero entry is positive.
We say that $X^d \succ_{lex} X^{d'}$ if $d \succ_{lex} d'$.

With the type of ordering we need also to define the order between the different variables. As in [SVV06], we will assume that the terms of each polynomial are ordered in descending order according to the lex ordering and with $X_0 > X_1 > \cdots > X_{n-1}$.

Definition 1.3. The multidegree of a polynomial p, denoted by $\mathrm{multideg}(p)$ is the largest exponent of monomials of p according to the lex ordering as in [SVV06].

The leading term, denoted by $LT(p)$ is the term of p with the largest exponent. The total degree of p is given by the maximum sum of the exponent vectors d of its terms.

Example 1.3. Let us consider a polynomial

$$p = 5X_0X_1^2 + X_0X_1X_2^4 + 6X_1X_2^2 \tag{1.8}$$

Using lex ordering and $X_0 > X_1 > X_2$, we have $X_0X_1^2 > X_0X_1X_2^4 > X_1X_2^2$ and (1.8) is ordered in descending lexicographic order.

$\mathrm{multideg}(p) = (1, 2, 0),\ LT(p) = 5X_0X_1^2$ and the total degree is $1 + 1 + 4 = 6$

As in Gaussian elimination, we will try to replace the initial set of equations by another set which defines the same variety to make easier the resolution of our system. Instead of scalar coefficients in the linear combinations with the Gaussian elimination, we will now use polynomial coefficient to combine different equations. At this stage, we need the notion of ideals.

Definition 1.4. (Ideal): A subset $I \subset \mathbb{C}[X_0, ..., X_{n-1}]$ is an ideal if it satisfies :

1. $0 \in I$.
2. If $p, q \in I$, then $p + q \in I$.
3. If $p \in I$ and $a \in \mathbb{C}[X_0, ..., X_{n-1}]$, then $ap \in I$.

If $p_0, \cdots, p_{s-1}$ are polynomials, then we set

$$I = \langle p_0, \cdots, p_{s-1} \rangle = \left\{ \sum_{i=0}^{s-1} a_i p_i : a_i \in \mathbb{C}[X_0, ..., X_{n-1}] \right\} \tag{1.9}$$

We call I the ideal generated by $p_0, \cdots, p_{s-1}$. See [SVV06]

1.2.2 Groebner bases

Given a basis $F = \{f_i\}$ generating an ideal I, the central idea in Groebner basis theory is to use F to find a basis G for I which has the good property, namely that the remainder of the division of any polynomial by G is unique. Such bases is known as Groebner basis.

This gives us a method for solving the Ideal membership problem see [SVV06]. In fact if the remainder of the division of a polynomial p by G is zero, then $p \in I$, otherwise not.

Groebner bases have many properties, for example they generalise the structure of the system (1.7) which will help us to solve our problems later in this essay. In addition, we have also the following fundamental theorem as given in [SVV06].

Theorem 1.5. *(Hilbert Basis Theorem): Every ideal of the ring of polynomials of n variables over a field has a finite generating set. That is, $I = \langle g_0, \cdots, g_{u-1} \rangle$ for some $g_0, \cdots, g_{u-1} \in I$. In particular, it is always possible to choose $g_0, \cdots, g_{u-1}$ so that they form a Groebner basis.*

The question now is: How to obtain the Groebner basis? The answer is given by Buchberger's algorithm, which is a method of transforming a given set of generators for a polynomial ideal into a Groebner basis. One can view it as an extension of Gaussian reduction. In the polynomial case, equations are combined using polynomial coefficients instead of scalar ones in the linear case. Then, by analogy we define S-polynomials as follows.

Definition 1.6. : Let p_0, p_1 be two non-zero polynomials in $X_0, \cdots, X_{n-1}$. If $\mathrm{multideg}(p_0) = d$ and $\mathrm{multideg}(p_1) = d'$, then let $d" = (d"_0, \cdots, d"_{n-1})$, where $d"_i = \max(d_i, d'_i)$.

The $S-polynomial$ of p_0 and p_1 is defined as the linear combination

$$S(p_0, p_1) = \frac{X^{d"}}{LT(p_0)}p_0 - \frac{X^{d"}}{LT(p_1)}p_1. \tag{1.10}$$

Using this definition of $S-$polynomials we can easily see with the following theorem, from [SVV06], giving a condition for a basis G to be a Groebner Basis or not.

Theorem 1.7. *Let I be a polynomial ideal. Then a basis $G = \{g_0, \cdots, g_{u-1}\}$ is a Groebner Basis for I if only and only if, for all pairs $i \neq j$, the remainder on division of $S(g_i, g_j)$ by G (listed in some order) is zero.*

Note that by taking a linear combination of the f_i in a basis F, we can have leading terms cancelling and new leading terms appearing which were not there before. For this reason, to construct a Groebner basis, one has to increase initially the number of elements of the basis. Such an extension ends when the conditions of Theorem 1.7 are satisfied. This algorithm is due to Buchberger and is given in Algorithm 1 in [SVV06].

Before going into the algorithm itself, we make some connections to two algorithms familiar to complexity theorists. This also helps us to understand better the algorithm.

- If we specialise the algorithm to the one variable case, we have essentially Euclid' s algorithm for computing the GCD of several univariate polynomials.
- If we restrict each variable to be linear, then we have essentially the Gaussian elimination algorithm.

Algorithm 1. Let $I = \langle p_0, \cdots, p_{s-1} \rangle \neq 0$ be a polynomial ideal. Then a Groebner Basis for I can be constructed in a finite number of steps by the following algorithm:

Input: $P = (p_0, \cdots, p_{s-1})$
Output: a Groebner basis $G = (g_0, \cdots, g_{u-1})$ for I, with $P \subseteq G$.
    $G := P$
Repeat
    $G' := G$
    For each pair $(p, q), p \neq q$ in G' do
    $S := \overline{S(p, q)}^{G'}$
    If $S \neq 0$ then $G := G \cup S$
until $G = G'$.

Often our process (Buchberger Algorithm) gives a basis which is larger than necessary, and it is not unique. Therefore, we define what we call a reduced Groebner basis. This one is unique for a given monomial ordering. Morover the leading coefficient of each element of the basis is 1 and no monomial in any element of the basis is in the ideal generated by the leading terms of the other elements of the basis.

1.2.3 Groebner bases used to solve polynomial equations

In this section, we will see how we can use a Groebner Basis corresponding to a system of polynomial equations to simplify the system.

Definition 1.8. (Elimination ideal)

The elimination ideal I_k is the set of all polynomials that can be deduced from the original system and contain only the variables $X_k, \cdots, X_{n-1}$, as given in [CLO96]

$$I_k = I \cap \mathbb{C}[X_k, \cdots, X_{n-1}]. \tag{1.11}$$

In fact, if we can find a basis for each ideal I_k for $k = 0, \ldots, n-1$, we can determine the solutions of the original systems using back substitution. As we can see that $I_{k+1} \subseteq I_k$ for any $k \geqslant 0$. Therefore, we can extend a solution of the system of equations associated to I_{k+1} to the system associated to I_k by computing the values of the variable X_k.

Now, we can ask ourself how we can compute the Groebner basis of each I_k. The elimination theorem [SVV06] says that the Groebner basis of each I_k can be determined from the Groebner basis of I:

Theorem 1.9. *(Elimination theorem) Let $I \subset [X_0, \cdots, X_{n-1}]$ be an ideal and let G be a Groebner basis of I with respect to lex order where $X_0 > X_1 > \cdots > X_{n-1}$. Then, for every $1 \leq k \leq n$, the set*

$$G_k = G \cap \mathbb{C}[X_k, \cdots, X_{n-1}] \tag{1.12}$$

is a Groebner basis for the k-th elimination ideal I_k.

From now on, we have the tools that we need for solving our problem. Therefore, let us set up our problem.

1.3 Problem setup

A mathematical formulation of the multichannel sampling problem presented in the introduction is given below. This setup is the same as the one used in [Van06], so the reader can find a more detailed description and some examples in that reference.

Let us consider a finite L-dimensional Hilbert space $\mathcal{H}$ spanned by a basis $\mathcal{B} = \{\varphi_l(t)\}_{l=0,\ldots,L-1}$. As in [SVV06], we assume that the function representing the signal is periodic, of period 1 and lies in the space $\mathcal{H}$. The time t can then be taken modulo 1, and we restrict our analysis to the interval $[0, 1)$. An arbitrary signal $f(t)$ in $\mathcal{H}$ can then be written as:

$$f(t) = \sum_{l=0}^{L-1} \alpha_l \varphi_l(t), \tag{1.13}$$

where α_l is the expansion coefficient corresponding to $\varphi_l(t)$. We sample $f(t)$ uniformly with N samples, resulting in

$$y_0(n) = f(\frac{n}{N}) = \sum_{l=0}^{L-1} \alpha_l \varphi_l(\frac{n}{N}),\ 0 \leq n < N. \tag{1.14}$$

If we choose the number of samples $N < L$, it is not possible to compute the L expansion coefficients α_l from the N samples $y_0(n)$. We will therefore consider M such sets of samples, with for each set a relative offsets t_m $(0 \leq m < M$ and $t_0 = 0)$. The m^{th} set of samples can be written as

$$y_m(n) = f\left(\frac{n+t_m}{N}\right) = \sum_{l=0}^{L-1} \alpha_l \varphi_l \left(\frac{n+t_m}{N}\right) \tag{1.15}$$

or in a vector as

$$y_m = \Phi_{t_m} \alpha,$$

The different sample vectors y_m can then be combined into a large vector y (and similarly for Φ_t):

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{M-1} \end{pmatrix} = \begin{pmatrix} \Phi_{t_0} \\ \Phi_{t_1} \\ \vdots \\ \Phi_{t_{M-1}} \end{pmatrix} \alpha = \Phi_t \alpha. \tag{1.16}$$

where both the expansion coefficients $\alpha_0, \ldots, \alpha_{N-1}$ and the offsets t (which appear in the matrix Φ_t) are unknown. Let us illustrate this setup with two examples:

1.3.1 The simple case when $\mathcal{B}$ is a polynomial basis

Example 1.4. Consider the case where $\mathcal{B}$ is given by the functions $\varphi_l(t) = t^l,\ l = 0, \ldots, L-1$ with $L = 3$. Assume that we took a set of two sets $(M = 2)$ of two samples $(N = 2)$. Consider the signal parameter vector $\alpha = (64, -24, -4)^T$ and the displacements $t = (0, 1/4)^T$. In this case, the two sets of measurements would be $y_0 = (-4, 0)^T$ and $y_1 = (-6, 6)^T$. Therefore, the system that we want to solve is

$$\begin{pmatrix} 0 & 0 & 1 \\ \frac{1}{4} & \frac{1}{2} & 1 \\ \frac{1}{4}t_1^2 & \frac{1}{2}t_1 & 1 \\ \left(\frac{1}{2} + \frac{1}{2}t_1\right)^2 & \frac{1}{2} + \frac{1}{2}t_1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_2 \\ \alpha_1 \\ \alpha_0 \end{pmatrix} = \begin{pmatrix} -4 \\ 0 \\ -6 \\ 6 \end{pmatrix}. \tag{1.17}$$

In the above example, we obtain a set of nonlinear polynomial equations. The equations are linear in the unknown signal coefficient α. Thanks to the specific choice of a polynomial basis $\{\varphi_l(t) = t^l\}$, the equations are polynomials in the offsets t. Note that for an arbitrary basis $\{\varphi_l(t)\}$, this is not valid. However, for certain bases, we can rewrite the equations (1.16) as a set of polynomial equations using a change of variables. We will illustrate this in the example 2.1 in the next chapter.

1.3.2 Other bases

The procedure in the example above can be applied to any polynomial basis. More generally, using a change of variable, we can solve the problem when the basis is a set of functions $\{\varphi_l(t) = h(t)^l\}$, with $h(t)$ an invertible function. We will see the reason later in the equation (1.22). An important case is when $h(t) = e^{2\pi jt}$, which gives the Fourier series. In fact, consider the case of a complex signal of the form

$$f(t) = \sum_{l=-K}^{K} \alpha_t \varphi_l(t), \tag{1.18}$$

with $\varphi_l(t) = e^{j2\pi lt}$ and $j^2 = -1$. Note that the basis functions and coefficients are now indexed from $-K$ to K, which is the usual way of indexing for Fourier series. For comparison with the previous example, we will assume here that $K = (L-1)/2$, with L odd. The samples are given by

$$y_m(n) = f\left(\frac{n+t_m}{N}\right) = \sum_{l=-K}^{K} \alpha_l W^{nl} e^{j2\pi \frac{lt_m}{N}} \quad \text{for} \quad 0 \le n < N \tag{1.19}$$

with $W = e^{\frac{j2\pi}{N}}$ and $z_m = e^{\frac{j2\pi t_m}{N}}$

$$y_m(n) = f\left(\frac{n+t_m}{N}\right) = \sum_{l=-K}^{K} \alpha_l W^{nl} z_m^l. \tag{1.20}$$

We multiply (1.20) with z_m^K to eliminate negative exponents:

$$z_m^K y_m(n) = z_m^K f\left(\frac{n+t_m}{N}\right) = \sum_{l=-K}^{K} \alpha_l W^{nl} z_m^{l+K}. \tag{1.21}$$

For each sample, this can be rewritten as a polynomial constraint

$$p_{mN+n} = \sum_{l=-K}^{K} \alpha_l W^{nl} z_m^{l+K} - z_m^K y_m(n) = 0. \tag{1.22}$$

Example 1.5. Assume $L = 5$ i.e. the input signal is represented by the parameter vector $\alpha = (\alpha_{-2}, \ldots, \alpha_2)$, where each entry is a complex value. For this example, we assume

$$\alpha = (3, 2-j, 1, 2+j, 3)^T. \tag{1.23}$$

We suppose that $M = 2$ sets of $N = 4$ samples are taken from the input signal, with the displacements $t = (0, 1/8)^T$. In this case, two sets of measurements are

$$y_O = (11, -7, 3, -3)^T y_1 = (1+\sqrt{2}, 1-3\sqrt{2}, 1-\sqrt{2}, 1+3\sqrt{2})^T. \tag{1.24}$$

Applying (1.22), we obtain these equations:

$$
\begin{aligned}
p_0 &= \alpha_2 + \alpha_1 + \alpha_0 + \alpha_{-1} + \alpha_{-2} - 11 \\
p_1 &= -\alpha_2 + j\alpha_1 + \alpha_0 - j\alpha_{-1} - \alpha_{-2} + 7 \\
p_2 &= \alpha_2 - \alpha_1 + \alpha_0 - \alpha_{-1} + \alpha_{-2} - 3 \\
p_3 &= -\alpha_2 - j\alpha_1 + \alpha_0 + j\alpha_{-1} - \alpha_{-2} + 3 \\
p_4 &= \alpha_2 z_1^4 + \alpha_1 z_1^3 + \alpha_0 z_1^2 + \alpha_{-1} z_1 + \alpha_{-2} - (1+\sqrt{2}) z_1^2 \\
p_5 &= -\alpha_2 z_1^4 + j\alpha_1 z_1^3 + \alpha_0 z_1^2 - j\alpha_{-1} z_1 - \alpha_{-2} - (1-3\sqrt{2}) z_1^2 \\
p_6 &= \alpha_2 z_1^4 - \alpha_1 z_1^3 + \alpha_0 z_1^2 - \alpha_{-1} z_1 + \alpha_{-2} - (1-\sqrt{2}) z_1^2 \\
p_7 &= -\alpha_2 z_1^4 - j\alpha_1 z_1^3 + \alpha_0 z_1^2 + j\alpha_{-1} z_1 - \alpha_{-2} - (1+3\sqrt{2}) z_1^2,
\end{aligned}
\tag{1.25}
$$

where the complex variable $z_1 = e^{j2\pi\frac{t_1}{4}}$ represents the displacement.

2. Solving the Problem using Groebner bases

Now we have defined our problem and we possess the necessary tools, we discuss the approach we want to follow to solve it. After a possible change of variable, if necessary, to write equation (1.16) in polynomial form, we can apply Buchberger's algorithm and use the Groebner bases to solve our problem. In real life, the measurements are often noisy and so our precision is limited. Therefore, in practice there are two cases to consider:

- the version of the problem if we have the exact data.
- the second version if our measurements are noisy.

2.1 Multichannel sampling without noise

In this part, we need to follow the algorithm below as given in [Van06] to solve our problem.

Algorithm 2. Algorithm for multichannel sampling with unknown offsets using Groebner bases

1. Write out equation (1.16) describing the samples as a function of the signal coefficients.
2. If necessary, perform a change of variables to convert the equations into a set of polynomial equations.
3. Compute a Groebner basis for the ideal generated by a set of polynomial equations using Buchberger's algorithm.
4. Use back substitution to compute the offsets and signal parameters from the Groebner bases.
5. If necessary, eliminate solutions that are not valid (e.g. offset values not on the unit circle in the Fourier case).

Example 2.1. (Polynomial signals) First, we reconsider the equations obtained in Example 1.4. In fact, we have considered a second degree polynomial signal with two sets of two samples ($L = 3, M = 2, N = 2$). We can represent the set of solutions of (1.17) as the points of the affine variety defined by the set of polynomials:

$$\begin{aligned} p_0 &= \alpha_0 + 4 \\ p_1 &= \tfrac{1}{4}\alpha_2 + \tfrac{1}{2}\alpha_1 + \alpha_0 \\ p_2 &= \tfrac{1}{4}\alpha_2 t_1^2 + \tfrac{1}{2}\alpha_1 t_1 + \alpha_0 + 6 \\ p_3 &= \tfrac{1}{4}\alpha_2 t_1^2 + \tfrac{1}{2}\alpha_2 t_1 + \tfrac{1}{4}\alpha_2 + \tfrac{1}{2}\alpha_1 t_1 + \tfrac{1}{2}\alpha_1 + \alpha_0 - 6, \end{aligned} \tag{2.1}$$

in the variables $\alpha_0,\ \alpha_1,\ \alpha_2$ and t_1. We fix the ordering of variables as $\alpha_2 > \alpha_1 > \alpha_0 > t_1$ and we use lex ordering on the monomials.

At the first step of Buchberger's algorithm, we find that

$$\begin{aligned}
S(p_0,p_1) &= 4\alpha_2 - 2\alpha_1\alpha_0 - 4\alpha_0^2 = (-2\alpha_1 - 4\alpha_0)p_0 + 16p_1, \\
S(p_0,p_2) &= \alpha_2 t_1^2 - \tfrac{1}{2}\alpha_1\alpha_0 t_1 - \alpha_0^2 - 6\alpha_0 \\
&= (\tfrac{1}{2}\alpha_1 t_1 - \alpha_0 - 4t_1^2 - 2)p_0 + 4t_1^2 p_1 - 2\alpha_1 t_1^2 + 2\alpha_1 t_1 + 16t_1^2 + 8, \\
S(p_0,p_3) &= -\tfrac{1}{2}\alpha_2\alpha_0 t_1 - \tfrac{1}{4}\alpha_2\alpha_0 + \alpha_2 t_1^2 - \tfrac{1}{2}\alpha_1\alpha_0 t_1 - \tfrac{1}{2}\alpha_1\alpha_0 - \alpha_0^2 + 6\alpha_0 \\
&= (-\tfrac{1}{2}\alpha_2 t_1 - \tfrac{1}{4}\alpha_2 - \tfrac{1}{2}\alpha_1 t_1 - \tfrac{1}{2}\alpha_1 - \alpha_0 - 4t_1^2 - 8t_1 + 6)p_0 \\
&\quad +(4t_1^2 + 8t_1 + 4)p_1 - 2\alpha_1 t_1^2 - 2\alpha_1 t_1 + 16t_1^2 + 32t_1 - 24, \\
S(p_1,p_2) &= \tfrac{1}{8}\alpha_1 t_1^2 - \tfrac{1}{8}\alpha_1 t_1 + \tfrac{1}{4}\alpha_0 t_1^2 - \tfrac{1}{4}\alpha_0 - \tfrac{3}{2} \\
&= (\tfrac{1}{4}t_1^2 - \tfrac{1}{4})p_0 + \tfrac{1}{8}\alpha_1 t_1^2 - \tfrac{1}{8}\alpha_1 t_1 - t_1^2 - \tfrac{1}{2}, \\
S(p_1,p_3) &= -\tfrac{1}{8}\alpha_2 t_1 - \tfrac{1}{16}\alpha_2 + \tfrac{1}{8}\alpha_1 t_1^2 - \tfrac{1}{8}\alpha_1 t_1 - \tfrac{1}{8}\alpha_1 + \tfrac{1}{4}\alpha_0 t_1^2 - \tfrac{1}{4}\alpha_0 + \tfrac{3}{2} \\
&= (\tfrac{1}{4}t_1^2 + \tfrac{1}{2}t_1)p_0 + (-\tfrac{1}{2}t_1 - \tfrac{1}{4})p_1 + \tfrac{1}{8}\alpha_1 t_1^2 + \tfrac{1}{8}\alpha_1 t_1 - t_1^2 - 2t_1 + \tfrac{3}{2}, \\
S(p_2,p_3) &= -\tfrac{1}{2}\alpha_2 t_1 - \tfrac{1}{4}\alpha_2 + \tfrac{1}{2}\alpha_1 + 12 = (2t_1+1)p_0 + (-2t_1 - 1)p_1 + \alpha_1 t_1 - 8t_1 + 8.
\end{aligned} \tag{2.2, 2.3}$$

Therefore, we add the remainders that are non-zero to the basis:

$$\begin{aligned}
p_4 &= \overline{S(p_0,p_2)}^G = -2\alpha_1 t_1^2 + 2\alpha_1 t_1 + 16t_1^2 + 8 \\
p_5 &= \overline{S(p_0,p_3)}^G = -2\alpha_1 t_1^2 - 2\alpha_1 t_1 + 16t_1^2 + 32t_1 - 24 \\
p_6 &= \overline{S(p_2,p_3)}^G = \alpha_1 t_1 - 8t_1 + 8.
\end{aligned} \tag{2.4}$$

The remainders of $S(p_1,p_2)$ and $S(p_1,p_3)$ are not added, because they are the same as polynomials p_4 and p_5, respectively. Following the same procedure, in the second iteration, we find that only $S(p_2,p_6)$ and $S(p_4,p_6)$ give distinct, non-zero remainders. We add the polynomials

$$\begin{aligned}
p_7 &= \overline{S(p_2,p_6)}^G = -2\alpha_1 - 48 \\
p_8 &= \overline{S(p_4,p_6)}^G = 32t_1 - 8
\end{aligned} \tag{2.5}$$

to the basis. In the following iteration all remainders are zero and by Theorem 1.7 we conclude that $p_0, \cdots, p_8$ is a Groebner bases. After reducing the elements of the basis. We can see that p_2, p_3, p_4, p_5, p_6 can be removed and the final basis is given by p_0, p_1, p_7, p_8. In order to apply the elimination theorem, we rename the elements of the basis as:

$$\begin{aligned}
g_0 &= \frac{1}{4}\alpha_2 + \frac{1}{2}\alpha_1 + \alpha_0 \\
g_1 &= -2\alpha_1 - 48 \\
g_2 &= \alpha_0 + 4 \\
g_3 &= 32t_1 - 8.
\end{aligned} \tag{2.6}$$

With the definition of Elimination ideal, we have:
$I = \langle g_0, g_1, g_2, g_3\rangle$ so $I \cap \mathbb{C}[\alpha_1, \alpha_0, t_1] = I_1$. And similarly we get I_2 and I_3.Then the elimination ideals are $I_1 = \langle g_1, g_2, g_3\rangle$, $I_2 = \langle g_2, g_3\rangle$, and $I_3 = \langle g_3\rangle$. The solution of the problem can be

obtained by computing the points of the affine variety associated to I_3 and extending it by back substitution to I_2, I_1 and I. We easily find that the unique solution is given by $t_1 = \frac{1}{4}$, $\alpha_0 = -4, \alpha_1 = -24$ and $\alpha_2 = 64$.

The procedure described in the above example can be applied to any multichannel sampling problem in the polynomial space $\mathcal{H}$. For any value of the variables L, M and N, the equations in (1.16) form a set of polynomial equations and we can therefore compute the parameter values by calculating a Groebner basis for the corresponding ideal. Similarly, the same algorithm can be applied to Fourier series, using the change of the variables given in Section 1.3.

Example 2.2. (Fourier series)[Van06].
Assume $K = 2$ (i.e $L = 5$).Then the input signal is represented by the parameter vector

$\alpha = (\alpha_{-2}, \dots, \alpha_2)$, where $\alpha_i \in \mathbb{C}$ for $i \in \{-2, -1, 0, 1, 2\}$.

In our example, let us take:

$$\alpha = (3, 2 - j, 1, 2 + j, 3)^T.$$

We suppose that $M = 2$ sets of $N = 4$ samples are taken from the input signal, with the displacement $t = (0, \frac{1}{2})$. So by (1.18)

$$f(t) = \sum_{l=-2}^{2} \alpha_l (e^{j2\pi t})^l, \tag{2.7}$$

and

$$y_m(n) = f\left(\frac{n + t_m}{N}\right),$$

for $n = 0, 1, 2, 3$, $N = 4$ and $m = 0, 1$.
Then $y_0 = f(\frac{n}{N})$ and $y_1 = f\left(\frac{n + t_1}{N}\right)$ with $t_1 = \frac{1}{2}$.

When n varies it gives $y_0 = [f(0),\ f(\frac{1}{4}),\ f(\frac{1}{2}),\ f(\frac{3}{4})]$ by using (2.7).

For $t = 0$, $e^{j2\pi t} = 1$ which gives us:

$$f(0) = \sum_{l=-2}^{2} \alpha_l = 3 + 2 - j + 1 + 2 + j + 3 = 11.$$

Doing so, we can find that:

$$y_0 = (11, -7, 3, -3)^T. \tag{2.8}$$

For $t_1 = \frac{1}{2}$. Then: $y_1 = [f(\frac{1}{8}),\ f(\frac{3}{8}),\ f(\frac{5}{8}),\ f(\frac{7}{8})]$.

We know that $(e^{j\frac{2\pi}{8}}) = (e^{j\frac{\pi}{4}}) = \frac{\sqrt{2}}{2}(1 + j)$.

So:

$$\begin{aligned}
f(\frac{1}{8}) &= 3\left[\frac{\sqrt{2}}{2}(1+j)\right]^{-2} + (2-j)\left[\frac{\sqrt{2}}{2}(1+j)\right]^{-1} + 1 + (2+j)\left[\frac{\sqrt{2}}{2}(1+j)\right] + 3\left[\frac{\sqrt{2}}{2}(1+j)\right]^{2} \\
&= 6\left(\frac{-j}{2}\right) + \frac{2}{\sqrt{2}}\frac{(2-j)(1+j)}{2j} + 1 + \frac{\sqrt{2}}{2}(2+j)(1+j) + 3(\frac{1}{2})(2j) \\
&= -3j + \sqrt{2}\frac{(1-3j)}{2} + 1 + \frac{\sqrt{2}}{2}(3j+1) + 3j \\
&= \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} + 1 \\
&= 1 + \sqrt{2}.
\end{aligned}$$

Doing so for all the elements of y_1, we have:

$$y_1 = \left(1+\sqrt{2}, 1-3\sqrt{2}, 1-\sqrt{2}, 1+3\sqrt{2}\right)^T. \tag{2.9}$$

Thus, the two sets of measurements are:

$$\begin{aligned}
y_0 &= (11, -7, 3, -3)^T \\
y_1 &= \left(1+\sqrt{2}, 1-3\sqrt{2}, 1-\sqrt{2}, 1+3\sqrt{2}\right)^T.
\end{aligned} \tag{2.10}$$

Now, let us reconstruct our signal using algorithm 2. For that, we pretend that we just have the data (2.9), and we want to find the signal parameters, i.e the vector α and the offset t_1. As shown in the section 1.3.2, first we need to calculate the W^{nl} with $W = e^{j2\frac{\pi}{4}}$ for $n = 0, 1, 2, 3$ and $l = -2, \cdots, 2$. So:

$$\begin{aligned}
W^{0l} &= (1\ , 1\ , 1\ , 1\ , 1) \\
W^{1l} &= (-1, -j, 1\ , j\ , -1) \\
W^{2l} &= (1\ , -1, 1\ , -1,\ 1) \\
W^{3l} &= (-1,\ j,\ 1, -j, -1).
\end{aligned}$$

Then for $t_0 = 0$ we have $z_0 = 1$. So for our case, the system in (1.21) becomes:

$$\begin{aligned}
p_0 &= \alpha_2 + \alpha_1 + \alpha_0 + \alpha_{-1} + \alpha_{-2} - 11, \\
p_1 &= -\alpha_2 + j\alpha_1 + \alpha_0 - j\alpha_{-1} - \alpha_{-2} + 7, \\
p_2 &= \alpha_2 - \alpha_1 + \alpha_0 - \alpha_{-1} + \alpha_{-2} - 3, \\
p_3 &= -\alpha_2 - j\alpha_1 + \alpha_0 + j\alpha_{-1} - \alpha_{-2} + 3, \\
p_4 &= \alpha_2 z_1^4 + \alpha_1 z_1^3 + \alpha_0 z_1^2 + \alpha_{-1} z_1 + \alpha_{-2} - (1+\sqrt{2})z_1^2, \\
p_5 &= -\alpha_2 z_1^4 + j\alpha_1 z_1^3 + \alpha_0 z_1^2 - j\alpha_{-1} z_1 - \alpha_{-2} - (1-3\sqrt{2})z_1^2, \\
p_6 &= \alpha_2 z_1^4 - \alpha_1 z_1^3 + \alpha_0 z_1^2 - \alpha_{-1} z_1 + \alpha_{-2} - (1-\sqrt{2})z_1^2, \\
p_7 &= -\alpha_2 z_1^4 - j\alpha_1 z_1^3 + \alpha_0 z_1^2 + j\alpha_{-1} z_1 - \alpha_{-2} - (1+3\sqrt{2})z_1^2,
\end{aligned} \tag{2.11}$$

where the complex variable $z_1 = e^{j\frac{2\pi t_1}{4}}$ represents the displacement. By Buchberger's algorithm, we obtain a Groebner basis.

$$\begin{aligned}
g_0 &= 2\alpha_2 - 3j\sqrt{2}z_1 + 3\sqrt{2}z_1 - 12\\
g_1 &= \alpha_1 - 2 - j,\\
g_2 &= \alpha_0 - 1,\\
g_3 &= \alpha_{-1} - 2 + j,\\
g_4 &= 2\alpha_{-2} + 3j\sqrt{2}z_1 - 3\sqrt{2}z_1,\\
g_5 &= 2z_1^2 - \sqrt{2}(1+j)z_1 = 2z_1(z_1 - \frac{\sqrt{2}}{2}(1+j)).
\end{aligned} \tag{2.12}$$

Assuming the ordering $\alpha_2 > \alpha_1 > \cdots > \alpha_{-2} > z_1$, we obtain:

$$\begin{aligned}
2\alpha_{-2} &= 3\sqrt{2}z_1 - 3j\sqrt{2}z_1\\
&= 3\sqrt{2}\left(\frac{\sqrt{2}}{2}\right)(1+j) - 3j\sqrt{2}\left(\frac{\sqrt{2}}{2}\right)(1+j)\\
&= 3 + 3j - 3j + 3 = 6
\end{aligned}$$

$$\alpha_{-2} = 3$$

$$\begin{aligned}
2\alpha_2 &= 3j\sqrt{2}z_1 - 3\sqrt{2}z_1 + 12\\
&= 3j\sqrt{2}\left(\frac{\sqrt{2}}{2}\right)(1+j) - 3\sqrt{2}\left(\frac{\sqrt{2}}{2}\right)(1+j) + 12\\
&= 3j - 3 - 3 - 3j + 12
\end{aligned}$$

$$\alpha_2 = 3.$$

In fact, $g_5 = 0$ gives us $z_1 = 0$ or $z_1 = \frac{\sqrt{2}}{2}(1+j)$. And we can see that $z_1 = 0$ does not belong to the unit circle. So, it is not a solution. We can compute easily the others coefficients:

$$\begin{aligned}
\alpha_{-1} &= 2 - j\\
\alpha_0 &= 1\\
\alpha_1 &= 2 + j.
\end{aligned}$$

In the above examples we have taken known signals (exact data). Next we applied the algorithm and we saw that we could reconstruct our signal. Let us examine the other case when the measurement is noisy.

2.2 Multichannel sampling under noisy conditions

It is inevitable that measured signals are contaminated with "noise" when a data acquisition system is used for an experimental measurement. For example, when we call someone and we are in a crowded or windy place. Then the transmitter receives the signals of our voice mixed with other signals due to other people around or the wind. The communication is often unclear because our data is noisy. In our case, such a situation does not allow us to apply directly Groebner bases from the system (1.16). The computation of Groebner bases is typically performed with infinite precision. For the reason that a Groebner basis is defined as a set of polynomials that generates the same affine variety as the original set of polynomial equations. Therefore, the solution that is computed using Groebner bases is an exact solution to the set of polynomial equations.

Moreover, concepts such as projections or distance do not have any meaning over the ring of polynomials. Hence, if the measurements are noisy, Buchberger's algorithm would generally conclude that there is no solution. As there are usually more equations than unknowns in our examples, the errors on the sample values make the equations from (1.16) inconsistent.

To solve our problem, we propose another algorithm as in [Van06]. In fact, we divide the complete set of polynomial equations into multiple (overlapping) critical subsets. As in [SVV06], we mean by critical that there is a finite, non empty set of solutions, typically when the number of unknowns is equal to the number of equations. So if the computational time allows us we could use all the critical subsets that can be derived from the original set of equations. If not we can just use a limited number of them. Then we search a Groebner basis for each subset, and obtain a set of parameter values using back substitution. The final solution can then be defined as an average of the different solutions from the subsets.

Algorithm 3. Algorithm for multichannel sampling from noisy samples [Van06]

1. Write out the equations from (1.16) describing the samples as a function of the signal coefficients.
2. If necessary, perform a change of variables to convert the equations into a set of polynomial equations.
3. Divide these equations into at most $\binom{MN}{L+M-1}$ critical subsets of equations S_i.
4. Compute a Groebner basis for each ideal $\langle S_i \rangle$.
5. Use back substitution to obtain the offsets and the signal parameters.
6. Eliminate solutions that are not valid (e.g. offset values not on the unit circle in the Fourier case).
7. Compute the average of the offsets corresponding to the remaining solutions (typically one per set S_i).
8. Fill in the offsets in the equations from (1.16) and solve the set of linear equations for the unknown signal parameters.

The application of the algorithm3 is illustrated below on the problem of Fourier series with noisy measurements.

Example 2.3. Fourier series with noisy measurements:
Consider a signal that is represented by its $L = 5$ Fourier series coefficients, given by:

$$\alpha = (5 - j, -3j, -6, 3j, 5 + j)^T. \tag{2.13}$$

The signal is sampled with two sets of four samples ($M = 2, N = 4$), with an offset vector $t = (0, \frac{24}{11})$, then $t \simeq (0, 2.1818)$. In a noiseless case, this would result in the following two sets of samples:

$$\begin{aligned} y_0 &= (4, -22, 4, -10)^T \\ y_1 &= (3.0217, -7.5743, -0.3591, -19.0882)^T. \end{aligned} \tag{2.14}$$

The second set of samples is given numerically, because the exact expressions are quite complicated. Now we add white Gaussian noise to the set of samples with mean zero and standard deviation 1, resulting in the noisy sample values

$$\begin{aligned} y_0 &= (3.4845, -21.2468, 3.6672, -9.5310)^T, \\ y_1 &= (2.0917, -7.4480, 0.7300, -19.3078)^T. \end{aligned} \tag{2.15}$$

We obtain a similar set of polynomials as in (2.11), with just different sample values.That is to say:

$$\begin{aligned} p_0 &= \alpha_2 + \alpha_1 + \alpha_0 + \alpha_{-1} + \alpha_{-2} - 3.4845, \\ p_1 &= -\alpha_2 + j\alpha_1 + \alpha_0 - j\alpha_{-1} - \alpha_{-2} + 21.2468, \\ p_2 &= \alpha_2 - \alpha_1 + \alpha_0 - \alpha_{-1} + \alpha_{-2} - 3.6672, \\ p_3 &= -\alpha_2 - j\alpha_1 + \alpha_0 + j\alpha_{-1} - \alpha_{-2} + 9.5310, \\ p_4 &= \alpha_2 z_1^4 + \alpha_1 z_1^3 + \alpha_0 z_1^2 + \alpha_{-1} z_1 + \alpha_{-2} - 2.0917 z_1^2, \\ p_5 &= -\alpha_2 z_1^4 + j\alpha_1 z_1^3 + \alpha_0 z_1^2 - j\alpha_{-1} z_1 - \alpha_{-2} + 7.4480 z_1^2, \\ p_6 &= \alpha_2 z_1^4 - \alpha_1 z_1^3 + \alpha_0 z_1^2 - \alpha_{-1} z_1 + \alpha_{-2} - 0.7300 z_1^2, \\ p_7 &= -\alpha_2 z_1^4 - j\alpha_1 z_1^3 + \alpha_0 z_1^2 + j\alpha_{-1} z_1 - \alpha_{-2} + 19.3078 z_1^2. \end{aligned} \tag{2.16}$$

As we have 8 equations in 6 unknowns (5 signal parameters and an offset), we compute a Groebner bases for all $\binom{8}{6} = 28$ subsets S_i of 6 polynomials from the total set $(p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7)$. The one for the ideal $\langle p_0, p_1, p_2, p_3, p_4, p_5 \rangle$ is given here:

$$\begin{aligned} g_0 &= \alpha_2 - (11.5043 + 8.2663j) z_1 - (8.5363 + 13.258249j) z_1^2 - 9.4824, \\ g_1 &= 0.04567 - 2.9289j + \alpha_1, \\ g_2 &= 5.9065 + \alpha_0, \\ g_3 &= 0.04567 + 2.9289j + \alpha_{-1}, \\ g_4 &= \alpha_{-2} + (11.5043 + 8.2663j) z_1 + (8.5363 + 13.2582j) z_1^2, \\ g_5 &= z_1^3 + (1.1192 + 1.0848j) z_1^2 + (0.0312 + 0.9995j) z_1. \end{aligned} \tag{2.17}$$

As in [SVV06], after computing all the possible solutions for each Groebner bases, we eliminate the invalid ones. Then, from the remaining solutions we compute the offsets t_1, and compute their average value:

$$t_{1,avg} = 2.0660. \tag{2.18}$$

This way of proceeding decreases our computational cost. Because if we just compute the average value of z_1, not only do we waste our time, but we also take in account some solutions which do not belong to the unit circle. This leads to an invalid offset. Note that we performed a simple averaging operation here. We replace this average offset value in the original equations (2.16), and compute the Least Squares solution of this set of linear equations in the unknown signal parameters α. Least Squares is a powerful technique to solve for unknown values, given a lot of data. Least Squares is usually used by scientists or engineers when they have a lot of data (or many observations) and they want to approximate the data with a formula. In our case, we search for the best set of coefficients which approximates our system and minimizes the error:

$$\hat{\alpha} = \begin{pmatrix} 4.7412 - 4.5812j \\ -0.0388 - 2.9566j \\ -5.9450 \\ -0.0388 + 2.9566j \\ 4.7412 + 4.5812j \end{pmatrix}. \tag{2.19}$$

Remark In order to see how accurate our result is, we have to compute the relative error which is given by:

$$\parallel \alpha - \hat{\alpha} \parallel / \parallel \alpha \parallel .$$

For this simulation, we obtain a relative error of 0.493.

This error can be compared to the error that would be obtained from the noisy samples with the exact offset t_1, which is 0.080. Averaged over 250 such simulations with random signal coefficients and offsets, the estimated relative error is 0.340, compared to 0.095 in the ideal case using the exact offsets with the noisy samples, see [SVV06].

3. Complexity and Optimisation

The last part of our work will be focused on the computational cost and the optimisation of the Buchberger's algorithm.

3.1 Computational cost

Using Groebner bases, we have a nice method to solve our multichannel sampling with unknown offsets. We have seen that our problem becomes easily expressible and algorithmically solvable by Buchberger's algorithm. One significant factor when designing algorithms is the algorithm's performance. The effeciency of an algorithm is determined by the amount of time it takes to run the program and the amount of memory space the program requires. Such study is known as complexity theory. The two main measures of complexity are time complexity and space complexity. Time complexity refers to the number of operations performed for a given input; space complexity refers to the space used on a given input.

Recall that we have the following upper degree bound for polynomials in a Groebner basis with respect to any monomial order as you can see in [BW98].

Lemma 3.1. *[Dub90] Let $\mathbf{K}[X_0, \cdots, X_{n-1}]$ be a ring of multivariate polynomials with coefficients in a field $\mathbf{K}$, and let $\mathcal{F}$ be a subset of this ring such that d is the maximum total degree of any polynomial in $\mathcal{F}$. Then for any monomial order, the total degree of polynomials in a Groebner basis for the ideal generated by $\mathcal{F}$ is bounded by*

$$(d^2 + 2d)^{2^{n-1}}. \tag{3.1}$$

3.1.1 Time complexity of Buchberger's algorithm

In the case of Groebner bases, unfortunately, our algorithm presents a high complexity. It is given by the fact that:

- in the back substitution step of our solution method, we need to compute the zeros of a polynomial. The complexity of this operation will depend on the order of the specific polynomial that is obtained when we compute Groebner bases. Although theoretically this order can only be bounded by (3.1), in practice, it is often much lower(as we have seen from our examples). And the roots of a polynomial with degree $\mathcal{E}$ can be computed using an algorithm with complexity

  $$\left(\mathcal{E}\,(log\mathcal{E})^2 \mid log\epsilon \mid +\mathcal{E}^2\right) \tag{3.2}$$

  if $\mathcal{E}$ is the degree of the polynomial and ϵ is the precision of the computed roots [SVV06]. The upper bound on $\mathcal{E}$ is given by (3.1).

- by adding the non-zeros remainders S- polynomials, we perform many operations and this increases the time cost of our algorithm.

3.1.2 High memory of Buchberger' s algorithm

- As explained in chapter 1, the set of polynomials p_i has to be expanded in the first part of the algorithm by adding the non-zero remainders of S-polynomials.

  This expansion can become very large, and is one of the reasons for the high-memory (storage) requirements of Buchberger's algorithm.

- Another reason for the high complexity of our algorithm also is given by the fact that it performs computations with infinite precision.

  Example 3.1. : Using Maple [SVV06] to solve a polynomial problem like the one in example 2.2, a 6-th degree polynomial, and 2 sets of 4 samples, the algorithm already requires more than 1 GB of memory which is probably double the amount of an AIMS computer machines .

How difficult is it to construct Groebner bases with our algorithm? In one way it is very easy because the operations needed are elementary. In another way, the intrinsic complexity of the Buchberger algorithm is shown to be exponential [Dub90] due to the expansion of the number of non-zeros remainders S-polynomials. Fortunately, the structure of our problem makes it easier. Our case is specific because the polynomials are linear in the signal parameters and non linear in the offsets. In addition, the upper bound (3.1) is much worst than the average case. In practice, the complexity is much lower. Presently works are in progress in order to optimize the Buchberger's algorithm such as the F4 algorithm [Ham05], F5 algorithm [Fau02] and improvement using syzygies modules [JS06].

3.2 Optimization

Optimization has always been considered in terms of deriving algorithms with a guaranteed behaviour, where such a guarantee refers both to the accuracy of the returned solution (in terms of worst case versus average case) and to the computational cost (polynomial in most cases).

3.2.1 For Fourier series

For the first set of N equations ($t_0 = 0$), we have $z_0 = e^{j2\pi t_0/N}$, which always gives $z_0 = 1$. As a consequence, the signal parameters can be eliminated by multiplying with certain complex numbers and adding together the equations. We never need to multiply any of the equations from the second set corresponding to the offset variable t_1. Therefore, we do not increase t_1 degree.

3.2.2 For Polynomial signals

We can perform a similar elimination by ordering the signal parameters as $\alpha_{L-1}, \alpha_{L-2}, \ldots, \alpha_0$. We can eliminate each of the parameters without needing to multiply equations by the offset variable t_1. Unfortunately, if more than $M = 2$ sets are considered, the different offsets have to be multiplied in the Gaussian elimination, and the results are more complex.

Using Gaussian elimination, we can replace the signal parameters by their expression as a function of t in the $MN-L$ remaining equations. Then, instead of solving MN equations with $L+M-1$ unknowns we have $MN-L$ equations with $M-1$ unknowns. It is now sufficient to compute a Groebner basis for this smaller set in much fewer unknowns ($M << L$) with noisy samples. We can now compute a Groebner basis for the $\binom{MN-L}{M-1}$ subsets of $M-1$ equations instead of the $\binom{MN}{L+M-1}$ sets of $L+M-1$ equations previously.

The maximum total degree of a Groebner basis for a such subset is reduced to:

$$2\left(\frac{L^2}{2}+L\right)^{2^{M-2}}. \tag{3.3}$$

Once this smaller Groebner basis is computed, the offset values can be obtained using back substitution and a method to compute the zeros of a polynomial. We can compute the signal parameters by substituting the offset values in the first L equations. Once the offsets are known, other methods (such as Least Squares) can also be used to compute the signal parameters from the original equations. The algorithm is given below.

Algorithm 4. Algorithm for multichannel sampling from noisy samples using Gaussian elimination for the linear part.

1. Write out the equations from (1.16) describing the samples as a function of the signal coefficients.
2. If necessary, perform a change of variables to convert the equations into a set of poynomial equations. (These are linear in the signal coefficients α, which are functions of the offsets t.)
3. Apply the Gaussian elimination on the first L equations to compute the signal coefficients α as a function of the offsets t.
4. Replace the values of the signal coefficient α in the remaining $MN-L$ equations and multiply each equation by their common denominator to obtain a set of $MN-L$ polynomial equations in the offsets t.
5. Divide these equations into at most $\binom{MN-L}{M-1}$ critical subsets of these equations S_i.
6. Compute a Groebner bases for each set S_i.
7. Calculate the possible offset values using back substitution and by computing the zeros of polynomial equations.

8. Eliminate offset values that do not give a valid solution (values not on the unit circle in the Fourier case) .

9. Compute the average of the offsets corresponding to the remaining solutions (normally one per set S_i).

10. Replace this value in the original equations and solve for the signal parameters α.

Now let us illustrate this algorithm.

Example 3.2. (Fourier series using Gaussian elimination)
We use the same signal and sample values as in example 2.3. Instead of calculating a Groebner basis for the 28 subsets of 6 equations, we now eliminate the signal parameters first from the first $L = 5$ polynomials using Gaussian elimination.

This gives us the signal parameters as a function of the offset:

$$\begin{aligned}
\alpha_{-2} &= 9.4824 + \frac{9.4824 - (0.0457 + 2.9289j)z_1 - 7.9982z_1^2 - (0.0457 - 2.9289j)z_1^3}{-1 + z_1^4},\\
\alpha_{-1} &= -0.0457 - 2.9289j,\\
\alpha_0 &= -5.90654,\\
\alpha_1 &= -0.0457 + 2.9289j,\\
\alpha_2 &= \frac{-9.48237 + (0.0457 + 2.9289j)z_1 + 7.9982z_1^2 + (0.0457 - 2.9289j)z_1^3}{-1 + z_1^4}.
\end{aligned} \tag{3.4}$$

Where we assume that $z_1 \neq 1$. We can then replace these values in the remaining equations, and multiply them by their common denominators. This results in three polynomial equations in the unknown offset z_1:

$$\begin{aligned}
&(2.9746 + 2.8833j)z_1 + 6.4568z_1^2 + (2.9746 - 2.8833j)z_1^3\\
&-(2.9746 + 2.8833j)z_1^5 - 6.4568z_1^6 - (2.9746 - 2.8833j)z_1^7 = 0,\\
&(-0.0913 - 5.8579j)z_1 - 1.3617z_1^2 - (0.0913 - 5.8579j)z_1^3\\
&+(0.0913 + 5.8579j)z_1^5 + 1.3617z_1^6 + (0.0913 - 5.8579j)z_1^7 = 0,\\
&(-2.8833 + 2.9746j)z_1 - 5.4031z_1^2 - (2.8833 + 2.9746j)z_1^3\\
&+(2.8833 - 2.9746j)z_1^5 + 5.4031z_1^6 + (2.8833 + 2.9746j)z_1^7 = 0
\end{aligned} \tag{3.5}$$

We do not need to compute the Groebner basis for the ideal generated by these three equations in order to get the unknown z_1. Then we can directly compute the zeros for each of the polynomials separately. After elimination of the zeros that are not valid solutions, we have the following zeros remaining for the three polynomials:

$$\begin{aligned}
z_1^{(1)} &= -0.9957 - 0.0924j,\\
z_1^{(2)} &= -0.9949 - 0.1007j,\\
z_1^{(3)} &= -0.9982 - 0.0594j,
\end{aligned} \tag{3.6}$$

From these values, we get the offsets $t_1^{(1)} = 2.0589$, $t_1^{(2)} = 2.0642$, and $t_1^{(3)} = 2.0378$. We take the average of these solutions ($t_{1,avg} = 2.0537$), replace the corresponding value of z_1 in the original equations and compute a least squares solution to these linear equations. We obtain the coefficient vector

$$\hat{\alpha} = \begin{pmatrix} 4.7412 - 5.7629j \\ -0.0457 - 2.9289j \\ -5.9065 \\ -0.0457 + 2.9289j \\ 4.7412 + 5.7629j \end{pmatrix}. \qquad (3.7)$$

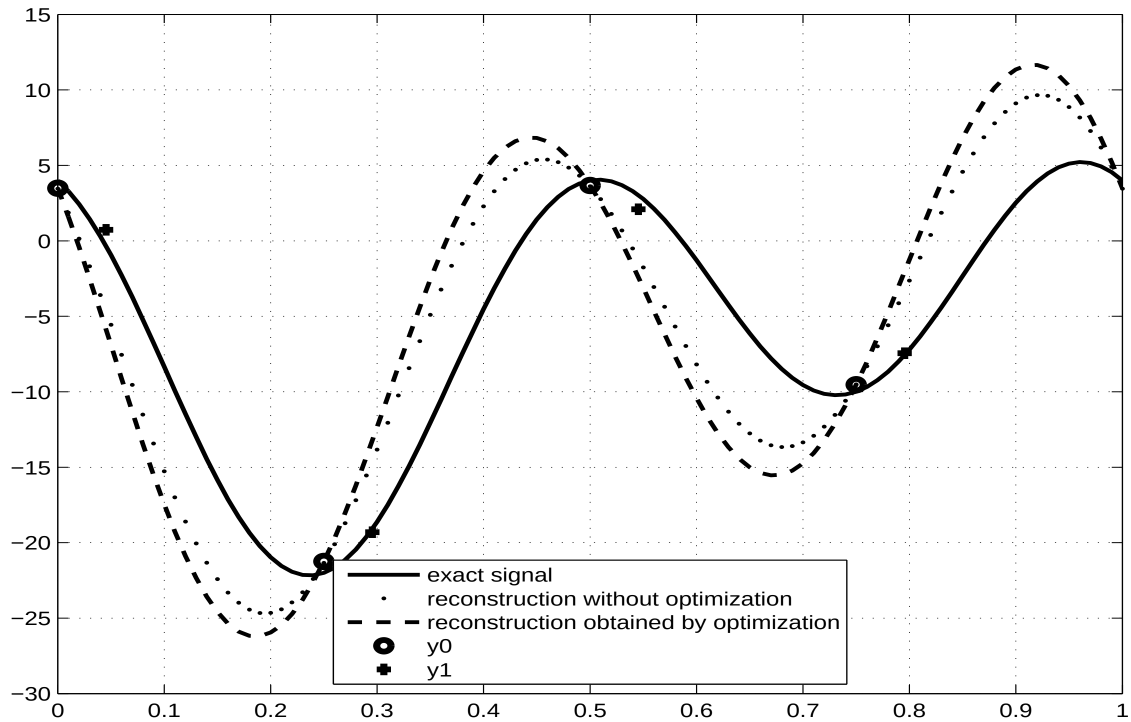

Figure 3.2: The signal with Fourier series coefficients $\alpha = (5 - j, -3j, -6, 3j, 5 + j)^T$ is sampled with two sets of our noisy samples $y_0 = (3.4845, -21.2468, 3.6672, -9.5310)^T$ and $y_1 = (2.0917, -7.4480, 0.7300, -19.3078)^T$ with offset $t_1 = 24/11$. Its reconstruction both in Example 2.3 and Example 3.2 are shown.

Remark Note that with the optimized algorithm, the precision with which the parameters are computed is slightly lower. In [SVV06] the relative error for our estimation is

$$\| \alpha - \hat{\alpha} \| / \| \alpha \| = 0.655.$$

We can compare this result to the error that would be obtained by applying a Least Squares estimation on the noisy samples with the exact offset t_1, which is 0.080. By taking also the average over 250 simulations with random coefficients and offsets, the estimated error is 0.618, compared to 0.095 with the exact offsets. However our estimation has a larger error than the result that we get in the example 2.3, it is important to note that we significantly reduce our computational cost. Instead of computing 28 Groebner basis for sets of 8 equations, only 3 sets of a single equation remain. Taking this into account we can solve directly our equation for this example.

Various optimizations of Buchberger's algorithm also exist. Often, other orderings than the lexicographic ordering result in lower complexity. In addition, algorithms exist to convert a Groebner basis using one ordering, into a Groebner basis for another ordering, and this is known as Groebner walk [CLO04].

3.3 Computer Algebra software

Various implementations of Buchberger's algorithm exist: Singular, Macaulay2, Maple, Magma, and Mathematica. The first two are released under Free Software licences.

For the result obtained in this paper, all the simulations are available online at
`http://lcavwww.ep.ch/reproducible_research/SbaizVV06`.

Conclusion

We presented a method to reconstruct a signal from the multiple sets of samples. The problem is first rewritten as a system of polynomial equations. Next, we compute a Groebner basis for the corresponding ideal. The signal parameters can then be easily derived from this Groebner basis. In this way, the non linear problem in the joint unknown signal coefficients and offsets is highly simplified.

We illustrated our method with examples for the reconstructions of the polynomials and bandlimited signals. Then, we present an adaptation of our algorithm to the case of noisy measurements where Groebner bases are computed for critical subsets of polynomials. Finally, some complexity issues were discussed, and a more efficient method was presented that computes the linear signal parameters first, so that Groebner bases have to be computed only for a significantly smaller set of ideals, generated by equations in the unknown offsets.

Unfortunately, the high complexity of Buchberger's algorithm in the worst case does not make our method feasible in all cases. However, it is important to note that many research works and theses are based on Buchberger's algorithm. So we can see that the field is still under active development, both into the direction of improving the method by new theoretical insights and by finding new applications.

The Buchberger algorithm is very important in the way that without such a Groebner basis we can not even talk about polynomial division if you have more than one variable. It is very helpful for us for example in solving the problem of ideal membership. Another reason to study Groebner basis theory is that it is an important technique which gives exact solutions of non linear problems.

In signal processing even if we do not comply with the Shannon Nyquist Sampling theorem we can reconstruct the signal only with the method presented in this paper.

Acknowledgment

Firstly I am grateful to God for giving me the force and courage to finish this work. Secondly, I would like to thank my supervisor Barry Green for his guidance and assistance in this project. Then, I am also thankful to my tutor Paul Razafimandimby who has assisted me during my essay. Next my family and all my close friends who have supported me. Specially thanks to Patrick Vandewalle who gives me useful documents and advice when I have worked on my essay. And finally my thanks to all the people at AIMS who have helped me in my time here. Especially Igsaan Kamalie, Jan Groenewald and the director of Aims Fritz Hahne who gave me support during those last hard days.

Misaotra betsaka ry mama fa teo anilako foana ianao tamin' ny tolona rehetra natrehako teo amin'ny fiainako. Misaotra ry zoky malala fa tsy irery ahy raha nisy ny ady sarotra fa nifankahery hatrany isika. Misaotra an' ilay belokely malalako izay tena andry niankinana tokoa. Ary ho anao ry dada, tsy mba ho faty velively ato am-poko ny fitiavanao. Ny anatrao hotehiriziko ato amiko mba tsy ho very maina ny herim-ponao sy mama niezaka hanabe ahy ho olom-banona andry sy reharehany ny fianakaviana.
Tiako ianao ry dada.

Bibliography

[BW98] B. Buchberger and F. Winkler, *Grobner Bases and Applications*, Cambridge University Press, 1998.

[CIDoE] The Collaborative International Dictionary of English, *http://www.dico.org*.

[CLO96] D. A. Cox, J. Little, and D. O'Shea, *Ideals, varieties and algorithms*, Springer, 1996.

[CLO04] ______, *Using Algebraic Geometry*, Springer, 2004.

[Dub90] Thomas W Dube, *The structure of polynomial ideals and groebner bases*, SIAM J. Comput (1990), Submitted for publication.

[Fau02] Jean-Charles Faugere, *Efficient algorithm for computing groebner bases without reduction to zero(f_5), proceedings of the 2002 international symposium on symbolic and algebraic computation, ISSAC,*, ACM Press, Italy, 2002.

[GOT] Glossary Of Terms, *http://www.eas.slu.edu/ggp/ggpgloss.html*.

[Ham05] B. Hammersholt, *The F4 algorthm for computing Groebner Bases*.

[JS06] Winfried Just and Brandilyn Stigler, *Computing groebner bases of ideals of few points in high dimension*, ACM Communication in Computer Algebra **40** (2006).

[SVV06] L. Sbaiz, P. Vandewalle, and M. Vetterli, *Groebner Basis Methods for Multichannel Sampling with Unknown Offsets*, Appl. and Comp. Harm. Analysis (2006), Submitted for publication.

[Van06] P. Vandewalle, *Super-Resolution from Unregistered Aliased Images*, Ph.D. thesis, Ecole Polytechnique Federale de Lausanne, Switzerland, 2006.

[VSVV06] P. Vandewalle, L. Sbaiz, J. Vandewalle, and M. Vetterli, *Super-resolution from unregistered and totally aliased signals using subspace methods*, IEEE Transactions on Signal Processing (2006), Draft.

[WD] Wordnet Dictionary, *http://www.word.org*.

[Wei07] E. Weisstein, *NyquistShannon Sampling Theorem — Wolfram MathWorld, the web's most extensive mathematics resource*, 2007.