NTNU

Norwegian University of
Science and Technology

High resolution spectral estimation for radar in a pipe

Henning Schei

Master of Science in Electronics
Submission date: July 2017
Supervisor: Torbjørn Ekman, IES

Norwegian University of Science and Technology
Department of Electronic Systems

NTNU – Trondheim
Norwegian University of
Science and Technology

KONGSBERG

High Resolution Spectral Estimation For Radar In Pipe

Henning Schei

June 2017

MASTER THESIS
Department of Electronic Systems
Norwegian University of Science and Technology

Supervisor: Prof. Torbjørn Ekman, Prof. Egil Eide
External supervisor: Stein Arne Askeland, Kongsberg Maritime

Summary and Conclusions

In this project I have studied and evaluated several high resolution spectral estimation algorithms to improve accuracy and robustness in a FMCW radar tank gauge (RTG). The system parameters are taken from KONGSBERG's GLA-300 LNG-radar. I have in particular investigated how subspace based spectral estimation algorithms can improve liquid level estimation in the situation when the liquid surface is close to a reflector. A radar tracking algorithm have also been implemented to increase accuracy of measurements and follow the time variability of the measurements.

The project includes data from simulation of a waveguide, as well as experiments using physical measurements from the actual RTG. All signal processing have been implemented in MATLAB. Implementations of the spectral estimation algorithms are partly taken from [8] and partly developed self.

The simulation results show that one can not achieve measurement accuracy that satisfies the accuracy criterion of $\pm$5mm of maximal estimation error with the proposed signal processing, but the accuracy in measurements are approximately doubled compared to the existing setup. Using a radar tracking in addition, the accuracy satisfies the accuracy criterion.

The liquid level estimation using physical measurements confirms the results from simulation and a more stable liquid level estimation is achieved. Compared to the existing method, the high resolution methods are to a lesser extent affected by resolution problems close to reflectors.

Chapter 1

Acronyms

FMCW Frequency modulated continuous wave

LNG Liquefied natural gas

RTG Radar tank gauge

SVD Singular value decomposition

SNR Signal to noise ratio

Ullage Unfilled space in a tank

Contents

Chapter 2

Introduction

The main focus in this project is to investigate the improvement of radar tank level gauge using high resolution spectral estimation algorithms such as ESPRIT and Root-MUSIC in lieu of an existing Fourier-transform based method. The radar system is using a frequency modulated continuous wave (FMCW) signal and a Fourier-transform based spectral estimation technique for detection. The range resolution of is limited by the bandwidth of the transmitted wave [3]. These spectral estimation methods concerned in this project are so called *high-resolution* (also called subspace-based, eigenanalysis-based or super-resolution) techniques. This is due to their ability to resolve spectral lines in frequency $f = \frac{\omega}{2\pi}$ by less than $\frac{1}{N}$ cycles per sampling interval, which is the resolution limit for classical non-parametric methods. [4] [8]

The *a priori* assumption is that the signal model of the reflection in the tank can be approximately described as a sum of real sinusoids. Non-parametric methods maximize the power spectral density with no information of signal content, whereas the high resolution methods tries to fit the data the some kind of assumed sinusodal model. The performance of the high resolution algorithms will in turn not only be a question about the algorithm itself, but how well the signal fit in the signal model.

2.1 Background and Motivation

KONGSBERG Maritime, Lade, develops and produces sensors for ship cargo and engine room monitoring. One of their high-end products and the concern in this thesis is the GLA-310 Radar Tank Gauge for LNG-fuel tank. The RTG employs a Frequency Modulated Continuous Wave (FMCW) principle with a frequency sweeping signal that is emitted by a radar through a standing pipe acting as a circular waveguide. The radar unit is mounted on a socket on top of the tank as well as a pressure sensor. The tank has an over pressure system for keeping oxygen coming in in the tank, as well as it is filled with a high concentration of nitrogen.
The signal speed in the waveguide is dependent of several factors:

- The amount of LNG vapor inside the pipe

- Temperature
- Pressure

In order to measure the signal speed in the pipe, there are six teflon reflectors with exact known locations. (the terms reflectors and disc are used interchangeably in this text). Measurements from these reflectors are used to calibrate for the change in signal speed.

This radar approach for tank level gauging permits high accuracy and reliability measurements of tank level as well as resistance to radar clutter. However, due to the high economically value of the cargo, there is an extreme demand for accuracy. Additionally, the current implementation has robustness issues in some scenarios, especially when the distance between a reflector and the liquid surface is small.

Problem Formulation

When the liquid surface is close to a reflector in the standing pipe, it is not possible to distinguish the reflector echo from the liquid echo by standard signal processing. The goal of this thesis is to test and evaluate other high resolution spectral estimation algorithms such as MUSIC (MUltiple SIgnal classification), ESPRIT or others to obtain a higher spectral resolution and improve the accuracy of measurements when the liquid surface is close to a reflector. The goal is to obtain an accuracy of 5 mm in measurements in ranges up to 50 m.
In addition to high resolution spectral estimation, a suitable radar tracking algorithm will be implemented to follow the time variability of the surface due to among other things waves and vibration.

We can provide both simulated and measured data for evaluation of the algorithms.

2.2 Objectives

The main objectives of this project are:

1. Study relevant theory and develop methods and signal processing algorithms that are fit for processing a signal where interference between close separated echos make conventional spectral estimation algorithms unsuitable. In particular where the liquid surface and a reflector are close. A particular emphasis are put on Root-MUSIC, ESPRIT and how they perform against existing routines.

2. Develop a simulation environment for evaluation of Root-MUSIC, ESPRIT and other methods. The reflection signal model should be as close as possible approximation to the real physical case. In the model, it should be easy to move the liquid surface in the pipe and see the effects of measurements when liquid surface and reflector approaches each other.

3. Quantify the magnitude of the estimation error in the simulation environment and compare the existing algorithm to the proposed high resolution algorithms.

4. Evaluate the algorithms on real physical data from the actual RTG. Evaluate, if any, improvements on accuracy, robustness, computational complexity or memory usage.

5. Develop a Kalman filter based radar tracking algorithm to follow the time variability and minimize estimation error.

2.3 Limitations

This project includes simulated and measured data. It is necessary to divide the limitations of the project into limitations in the simulation environment and limitations regarding real measured data. The following table describes the limitations of the simulation environment.

- The reflection associated with the bottom of the tank, as well as any possible reflection outside the pipe is neglected.
- The noise in these measurements are assumed to be zero mean additive Gaussian noise with varying variance.
- Assume that the pipe sections, LNG-sections and reflectors are homogeneous.

Limitations using real measured data

- The physical data is not a complete filling of the tank, but a measurements of about 3 hours of constant filling up to the lowest reflector in the pipe.
- Measurements are sampled at uneven time instants. This could lead the radar tracking algorithm to have poorer performance than if it had an uniform sampling period.

Chapter 3

System Description

3.1 The Radar

The theory behind a FMCW radar is well known, but it is useful to include a summary. The following is a mathematical description of the radar system. The modulation pattern is a stepped staircase modulated signal with the advantage of of expanding unambiguous range. Pulse radars detect the target by emitting a short pulse and observing the time of flight of the target echo. This requires the radar to have high instantaneous transmit power and may need a large antenna apperature. In contrast, FMCW radars achieve can obtain similar results using a much smaller transmit power and physical size. The FMCW radar continuously emitting periodic pulses whose frequency varies with time. The range to the target is calculated by frequency difference between the received and emitted radar signals. The range to the target is a function of this frequency difference, which in the latter will be referred to as the beat frequency.

As shown in figure 3.1, a microwave signal with a frequency f_0 is transmitted every T_s period, and in the next period stepped to a frequency f_1
Frequency step number i:

$$f_i = f_0 + i\Delta f. \tag{3.1}$$

The emitted signal on frequency step number i, we have

$$S_{t,i} = e^{j2\pi f_i t} \tag{3.2}$$

Where t is the local time within a frequency step.
From a target with distance R the time delay is $\tau = \frac{2R}{c}$, and the received signal can be expressed as:

$$S_{r,i} = e^{j2\pi f_i (t-\tau))} \tag{3.3}$$

For the downmixed signal, we have from 3.2 and 3.3

$$S_i = S_{r,i} S_{t,i}^* = e^{j2\pi f_i (t-\tau)} e^{-j2\pi f_i t} = e^{-j2\pi f_i \tau} \tag{3.4}$$

The phase for the downmixed signal:

$$\phi_i = -2\pi f_i \tau = -2\pi (f_0 + i\Delta f)\tau \tag{3.5}$$

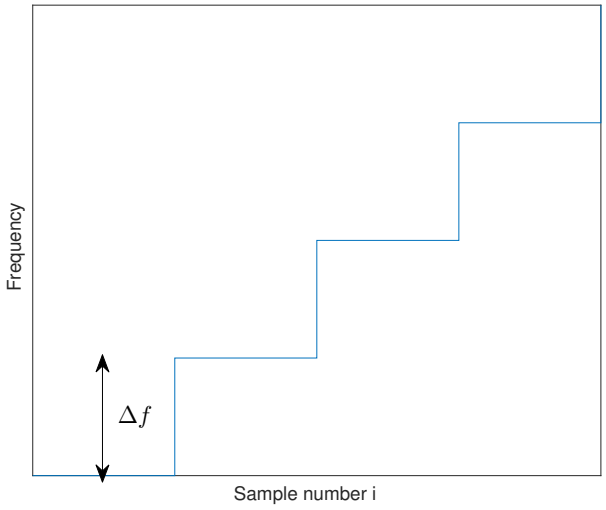

Figure 3.1: Stepped staircase modulated signal

The "frequency" if the downmixed signal, substituted variables from 3.1:

$$f = \frac{\partial \phi}{\partial i} = -2\pi \Delta f \tau = -2\pi \Delta f \frac{2R}{c} \tag{3.6}$$

We can see from 3.6 that the frequency of the received signal is dependent on the frequency step and the distance. And the range resolution is given by

$$RES = \frac{c}{2\Delta f} \tag{3.7}$$

Where c is the speed of light in the medium.

From the estimation of the range to the target is equivalent to the estimation of the beat frequency. The range resolution is then dependent upon the resolution of the frequency estimation algorithm used in determination of the beat frequency. By using a standard FFT-based method, we can obtain a frequency resolution of f_s/N [8].

3.2 Reflection model

The generated data are based on the reflection from a circular standing pipe with a free space impedance terminated end. This pipe has the same configuration as the pipe in use today. The pipe is assumed to consist of 5 segments, and 6 reflectors. The figure 3.2 shows an example of how the pipe could look like. The waveguide is filled with gas with relative permitivity (approx. $\epsilon_r = 1$) and the blue sections modeling reflectors are filled with a permitivity equal to the reflector permitivity. Note that the lengths of the reflectors in figure 3.2 are oversized and the number of reflectors are more than shown on the figure.

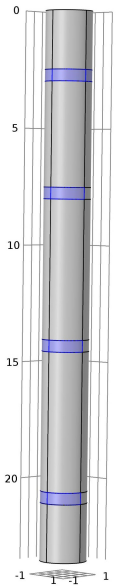

Figure 3.2: Sketch of the pipe used to generate the data. Blue sections are reflectors.

In the reflection model we assume that a plane equiphase electromagnetic wavefront hits the reflectors propagating down the waveguide. A part of the incident effect will be reflected and a part will be transmitted forming a multi-reflection pattern. The final reflected signal, Γ_{in}, will be a sum of all refected signals with a phase shift that corresponds to the distance to the objects.

Defining the generated reflection as $raw(f)$

$$raw(f) = real(\Gamma_{in}) \tag{3.8}$$

3.3 Modeling the waveguide in MATLAB

In this following describes the model to simulate the waveguide in MATLAB. Starting from the waveguide model and associated explanations on FMCW-radar, we can create a transmission line model that simulates the reflection of reflectors and liquid surface.

This MATLAB model is built up of a script and its related functions to simulate the waveguide. It has the same configuration as the 10Ghz radar in use today and with the same distance between reflectors. The waveguide consists of 5 cylindrical sections with 4 discs to simulate reflectors and a section to simulate LNG with appropriate dielectrical constant. The dielectrical constant can be changed to simulate other types of liquefied gases. There is possible to move the liquid surface within the pipe.

The waveguide is modeled in MATLAB using ABCD-matrices. For each section, the ABCD-parameters are calculated on the basis of the current mediums dielectrical constant, impedance, wavenumber and length. So each sub-pipe and reflector are represented by one homogeneous section. The end of the pipe is terminated by LNG liquid impedance. The total cascade of all ABCD-matrices makes up the model of shunts that produce a reflective wave. Using conversation methods from [5] the ABCD-parameters are coverted into S-parameters where the real value constitutes the reflection pattern. The design choice of taking the real value is due to simplicity and because the SNR is not an issue in the application according to figure 3.2.

The usefulness of the approach lies in the relatively pragmatic way of building a complex multiple reflection waveguide structure from simple elementary two-port networks. In appendix the MATLAB code for creating the waveguide is included.

Parameters that are used include:

- Waveguide diameter : 50.16 mm
- Carrier frequency : 10 GHz
- Bandwidth : 1.5 GHz
- Relative dielectric constants for discs : 1.39 (in air), 1.78 (submerged in LNG), values are taken from COMSOL simulation.
- Relative dielectic constant for LNG : 1.6
- Conductivity of the pipe : 1.35 $10^7 \frac{S}{m}$
- Reflectors are located at: : [2.6, 7.6, 13.6, 19.6, 25.6, 30.05] m
- Pipe length: : 31,2m

Figure 3.3 shows a typical reflection pattern for the model using Fourier transform. As the channel is dispersive, there is also a dispersion corrected version of the reflection, using a proprietary dispersion correction method from KONGSBERG. As we can see from the figure, there are 5 echoes from the reflectors at distances $d = [2.6, 7.6, 13.6, 19.6, 25.6]$. The last echo is due to the liquid surface approaching the reflector. The accuracy of reflector position are $\pm 2mm$ using a 5-point center-of-gravity calculation, without noise.

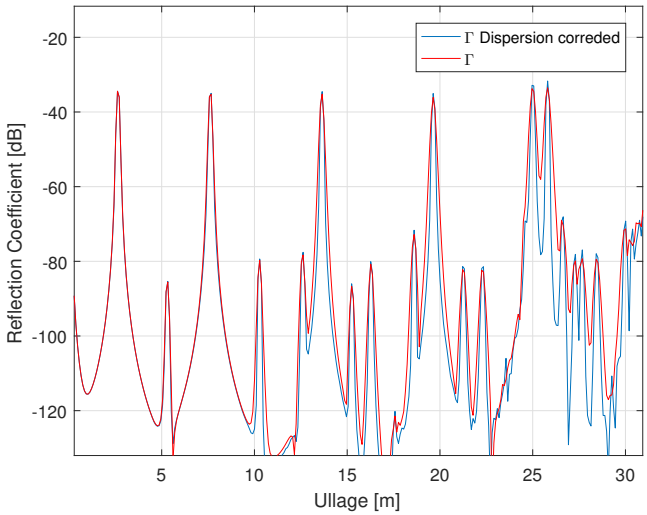

Figure 3.3: An example of how the reflection coefficients modeled with ABCD-matrices and using Fourier transform. Both the dispersion corrected and non-dispersion corrected versions are shown. Note that echo at 30.05m are not shown. Liquid surface close to 25m.

3.4 Radar tracking / Kalman Filtering

The motivation for utilizing a Kalman filter to this problem is the ability to have more control over the measurements in the cases where the spectral resolution is not big enough to separate reflections in the system. In a small region close to the reflectors, the liquid level can not be estimated correctly, and this feature is an attempt to overcome this problem. In addition to minimize the mean-square error of the measurements [1]

The following gives a brief summary of the Kalman filter algorithm in general as well as justification for various design choices made when adopting this to the problem.

3.4.1 Kalman filter algorithm

It is necessary to include a brief summary of the Kalman filter algorithm in order to discuss the design choices in this project. From [1] and [7] we develop the recursive equations. The optimization criterion is minimization of the mean-square estimation error of the input.

Assuming we have an initial estimate of the process at time t_k. The a priori estimate, denoted by $\hat{\boldsymbol{x}}_k$, where "hat" means the estimate and super-script minus means that it is our so far best estimate.

The system is assumed to be linear. When designing the Kalman Filter, the following notations are introduced:

$\boldsymbol{e}_k^- = \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k$ - Prediction error
$\boldsymbol{P}_k = E[\boldsymbol{e}_k^- \boldsymbol{e}_k^{-\ T}]^T = \boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_k^-$ - Prediction error covariance
$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-)$ - State - Estimated liquid level
$\boldsymbol{\phi}_k$ - State transition matrix
$\boldsymbol{K}_k$ - Kalman gain
$\boldsymbol{Q}_k$ - Process variance
$\boldsymbol{R}_k$ - Measurement noise variance
$\boldsymbol{H}$ - Measurement function

Figure 3.4.1 shows a graphical description of the steps adopted from [1].

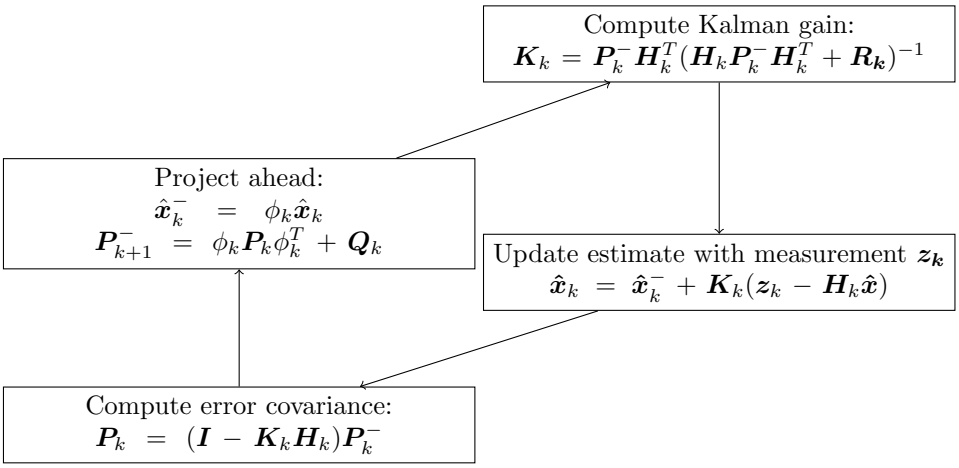

Figure 3.4: Graphical description of the Kalman filter algorithm

3.4.2 Modeling the system

To make this possible, it is necessary to constrain the problem in which we can describe by Newton's equations of motion. We know that the liquid surface can not make suddenly jumps. By Newton's equations of motion, given a constant velocity v, the position of x after time t equals

$$x = vt + x_0 \tag{3.9}$$

Where x_0 is the starting position. In this problem, both the position and velocity of the surface will be tracked.

Remaining design choices is the values for R_k, H_k and Q_k, ϕ_k

$\boldsymbol{R}_k$ was set to the measurement error covariance.
$\boldsymbol{Q}_k$ and $\boldsymbol{\phi}_k$ was computed with the Van Loan Method from [1]
$\boldsymbol{H}_k$ was set to $H = [1\ ,\ 0]$

3.5 Real Measured data

The real measured data are obtained from an experiment where the tank are continuously filled to above the lowest reflector in a time frame of about three hours. The measurement includes more than 7000 measurements of FMCW sweeps and creates a basis on which the algorithms accuracy and robustness where evaluated. There is taken a measurement about every 2nd second at nonuniform intervals. The measurements are time stamped, so there is possible to set up at time vector of the experiment. The measurements are split into several sets that are meant to be consecutive in time, but between two sets there are a small gap in the measurements as shown in the result chapter.

Chapter 4

Estimation of liquid level: spectral estimation techniques

4.1 Introduction

We may summarize from [8] the traditional spectral estimation techniques in two main classes.

1. Non-parametric methods.
2. Parametric methods

The non-parametric methods relie entirely on the Fourier transform or PSD to provide spectral estimates. They provide a reasonably high resolution for sufficiently long data lengths, but are poor spectral estimators because of their high variance and does not decrease with increasing data lengths. The variance of the non-parametric methods have motivated the development of different methods that have lower variance with a trade off of having poorer resolution. Several methods have been introduced, we will not go into further details on this methods because they are not directly relevant to the work done in this project.

The latter approach, the *parametric* approach, is to postulate some a model for the data, in which the data provide, a way to parametrize the spectrum, and in turn *reduce* the problem of spectral estimation into a problem of estimating the parameters in an assumed model.
A sub class of this parametric methods is subspace-based (also called eigenanalysis-based or super-resolution-methods), in which MUSIC, Root-MUSIC and ESPRIT are included.

The main drawback of these methods, compared to other non-parametric methods, is that the performance degrades if the measurements noise can not be assumed to be Gaussian [8].

4.2 High resolution spectral estimation methods: MUSIC, Root-MUSIC and ESPRIT

According to [8], [6], the high resolution methods consider a signal model as a mixture of noisy complex exponentials.

$$x[n] = \sum_{k=1}^{N} c_k \; exp(j2\pi f_k t_k) + w[n] = s[n] + w[n], n = 0 \dots N_s - 1 \tag{4.1}$$

where $x[n]$ are the N_s samples of the observed noisy signal with noise $w[n]$. c_k stands for the amplitude, f_k are the frequencies of the complex exponentials, sampled at time instants $t_n = nT_s$.

Decomposition of eq. 4.1 eigenstructure is the core of this techniques. The aim is to separate the observation space in a signal subspace and a noise subspace. Where as the signal subspace containing only the useful information. To perform this estimation, a singular value decomposition (SVD) of its covariance matrix is used.

Equation 4.1 can be expressed in matrix form

$$\boldsymbol{x} = \sum_{k=1}^{N} c_k \boldsymbol{e_k} + \boldsymbol{w} \tag{4.2}$$

Where

$$\begin{gathered} \boldsymbol{e_k} = [1 \;\; exp(j2\pi T_s f_k) \; ... \; exp(j2\pi (N_s - 1) T_s f_k)]^T \\ \boldsymbol{w} = [w(0) \;\; w(1) \; ... \; w(N_s - 1)]^T \\ \boldsymbol{E} = [\boldsymbol{e_1} \; \boldsymbol{e_2} \; ... \; \boldsymbol{e_N}] \; , \boldsymbol{c} = [c_1, c_2, \; ... \; c_N]^T \end{gathered}$$

As stated above, the first step for implementing the concerned high resolution methods is the eigenanalysis of the observed data samples. The estimated autocorrelation matrix of the signal from eq.4.1 can be expressed as

$$\hat{R}_x = \frac{1}{N_{obs}} = \sum_{i=1}^{N_{obs}} = \boldsymbol{x}_i \; \boldsymbol{x}_i^H \tag{4.3}$$

Where $\hat{R}_x$ is the estimated autocorrelation matrix of the N_{obs} data samples. The R_x matrix must be of full rank in order to properly separate the subspaces and the spectral components.[8]. Spatial smoothing or so called forward-backward approach can be used obtain a full rank property.

The autocorrelation matrix of order $p > N$ is estimated in the following way according to [6]:

$$R_p = D_p D_p^H \tag{4.4}$$

Where the data matrix D_p in 4.4 is defined as follows:

$$D_p = \begin{bmatrix} x(0) & x(1) & \cdots & x(N_e - p) \\ x(1) & x(2) & \cdots & x(N_e - p + 1) \\ \vdots & \vdots & \ddots & \vdots \\ x(p-1) & x(p) & \cdots & x(N_e - 1) \end{bmatrix} \tag{4.5}$$

The columns of the matrix D_p in 4.5 can be seen as the new observation vector and constitutes the observation space with the dimension p.

4.2.1 MUSIC and Root-MUSIC

MUSIC and Root-MUSIC frequency estimation algorithm are one of the most widely used high resolution methods and was discovered in 1979. It can be as an improvement or generalization of Pisarenko's method. Most of the derivations in the following sections are adopted from [8], [6].

From eq. 4.2, the autocorrelation matrix results in:

$$R_p = E R_c E^H + \rho_w I_p = S_p + W_p \tag{4.6}$$

Where R_c is the autocorrelation matrix of vector $\boldsymbol{c}$, in 4.2. ρ_w are the noise power and I_p is the identity matrix of order p.

The matrices S_p and W_p in 4.6 constitutes the signal and noise in the signal space.

Let μ_i and $\boldsymbol{v}_i$ be the eigenvalues and eigenvectors of the matrix S_p, respectively. The decomposition of S_p yields,

$$S_p = \sum_{i=1}^{p} \mu_i \, \boldsymbol{v}_i \boldsymbol{v}_i^H = \sum_{i=1}^{N} \mu_i \boldsymbol{v}_i \boldsymbol{v}_i^H \tag{4.7}$$

Because of the full rank property of S_p in equation 4.7 $\mu_i = 0$ for $i = N+1 \ \ldots \ p$ and since alle the eigenvalues of the identity matrix are equal to 1 and any ortonormal vector can be its eigenvector, the following relationship holds [8]:

$$I_p = \sum_{i=1}^{p} \boldsymbol{v}_i \, \boldsymbol{v}_i^H \tag{4.8}$$

If we combine the equations 4.6, 4.7 and 4.8, we have

$$R_p = \sum_{i=1}^{N} \mu_i \boldsymbol{v}_i \, \boldsymbol{v}_i^H + \rho_w \sum_{i=1}^{p} \boldsymbol{v}_i \, \boldsymbol{v}_i^H = \sum_{i=1}^{N} (\mu_i + \rho_w) \boldsymbol{v}_i \, \boldsymbol{v}_i^H + \sum_{i=1}^{p} \rho_w \boldsymbol{v}_i \, \boldsymbol{v}_i^H \tag{4.9}$$

The main eigenvectors, corresponding to the N largest eigenvalues, span the same subspace as the signal vectors, while the other eigenvectors span the noise subspace. Since all the eigenvectors of an autocorrelation matrix are orthogonal, the two subspaces are also orthogonal.

Define the mode vector as:

$$\boldsymbol{a}_i = [1 \quad exp(j2\pi T_s v_i) \quad ... \quad exp(j2\pi(p-1)T_s v_i)]^T \tag{4.10}$$

$$PSD_{MUSIC}(v_i) = \frac{1}{\sum_{l=N+1}^{p+1} \left|\boldsymbol{a}_i^H v_i\right|^2} = \frac{1}{\boldsymbol{a}_i^H (\sum_{l=N+1}^{p+1} V_l \; V_l^H) \boldsymbol{a}_i} \tag{4.11}$$

Equation 4.11 is the classical MUSIC estimator. The peak values of the spectrum tend theoretically towards infinity at frequencies corresponding to a spectral component, but does not in practice because of limited memory in computation and estimation errors. Referencing [8], [2] the resolution is improved and are able to outperform non-parametric methods on frequency estimation.

The basis of the MATLAB implementation of Root-MUSIC used in this project is taken from [8], but major tweaks regarding picking out the useful roots of the polynomial and estimation of autocorrelation matrix where needed for sufficient resolution.

Root-MUSIC does not produce a spectrum like MUSIC, but instead its making a direct calculation of spectral components in a line specter.

Consider the polynomial

$$P_l(z) = v_l^H \; \boldsymbol{q}(z), l = N+1..p \tag{4.12}$$

where $\boldsymbol{q}(z) = [1 \quad z \quad ... \quad z^p]^T$

The N zeros of each polynomial in equation 4.12 are represented by $z_k = exp(j2\pi T_s f_k)$ for $k = 1 \; ... \; N$. In this case, the vector $\boldsymbol{q}(z)$ becomes a signal vector, that is, orthogonal to any eignenvector $\boldsymbol{v}_l$.

The problem left is now to calculate the zeros of the polynomials in 4.12. The useful zeros are the ones located closest to the unit circle.

$$P(z) = \boldsymbol{q}(z)^H V_n V_n^H \boldsymbol{q}(z) \tag{4.13}$$

And the Root-MUSIC frequency estimator

$$P_{RM}(z) = z^p \boldsymbol{q}(z^{-1})^T \; V_n \; V_n^H \boldsymbol{q}(z) \tag{4.14}$$

4.2.2 ESPRIT

ESPRIT (EStimation of signal Parameters via Rotational Invariance Techniques) is also a high resolution frequency estiamation algorithm. A detailed derivation of the estimator is omitted in this project, but can be found in [8], [4] and [6]. The MATLAB implementation is directly adopted from [8].

There also exists methods to reduce the complexity of root-MUSIC and ESPRIT. For instance [4], [8] proposes an frequency selective ESPRIT implementation that are not considered in this project.

4.2.3 Signal subspace dimension and covarinace matrix order

The introduced high resolution methods from 4.2 needs an accurate estimation of the signal subspace dimension, n [8]. One of the most common tools for estimating the signal subspace dimension is the Akaike information criterion
Defined as

$$AIC = 2k - 2ln(L) \tag{4.15}$$

Where L is the maximum value of the likelihood function for the model and k is the number of estimated parameters in the model.

The signal subspace dimension is chosen to twice the number of reflections in the tank. Since we have 6 reflectors and one liquids surface, we have a subspace dimension of 14.

4.3 The Existing Model

This section describes the existing algorithm for estimating the ullage in the tank. The procedure is first highlighted, than discussed in the subsequent sections.

4.3.1 The existing algorithm

From the generation of the reflection model based on a cascade of ABCD-matrices in section 3.3, the existing algorithm has essentially the following procedure:

- Take the real value of the reflection
- (optionally: Range equalizing)
- Dispersion correction.
- Take a Hanning windowed inverse Fourier transform to transform from a frequency dependent reflection to the ullage (distance) domain.
- Map the Fourier step to an ullage.
- Use a 5-point or 2-point center of gravity calculation on the echo associated to the liquid surface and return the result.

4.3.2 5-point center of gravity

The center of gravity (COG) are applied for improved precision of finding the fraction of the echo that is closest to the liqud surface. A weighted sum of the 4 closest points from the peak associated with the liquid surface is used. When the liquid surface is close close to a reflector, we would introduce an error because the reflector echo will impact the COG-calculation and pull the estimated level closer to the reflector. The sidelobes of the Hanning window will also contribute.

4.3.3 The effect on zeropadding

Zero padding refers to adding zeros at the end of a signal before transforming. It allows one to use a longer FFT, which will produce a longer FFT result vector. A longer FFT result has more frequency bins that are more closely spaced in frequency, but will essentially provide the same result as a high quality Sinc interpolation of a shorter non-zero-padded FFT of the original data.

However, this brings no aditional information to the signal and this interpolation will not help with resolving closely separated sinusoids, nor will it increase the resolution between adjacent samples. On the other hand, this might be beneficial to the 5-point-center of gravity calculation, since you have more points and in turn get less contribution from a close sinusoid in frequency.

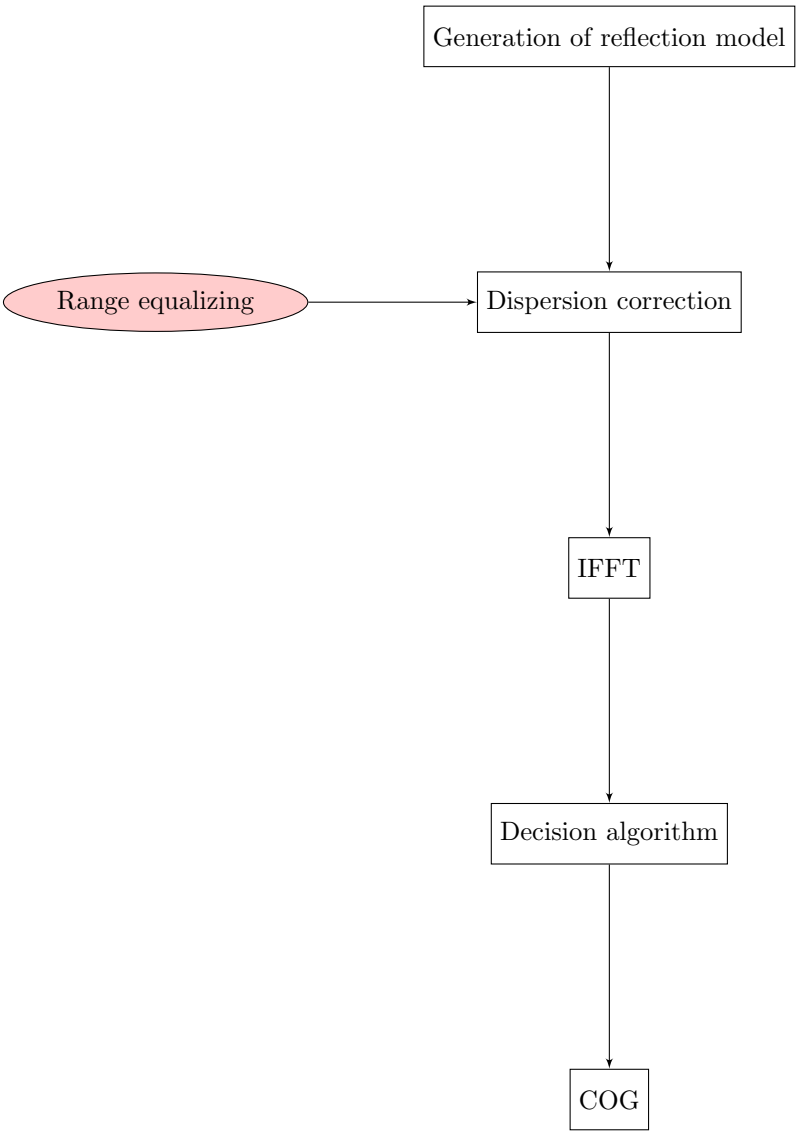

Figure 4.1: Existing procedure for estimating ullage in the system

4.3.4 Range equalizing

In the algorithm presented in the existing model section 4.3, there is a proposed a range equalizing filter. In order to equalize the attenuation of the signal from the reflector furthest away from the radar source. This is essentially a filter that amplifies the signal linearly as a function of frequency. As the signal contains noise, the noise will also be amplified by the same factor. We will then have an inverse pink noise process with a power spectral density proportional to the frequency of the signal, and the reflectors furthest away from the source will have much more noise than the closest. This is a very unpredictable way of designing a gauge system, and has been discarded in the current model.

4.3.5 Dispersion correction

As the waveguide is dispersive, there is developed a dispersion correction routine. The source code for the routine is proprietary.

4.4 High resolution approach

The following steps was made to apply MUSIC and ESPRIT

- Take the real value of the reflection coefficient
- Dispersion correction
- Remove the DC-component of the reflected signal
- Apply Root-MUSIC or ESPRIT
- Map the predefined n spectra to an ullage
- A decision algorithm to pick out the line specter associated with the liquid surface.
- Feed the measurement into the radar tracking algorithm

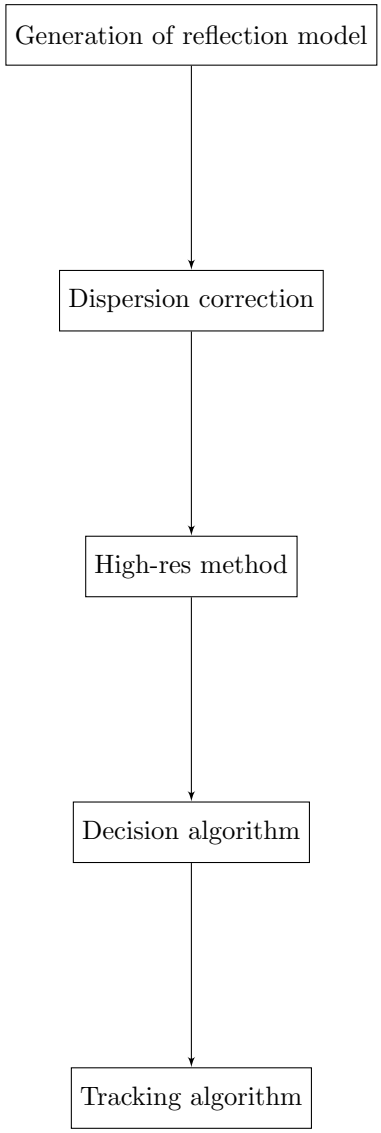

Figure 4.2: Proposed procedure for estimating ullage in the system

Chapter 5

Evaluation

The results in this chapter are obtained from evaluating the proposed algorithms for several different scenarios and noise. A comparison with the existing Fourier-transform model is also included. As mentioned in the limitations, since some of the algorithms produce line spectra, the problem of deciding which peak that corresponds to to liquid surface is done by a decision algorithm
In order to quantify the estimation error, there is natural to define the error as the difference between the estimated liquid level and the true liquid level. In several configurations, different simulations have been completed where the liquid surface have been consequently moved a distance of $\Delta x = 1mm$ for each iteration. In order to compare for different noise variances σ_n^2, we define σ_e^2 to be the variance of the estimation error in the following way:
Let the estimation error e be

$$e = v - \hat{v} \tag{5.1}$$

where v and $\hat{v}$ are the true and estimated liquid level. The variance of the error from 5.1 is

$$\sigma_e^2 = E(e - \bar{e})^2 \tag{5.2}$$

where $\bar{e}$ is the mean of the errors.

The tests used for evaluation are:

1. Filling the whole pipe at constant velocity with no noise.
2. Filling the whole pipe at constant velocity with added Gaussian noise

Consequently for all subspace based algorithms, a subspace dimension of 14 and a covariance matrix order of 200 have been used. A justification for this choice is made in the discussion section. The results in the next section shows figures of the estimation error defined in 5.1.

5.1 Tracking the liquid surface and decicion algorithm

As previously stated in section 4.2, the Root-MUSIC and ESPRIT produce a number of line spectra as equal to the specified number of complex sinusoids in

the model. We need a robust decision algorithm that can decide which frequencies that corresponds to reflector and which corresponds to the liquid surface. In the case when the liquid surface and reflector position completely or almost completely coincide, the number of complex sinusoids will be reduced by two. In other words, the subspace dimension will be higher than the number of complex sinusoids in the model in which the algorithm will produce a frequency not related to the liquid surface or any of the reflectors. This is a problem the decision algorithm will need to handle and could potentially produce erroneous results.

In the simulation environment this can be done in an ad-hoc fashion, since we know exactly the position of the liquid surface. By using this information in the decision algorithm, tracking the liquid surface can be done by choosing the frequency that is closest to the true liquid surface frequency, assuming that this is always correct. The simulation results are obtained in that fashion.

When we are dealing with physical measurements, we no longer have access to the true liquid level. It is therefore impossible to know *exact* where the true level is, and the decision algorithm need to be modified accordingly. The source code for the decision algorithm used in the Gastrail can be found in Appedix.

5.2 Simulation results

Firstly, the algorithms were evaluated using the reflection pattern described in section 3.3. The plot below shows the error in estimation of Root-MUSIC, ESPRIT and Fourier-based spectral estimation. The plots 5.1, 5.2 and 5.3 shows the difference between the estimated liquid level and the true liquid level. The blue line is the Kalman filtered version of the same estimation error. The inital values of the Kalman filter are on purpose set to be off the correct value to see how well it converges to the measurements.

5.3 Gastrial results

The results from measurements in this section are from real physical data of a 3 hour recording of filling with constant rate. The figures 5.6, 5.4, 5.5 shows the estimated values and the Kalman filtered values. Note that because the measurement are divided into several series, there is a small gap in the measurements at approximately 11000s and 17000s after start up. The reflector is located at $Ullage = 30.05m$

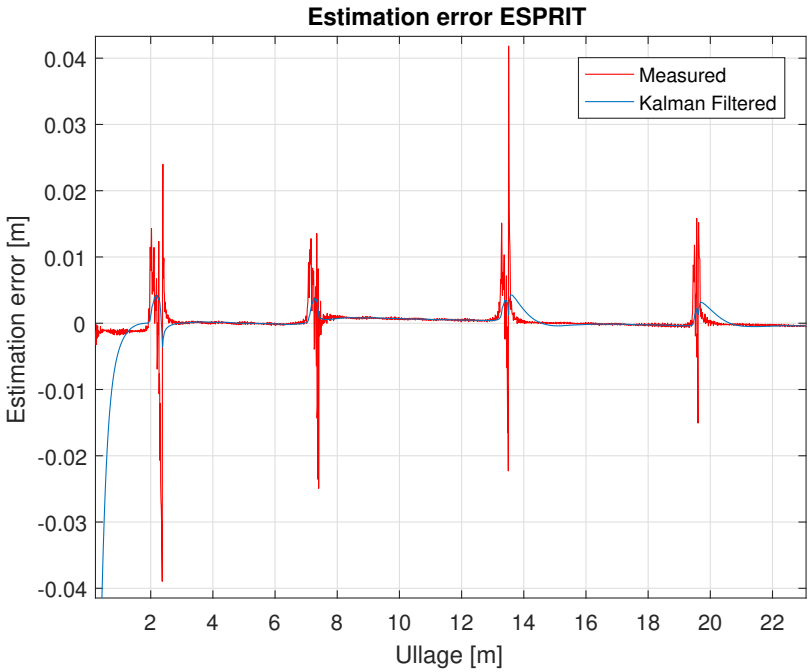

Figure 5.1: Estimation error, ESPRIT

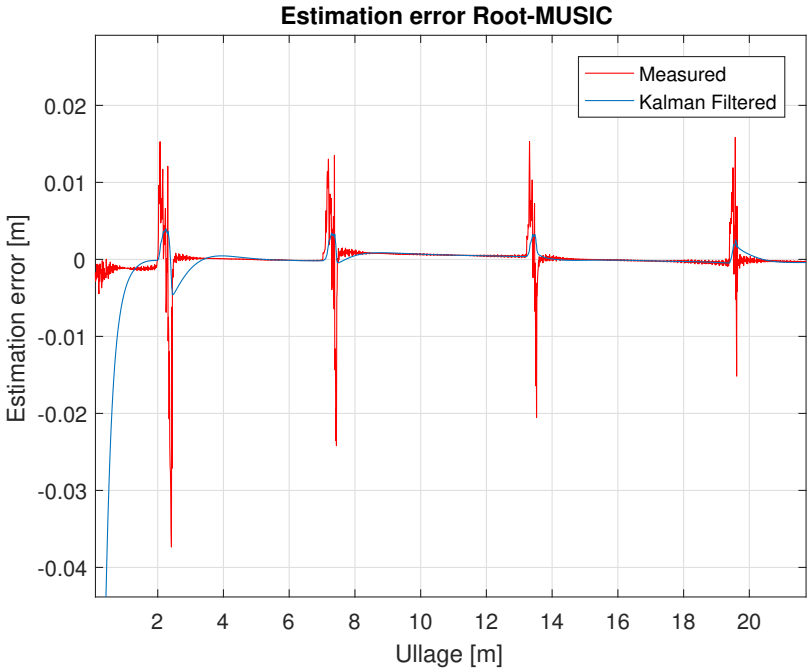

Figure 5.2: Estimation error, Root-MUSIC

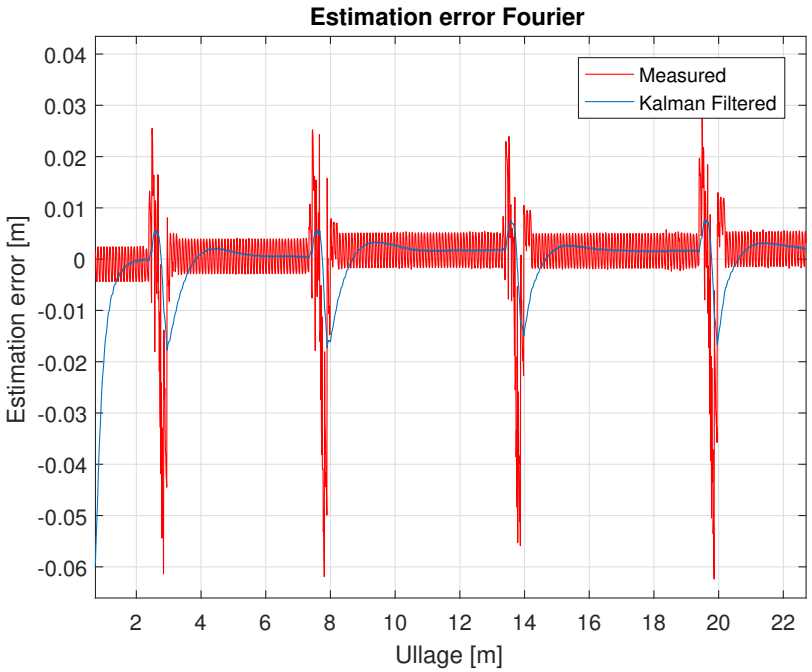

Figure 5.3: Estimation error, Fourier.

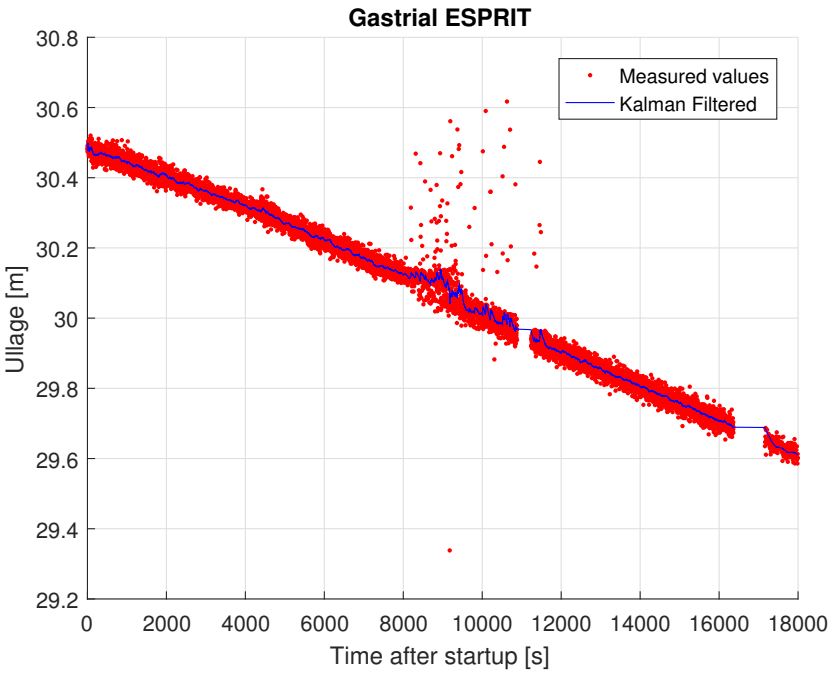

Figure 5.4: Gastrial ESPRIT

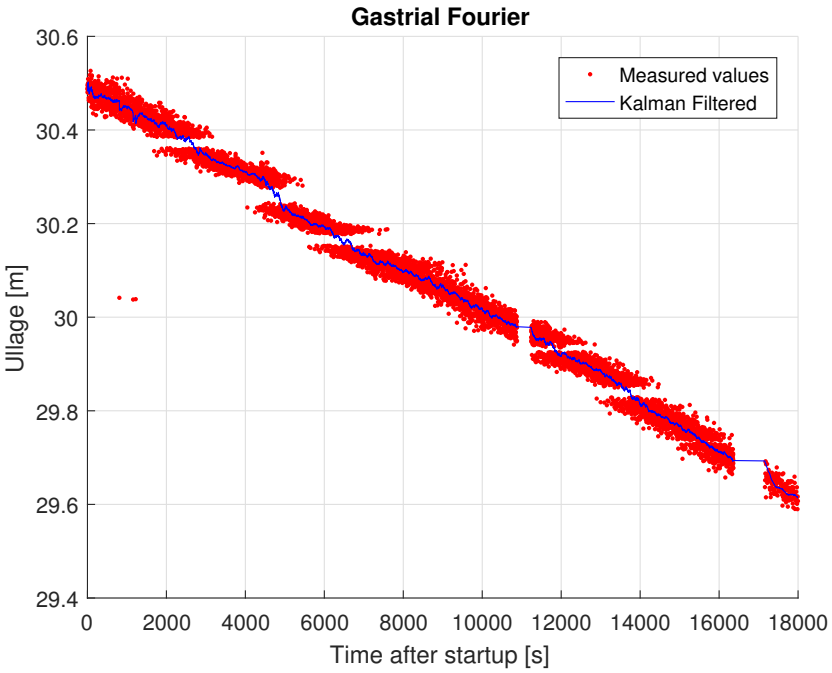

Figure 5.5: Gastrial Fourier

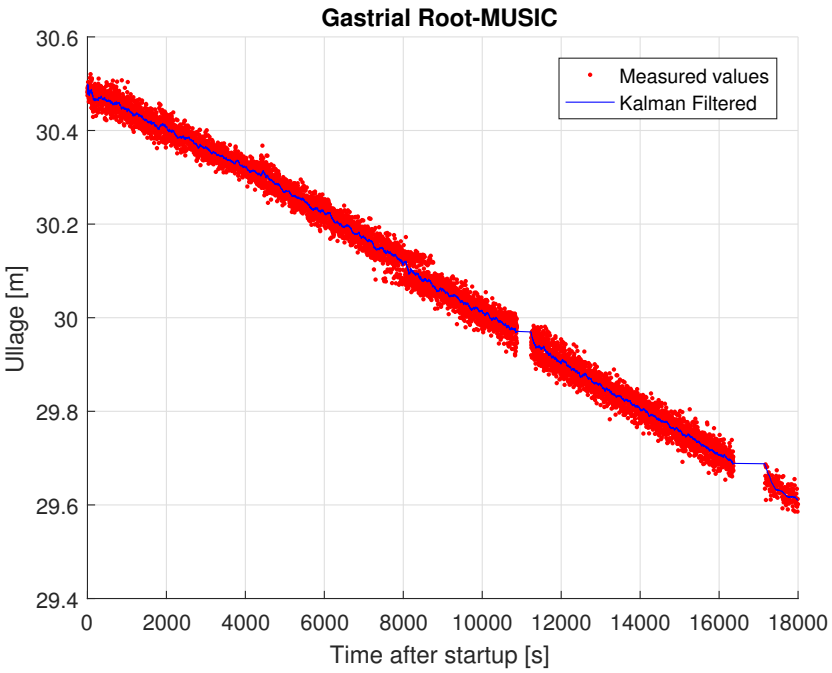

Figure 5.6: Gastrial Root-MUSIC

Chapter 6

Discussion and conclusion

The simulation results show that Root-MUSIC and ESPRIT are applicable to use for radar tank gauging. Regarding Root-MUSIC which turned out to have best accuracy with a maximum estimation error of approximately 3.75cm. Together with the tracking algorithm, it satisfies the accuracy requirement of $\pm 5mm$ estimation error. The results from figure 5.1 and 5.2 shows that there are not much difference between ESPRIT and Root-MUSIC and they outperform the Fourier-based algorithm in the simulation. Not only does the Fourier-based algorithm have the biggest estimation error, but it also have oscillations in the areas between reflectors that are relatively big.

The order of the covariance matrix n used in Root-MUSIC and ESPRIT are of great importance, both for the accuracy and the computational complexity. The matrix order used in the simulations as well as the Gastrail where decided to be $n = 200$. [8] argues that these parameters may be chosen as large as possible, but in order to have a reliable estimation, not too close to the available samples. This will be a tradeoff between statistical accuracy and computation time. In addition, there exist a lot of methods to estimate the covariance matrix efficient and in other ways then done in this project. [8]

The results from the Gastrail figure 5.4, 5.5 and 5.6 shows interesting differences between the algorithms. The Fourier-based Gastrial result, figure 5.5, have a overall higher variance than the two others. In addition, one can see areas where no measurements at all located at distances that is a multiple of $\frac{\lambda}{2}$ away from the reflector, where λ is the wave length of the carrier frequency. The reason for this is interference as well as the COG-calculation that pulls liquid spectrum away from its true position. For the Gastrail using ESPRIT, figure 5.4 we see the area around the reflector where measurements are way off its true value. This is due to the decision algorithm fail to pick out the correct frequency due to a reduction in the signal subspace.

One critical issue of using high resolution algorithms in this the fact that one need to know the signal subspace dimension in advance. Since we have 6 reflectors and a liquid surface, we have in total 7 reflections for each sweep. In the case when the the liquid surface completely coincide with a reflector, we will have a reduction in in the subspace dimension by 2. This result in a new

estimated frequency not related to any of the reflectors, nor liquid surface and may produce erroneous results if the decision algorithm is not optimal. Figure 5.4 showing the Gastrail of ESPRIT demonstrates this issue when the liquid surface is passing the reflector and obviously wrong results are picked out.

The radar tracking algorithm worked satisfactory in the both in the simulation and Gastrail. It is assumed that the liquid at constant rate fills or empties the tank. If we had take unexpected stops or other abnormalities in the filling process into account the Kalman filter tracking algorithm must have been extended to handle such cases and testing with the existing tracking algorithm for sudden stops did not produce good results.

Even though the Gastrail measurements are done in a controlled environment, filling or other movements in the tank give rise to different kinds of forces making the liquid move in ways it is hard to predict. The liquid could move in any direction and non-stationary sloshing waves takes place. Ideally, this set of algorithms should be able to handle sloshing and vibrations from for instance sea waves or thrusters, but there is a much higher variance in the measurements than in the simulated case.

Because we do not have any true liquid level in the Gastrial, we can not say how big the estimation error is. We can not with certainty conclude on the basis of this measurement alone.

A number of assumptions have been made in this project. First, the assumption that the transmission line model from section 3.3 is a good approximation of the waveguide. Also assuming the noise additive Gaussian might not reflect the real world case. The sections are also assumed to be homogeneous. In situations where it is not a clean transition between sections, mixture of liquids with different characteristics and such, can the accuracy decrease. There is also noise originating from imperfections and scratches inside the waveguide that can lead to noise that are quite difficult to model.

6.1 Conclusion

The results from this project have shown that, given this reflection model 4.2, we can improve the accuracy of level gauging using Root-MUSIC or ESPRIT as spectral estimation algorithm in lieu of the existing Fourier-based algorithm. Assuming that the reflection model used are a good approximation, it outperforms the Fourier-based spectral estimation algorithm in terms of accuracy in the problematic areas close to the reflectors.

Root-MUSIC and ESPRIT introduces increased computationally complexity and requires knowledge of the signal model to be applicable to the problem.

ESPRIT and Root-MUSIC can roughly double the accuracy compared to a non-parametric approach. The algorithms work under idealized conditions, but

the resolution problem is still visible around the reflectors.

The frequency estimates of the algorithms evaluated are very sensitive for changes in the parameters, and must be chosen carefully. Although the requirement accuracy of $5mm$ was not met with only spectral estimation, using a tracking algorithm can satisfy the accuracy criterion if we assume constant filling rate.

Bibliography

[1] R. Brown and P. Hwang. *Random Signals and Applied Kalman Filtering*. John Wiley Sons, Inc., 2012.

[2] S.M Kay. *Fundamentals of statistical signal processing : estimation theory*. Prentice-Hall Inc., 1993. ISBN: 0133457117.

[3] I.V. Komarow and S.M Smolskiy. *Fundamentals of Short-Range FM Radar*. Artech House, 2003.

[4] D Manolakis and J Proakis. *Digital Signal Processing: Principles, Algorithms, And Applications. 4th ed.* Prentice-Hall Inc., 1992. ISBN: 978-0131873742.

[5] 4th Edition Pozar D. *Microwave Engineering*. John Wiley Sons, Inc., 2004. ISBN: 9780470631553.

[6] A. Quinquis. *Digital Signal Processing: Using MATLAB*. John Wiley Sons, Inc, 2007. ISBN: 978-1-84821-011-0.

[7] L. Roger. *Kalman and Bayesian Filters in Python*. `https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python`. 2014.

[8] P. Stoica and R Moses. *Spectral Analysis of Signals*. Prentice-Hall Inc., 1997. ISBN: 0131139568.

Appendix A

Additional Information

A.1 MATLAB code for Root-MUSIC, ESPRIT and Kalman Filter

Listing A.1: Root-MUSIC MATLAB implementation

```
1
2  function w = rmusic(x,n,m,fs)
3  % Root-MUSIC frequency estimation algoritm
4  % Written by Henning Schei
5  % The basis of the code is adopted from Spectral Analysis
       of Signals,
6  % Stocia, 97
7
8
9  %       x  ->  the data vector
10 %       n  ->  the model order
11 %       m  ->  the order of the covariance matrix
12 %       w  <-  the frequency estimates
13
14 % variable check
15     narginchk(3,4)
16     if nargin == 3
17         fs = 1;
18     end
19
20
21     R = compute_autocovariance(x,m);
22 % singular value decomposition
23     [U,~,~] = svd(R);
24     G = U(:,n+1:end);
25     P = G * conj(G)';
26
27
28 % Calculate sum of each diagonal in P
```

```
29      [r ,c]=size(P);
30      idx=bsxfun(@plus,(r:-1:1)',0:c-1);
31      Q=flipud(accumarray(idx(:),P(:)));
32
33  % Find the roots inside the unit circle and imag-val != 0
34      rts = roots(Q);
35      rts = rts(abs(rts) < 1);
36      rts = rts(imag(rts) ~=0);
37  % Find the n roots closest to the unit circle
38      dfc = abs(abs(rts)-1);
39      idx_sort= argsort(dfc);
40      component_roots = rts(idx_sort(1:n));
41
42
43      ang = -angle(component_roots);
44      w = fs*ang;
45  end
46  %%
47  function R = compute_autocovariance(x,M)
48      % Henning Schei
49      % Function to compute the corvarinace matrix of a
            signal.
50
51
52      % x --> array of input signal of size N
53      % M --> int, opitional. Size of signal block
54      % covmtx <-- NxN auto-covariance matrix
55
56      N = length(x);
57
58      x_vect = x(:); % Force x to be a column vector
59
60      % Initial covariance matrix
61      yn = flipud(x_vect(1:M));
62      R = yn * conj(yn)';
63      for i = 2:N-M
64          yn = x_vect(M-1+i:-1:i);
65          R = R + yn * conj(yn)';
66      end
67
68      R = R/N;
69  end
70
71
72  %%
73  function [sorted_ind, sorted_M] = argsort(M,mode)
74      % Argument sort
75      % Henning Schei
76      if (~exist('mode', 'var'))
77          mode = 'ascend';
```

```
78          end
79          nRows = size(M, 1);
80          indices = (1 : nRows)';
81
82          M_with_indices = horzcat(M, indices);
83          [sorted_M, sorted_ind] = sort(M_with_indices(:,1),
                mode);
84      end
```

Listing A.2: ESPRIT MATLAB implementation

```
1   function w=esprit(y,n,m)
2   %
3   % The ESPRIT method for frequency estimation.
4   % Taken from 'Spectral Analysis of Signals',
5   % Written by R. Moses
6   % w=esprit(y,n,m);
7   %
8   %       y  ->  the data vector
9   %       n  ->  the model order
10  %       m  ->  the order of the covariance matrix in
        (4.5.14)
11  %       w  <-  the frequency estimates
12  %
13  %
14  y=y(:);
15  N=length(y);
16  R=zeros(m,m);
17  for i = m : N,
18      R=R+y(i:-1:i-m+1)*y(i:-1:i-m+1)'/N;
19  end
20  R=(R+fliplr(eye(m))*R.'*fliplr(eye(m)))/2;
21  % get the eigendecomposition of R; use svd because it
        sorts eigenvalues
22  [U,D,V]=svd(R);
23  S=U(:,1:n);
24  phi = S(1:m-1,:)\S(2:m,:);
25  w=-angle(eig(phi));
```

Listing A.3: Kalman filter recursive equation implementation

```
1
2   function s = kalmanf(s)
3       % Prediction for state vector and covariance:
4       s.x = s.A*s.x + s.B*s.u;
5       s.P = s.A * s.P * s.A' + s.Q;
6
7       % Compute Kalman gain factor:
8       K = s.P*s.H'/(s.H*s.P*s.H'+s.R);
9
10      % Correction based on observation:
```

```
11      s.x = s.x + K*(s.z-s.H*s.x);
12      s.P = s.P - K*s.H*s.P;
13  end
```