University of California San Diego

A Beat-Based Approach to Range Estimation

A Thesis submitted in partial satisfaction of the requirements

for the degree Master of Science

in

Engineering Sciences (Mechanical Engineering)

by

Junyou Yue

Committee in charge:

Professor Mauricio de Oliveira, Chair
Professor Andrew Lucas, Co-Chair
Professor Miller Puckette

The Thesis of Junyou Yue as it is listed on UC San Diego Academic Records is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Co-Chair

Chair

University of California San Diego

2019

Table of Contents

List of Figures

ABSTRACT OF THE THESIS

A Beat-Based Approach to Range Estimation

by

Junyou Yue

Master of Science Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2019

Professor Mauricio de Oliveira, Chair
Professor Andrew Lucas, Co-Chair

Range estimation is a widely used tool in many different disciplines and it can be achieved in a multitude of ways. This thesis outlines a new principle for range estimation that utilizes existing technologies and is inspired by previously established methods.

The main principle employs one pair of transmitter and receiver to send a continuous, passively-measured signal that contains all the information needed in order to determine the distance between them. This is done by retrieving the phase information inside the phase-modulated transmission signal and using it to calculate the time of arrival. All of this is achieved under the assumption that the clocks of transmitter and receiver are synchronized already.

Numerous simulations were run based on a model that was created just to test this principle. Simulations were successful and accurate to less than 1% once steady state was achieved. Physical testing was done using a microphone and speaker setup inside a DSP with less accurate results, but further work can be done to refine the system.

Chapter 1

Introduction

The application of ranging techniques for position estimation is widely used in numerous fields, from wireless communication systems to the maritime industry. There are many established schemes that have been used. Time-of-arrival (ToA), Time-difference-of-arrival (TDoA), and received signal strength (RSSI) techniques, among others, have been all accepted as reliable detection methods [1]. These various methods can ultimately be incorporated into localized Wireless Sensor Networks (WSN), comprised of multiple transmitters and receivers, that can infer the receiver's position by using its surrounding distance estimates. Most of these methods are similar in that they leverage triangularization or multilateration to build a projection of their receivers' locations. However, there are some drawbacks to these methods. For example, ToA and TDoA are extremely dependent on the synchronization of clocks along the receivers. This leads to heavy packet transmission between the transmitter and receiver as well.

The goal of this thesis was to utilize ultrasonic acoustic signals for the purpose of creating a new range estimation scheme. Preceding work done into this topic all involve the previously listed methods, whereas the work achieved here presents a novel approach to range detection that leverages an elegant and simple solution. The presented principle for proximity measurement allows the user to passively measure

distance as a continuous waveform. The main focus of the following work is in the implementation and principle of the range detection, and the use of this method for localization purposes will not be explored in detail.

With this prospective scheme, an accurate estimation of the distance can be passively determined through the use of a single continuous-wave signal. This signal, which is the aggregate of two waveforms at their own distinctive frequencies, is sent under the condition that the transmitter and receiver both have synchronized clocks already. Once received, the signal consists of crucial phase information that can be leveraged to derive the distance. This main principle will be outlined in the next chapter.

In comparison to previous work, this range detection scheme is most similar to that done in Radio Interferometric Geolocation, by Maroti et al. [2]. The main difference between the works is that their method, the Radio Interferometric Positioning System (RIPS), requires more than one transmitter and receiver in order to be effective, whereas this method only needs one pair of transmitter and receiver to function. Another method named Chronos, by Vasisht et al. [3], utilizes multiple frequencies in radio WiFi systems. Similar to RIPS, Chronos takes advantage of *existing* signals in order to provide a single estimate of the distance. In the following presented method, a signal is specifically *designed* just for the purpose of range estimation. There are also other reviews of existing technologies that can be found in [4], [5], and [6].

Most of the work done on this approach was achieved through Simulink, a graphical environment within MATLAB that can be used to simulate complex models. The physical testing was done through Pure Data, an open source graphical programming tool built for the purpose of signal processing.

Chapter 2

Main Principle for Range Estimation

2.1 Main Principle

In this chapter, the fundamental estimation scheme will be concisely outlined without delving too deeply into each respective component. By comparison, Chapter 4 will provide a more thorough perspective on each step of the procedure.

Under the assumption that a receiver and transmitter are already synchronized, a signal comprised of the summation of two waveforms at different frequencies can be transmitted. Once received, the signal has phase data that contains the essential information needed to find the delay.

Let $f_2 = f_1 + \delta_f$, with δ_f being a small, positive frequency differential. The following waveform, composed of these two frequencies, produces a signal with a low frequency envelope beat.

$$y_t(t) = A_1 \cos(2\pi f_1 t + \phi_1) + A_2 \cos(2\pi f_2 t + \phi_2) \tag{2.1}$$

$$f_2 = f_1 + \delta_f, \quad \delta_f \geq 0 \tag{2.2}$$

At the receiver, the signal is delayed by ν seconds where $\nu = d/c$, in which d is the distance from the transmitter to the receiver and c is the speed of propagation. The

receiver therefore acquires the following signal

$$\begin{aligned} y_r(t) &= y_t(t-\nu) \\ &\approx A_1 \cos(2\pi f_1 t + \phi_1 - 2\pi f_1 \nu) + A_2 \cos(2\pi f_2 t + \phi_2 - 2\pi f_2 \nu) \end{aligned} \tag{2.3}$$

Note that the phase of each component of the received signal is

$$\psi_1 = \phi_1 - 2\pi f_1 \nu \tag{2.4}$$

$$\psi_2 = \phi_2 - 2\pi f_2 \nu \tag{2.5}$$

The values of $2\pi f_1 \nu$ and $2\pi f_2 \nu$ are expected to be very large relative to that of ϕ_1 and ϕ_2. Due to ambiguity and lack of knowledge of the transmitter phases, the values of ψ_1 and ψ_2 can't be measured at the receiver. Instead, a phase detector can be used to obtain these measurements. For instance, a phase-locked loop can be utilized, but other methods such as autocorrelation are also viable.

These recovered phases are such that

$$\hat{\psi}_1 = \psi_1 - 2\pi k_1 \tag{2.6}$$

$$\hat{\psi}_2 = \psi_2 - 2\pi k_2 \tag{2.7}$$

$$\hat{\psi}_1, \hat{\psi}_2 \in [0, 2\pi), \quad k_1, k_2 \in \mathbb{Z} \tag{2.8}$$

These phases are both able to be resolved in the interval $[0, 2\pi)$, with each respective k value being the integer that brings the value into that interval. From (2.4-2.7), it can be seen that

$$\hat{\psi}_1 - \hat{\psi}_2 = 2\pi \delta_f \nu - \delta_\phi + 2\pi(k_1 - k_2) \tag{2.9}$$

$$\delta_\phi = \phi_2 - \phi_1 \tag{2.10}$$

In order to fully solve for ν, the constraints of k_1 and k_2 must be resolved first. We can derive of the difference of the phases to be

$$\psi_1 - \psi_2 = 2\pi\delta_f\nu - \delta_\phi \tag{2.11}$$

It can be seen that $\psi_1 - \psi_2$ is bounded within $[0, 2\pi]$, and therefore the following must be held to be true:

$$\delta_\phi \leq 2\pi\delta_f\nu \leq 2\pi + \delta_\phi \tag{2.12}$$

By constraining $\psi_1 - \psi_2$ to be within the period of $[0, 2\pi]$, we obtain a crucial condition in that $\psi_2 \leq \psi_1 \leq \psi_2 + 2\pi$. This implies that the wrapping of the two phases across the periods are in close proximity of each other. At the most, ψ_1 and ψ_2 can't be separated by more than 2π, which means that k_1 and k_2 can't be more than 1 apart.

$$k_1 - k_2 \in \{0, 1\} \tag{2.13}$$

Now (2.9) can be revisited to solve for the delay in ν. We are presented with only two possible scenarios:

1. $\hat{\psi}_1 \geq \hat{\psi}_2$: in which case $k_2 = k_1$ and

$$\hat{v} = \frac{1}{2\pi\delta_f}\left(\hat{\psi}_1 - \hat{\psi}_2 + \delta_\phi\right) \tag{2.14}$$

2. $\hat{\psi}_1 < \hat{\psi}_2$: in which case $k_1 = k_2 - 1$ and

$$\hat{v} = \frac{1}{2\pi\delta_f}\left(2\pi + \hat{\psi}_1 - \hat{\psi}_2 + \delta_\phi\right) \tag{2.15}$$

If the speed of propagation is known, then the range d can be calculated to be

$$\hat{d} = c\,\hat{v} \tag{2.16}$$

This outlined principle for detection can be universally applied to any propagation method, but in the following chapters, acoustic signals are used to demonstrate this system's effectiveness. The principle itself is explored with much more detail in Chapter 3.4 and Chapter 4.

Chapter 3

Overview of Implementation

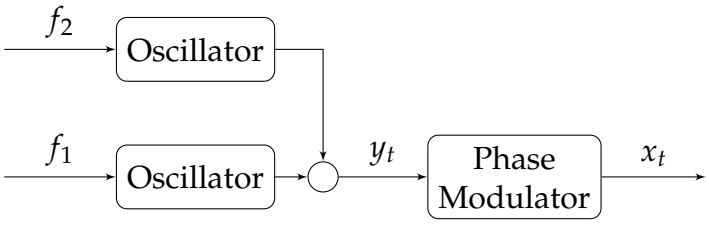


FIGURE 3.1: Transmitter block diagram

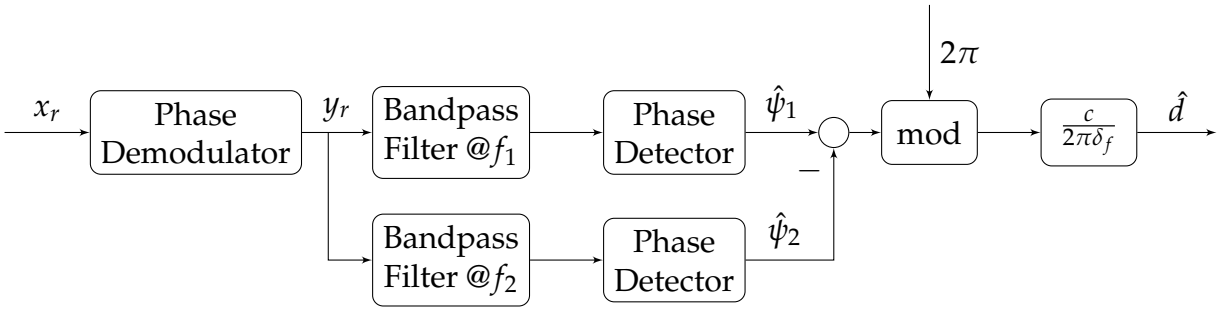


FIGURE 3.2: Receiver block diagram

A high level block diagram of the system is shown here separated between the transmitter and the receiver. The system ultimately consists of a signal generator, modulator, demodulator, and distance estimator, and each one of these will be explored within this chapter.

The signal generation and modulation takes place in the transmitter, as seen above. These steps consists mostly of setting up and preparing the signal for the distance estimator, in which most of the complexity within the system occurs in. The receiver applies the demodulation in association with a phase-locked loop, which also acts as the phase detector. These specific blocks within the receiver are discussed in detail in the next chapter.

3.1 Signal Generator

As stated in (2.1), the main signal is comprised of two separate waveforms at different frequencies.

$$y_t(t) = A\cos(2\pi f_1 t + \phi_1) + A\cos(2\pi f_2 t + \phi_2) \tag{3.1}$$

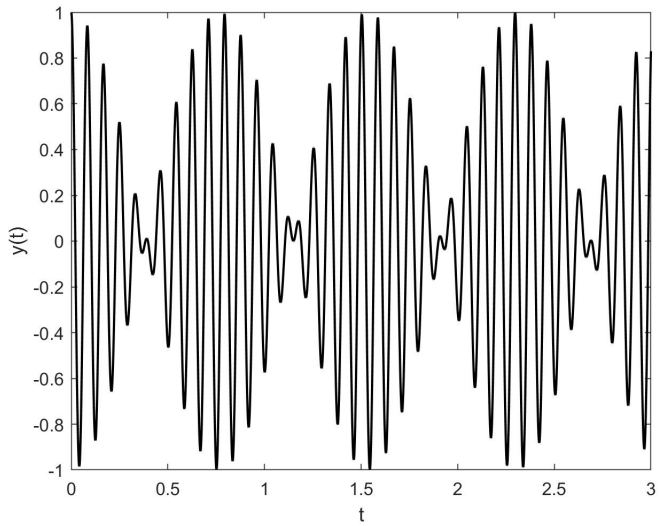


FIGURE 3.3: Plot of $y_t(t)$

A representation of $y_t(t)$ can be seen in Figure 3.3, where a low frequency envelope beat can be clearly distinguished. Values for f_1, f_2, and A were chosen to be 71.1 Hz, 79.3 Hz, and 0.5 respectively, and phases are set to be zero for simplicity.

The selection of f_1, f_2, and δ were motivated by a desired maximum range.

$$d_{max} = \frac{x_{max}}{v} \tag{3.2}$$

$$\omega_\delta = \frac{2\pi}{d_{max}} \tag{3.3}$$

$$\delta = \frac{\omega_\delta}{2\pi} = \frac{1}{d_{max}} \tag{3.4}$$

The value of ω_δ was chosen to be within one period of 2π such to prevent ambiguity. With f_1 and f_2 being exactly δ apart, they are able to be resolved in the later steps. The actual values of f_1 and f_2 can be chosen to be anything that's a magnitude less than the carrier frequency f_c.

3.2 Modulation

Once $y_r(t)$ created, the signal is modulated to be sent out. The choice to use frequency modulation (FM) as opposed to amplitude modulation (AM) was made due to the natural immunity of frequency modulation to attenuation. However, as opposed to amplitude modulation, FM is affected more by physical impediments, such as walls or bodies.

The following modulated signal is transmitted after the modulation process:

$$x_t(t) = \cos(2\pi f_c t + A y_t(t)) \tag{3.5}$$

The carrier frequency f_c can be chosen by the user. For testing purposes, f_c was chosen to be a factor of the sampling frequency, which is set at 44.1kHz. In actual

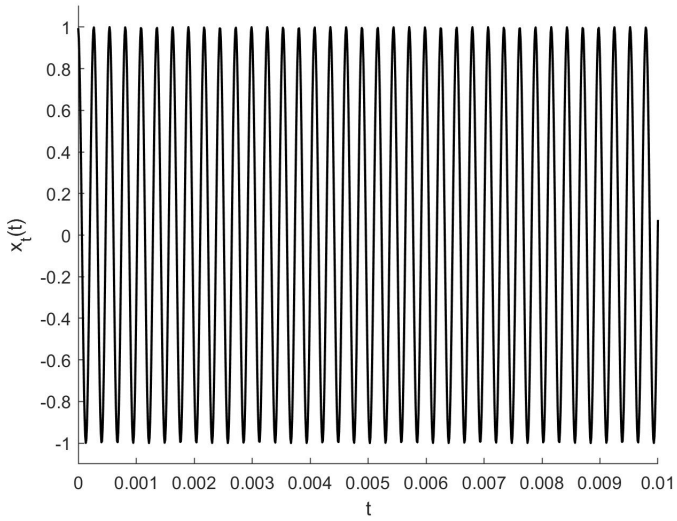


FIGURE 3.4: Plot of $x_t(t)$, with a carrier frequency f_c of 3675 Hz

practice, the carrier frequency is expected to be set in the ultrasonic range $(> 20kHz)$. A modulation index of $A = 0.1$ was used in order to reduce the amplitude of the modulated signal, allowing it to qualify for the small angle approximation $sin(x) \approx x$ in the later steps.

Potential use with multiple sensors in the future can be customized by having different f_c values for each transmitter. The receivers would be able to tell apart each signal by their carrier frequencies and easily assign a range estimate to their respective frequency. This would open a lot of different possibilities for potential localization schemes, but the amount of feasible sensors would be limited to the bandwidth of the system.

3.3 Receiver/Demodulation

In order for the distance algorithm to work correctly, the internal clocks of the transmitter and receiver must first be correctly synchronized. There are numerous

methods to achieve this, but they won't be explored in detail here. Ultimately, the outgoing signal, $x_t(t)$ (from Equation 3.5), is adjusted to have the same phase as the received signal, $x_r(t)$.

Once sent, the signal is treated by a coherent demodulation scheme. In order to reject any noise from the transmission process, a bandpass filter at f_c can be inserted at the reception of the signal. Subsequently, the gathered signal from the receiver is defined as such:

$$x_r(t) = x(t-\nu) = \cos(2\pi f_c(t-\nu) + y(t-\nu) + \phi_c) \tag{3.6}$$

Let $x_\tau(t)$ be a signal such that

$$x_\tau(t) = x(t-\tau) = \cos(2\pi f_c(t-\tau) + y(t-\tau) + \phi_c) \tag{3.7}$$

For $\tau = 1/(4f_c)$,

$$x_\tau(t) = \cos(2\pi f_c t - \pi/2 + y(t-\tau) + \phi_c) \tag{3.8}$$

$$= \sin(2\pi f_c t + y(t-\tau) + \phi_c) \tag{3.9}$$

$$r_p(t) = \cos(2\pi f_c t + \phi_r) x_\tau(t) \tag{3.10}$$

$$= \cos(2\pi t f_c + \phi_r)\cos(2\pi f_c(t-\tau) + y(t-\tau) + \phi_c) \tag{3.11}$$

$$r_q(t) = \cos(2\pi f_c t - \pi/2 + \phi_r) x(t) \tag{3.12}$$

$$= \sin(2\pi t f_c + \phi_r)\cos(2\pi t f_c + \phi_c + y(t)) \tag{3.13}$$

$$
\begin{aligned}
z(t) &= r_p(t) - r_q(t) \\
&= \frac{1}{2}\left(\sin(\phi_c - \phi_r + x(t)) + \cos\left(2\pi\tau f_c - \phi_c + \phi_r - x(t-\tau)\right)\right) + \\
&\quad \frac{1}{2}\left(\cos\left(2\pi f_c(2t-\tau) + \phi_c + \phi_r + x(t-\tau)\right) - \sin\left(4\pi f_c t + \phi_c + \phi_r + x(t)\right)\right)
\end{aligned}
$$

Once again using $\tau = 1/(4f_c)$

$$
z(t) = \frac{1}{2}\left(\sin(\phi_c - \phi_r + x(t)) + \sin\left(\phi_c - \phi_r + x\left(t-\tau\right)\right)\right) \tag{3.14}
$$

$$
\frac{1}{2}\left(\sin\left(4\pi f_c t + \phi_c + \phi_r + x\left(t-\tau\right)\right) - \sin\left(4\pi f_c t + \phi_c + \phi_r + x(t)\right)\right) \tag{3.15}
$$

Therefore, if $x(t-\tau) \approx x(t)$

$$
z(t) \approx \sin(\phi_c - \phi_r + x(t)) \tag{3.16}
$$

The carrier frequency f_c is known by the receiver, which employs coherent demodulation to recover the signal $y_r(t)$. Due to the modulation index applied to $x(t)$, the small angle approximation allows $r(t) \approx x(t)$.

3.4 Range Estimation

Once it's obtained, $y_r(t)$ is put through band-pass filters to separate the two distinct frequency components at f_1 and f_2. Each one of these separate signals, shown below, are put through their own respective PLL to recover their phase information.

$$
y_{r1}(t) = A_1 \cos(2\pi f_1 t + \phi_1 - 2\pi f_1 \nu) \tag{3.17}
$$

$$
y_{r2}(t) = A_2 \cos(2\pi f_2 t + \phi_2 - 2\pi f_2 \nu) \tag{3.18}
$$

Each PLL therefore outputs the recovered phase, which are $\hat{\psi}_1$ and $\hat{\psi}_2$ as seen previously in (2.6-2.7).

The effective range of distances that can be estimated without ambiguity is given by

$$d_0 \leq d \leq d_0 + \frac{c}{\delta_f}, \qquad\qquad d_0 = \frac{c\,\delta_\phi}{2\pi\delta_f}.$$

In particular, with $\delta_\phi = 0$, $d_0 = 0$ and

$$0 \leq d \leq \frac{c}{\delta_f}$$

There are many different methods that can be used to recover the phase of the modulated signal. In this particular design, a phase-locked loop (PLL) is used. Traditionally, a PLL consists of a voltage controlled oscillator (VCO), a phase detector, and some filters. These components are all used to ultimately create a recovered signal that has a phase equal to that of the input signal. Other methods aside from the PLL can also be utilized, with a popular approach being autocorrelation. The benefit of using autocorrelation is such that an aperiodic signal can be be tracked, but that is not necessary for this application.

The next chapter takes a more comprehensive look into the steps and functionality of the distance estimation.

Chapter 4

Detailed Look into the Range Estimator

The outline of the range detection algorithm has been laid out in Chapter 1. In this chapter, the various steps that happen prior to and during the process will be discussed.

4.1 Filtering

Once $y_r(t)$ is successfully retrieved from the demodulation, it needs to be divided into the two waveforms that comprise it. The optimal method to perform this is through two bandpass filters, set up such that only the respective frequency ranges of each signal is allowed through.

$$y_{r1}(t) = \cos(2\pi f_1 t + \phi_1 - 2\pi f_1 \nu) \tag{4.1}$$

$$y_{r2}(t) = \cos(2\pi f_2 t + \phi_2 - 2\pi f_2 \nu) \tag{4.2}$$

There are various classes of filters to derive from, but the Chebyshev Type II filter was chosen in this particular instance because of the filter's behaviours. In general, Chebyshev filters have a steeper transition between its stopband and passband at the expense of having ripples in the passband [7]. However, the component of the filter that was

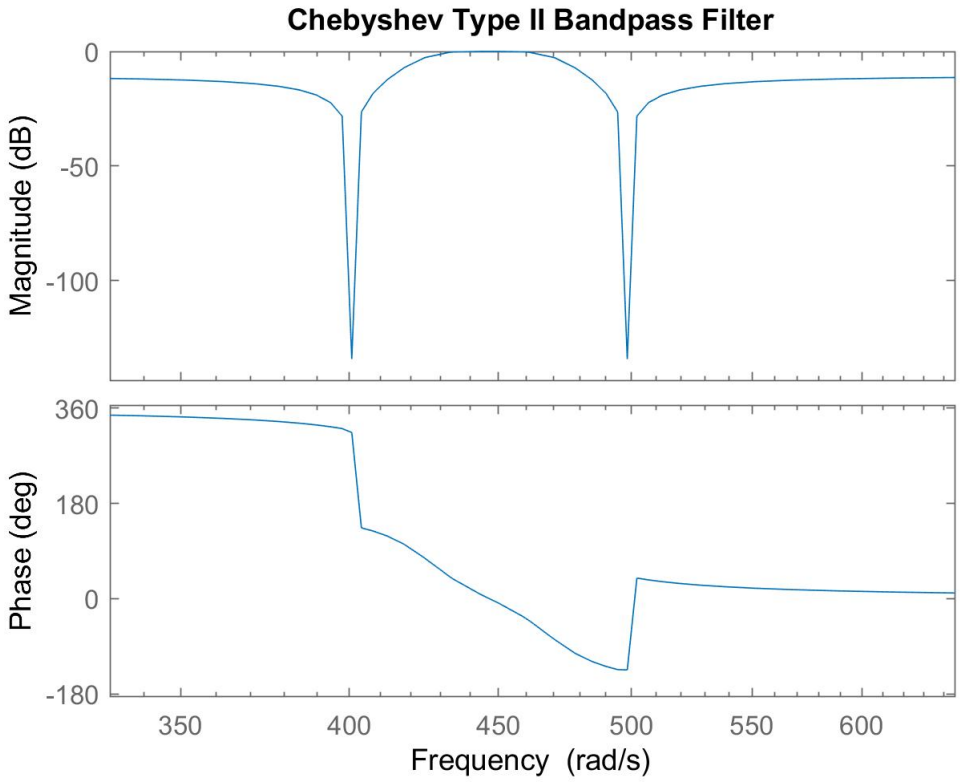


FIGURE 4.1: Bode plot of a Chebyshev type 2 filter

the most desirable was the optional placement of its zeros. If designed correctly, one zero can be placed on the frequency that should be filtered out, while the passband covers the necessary frequency range. Specifically, if the bandpass filter is being created around f_1, then a zero can be placed exactly on f_2, such that not only is extraneous noise filtered out, but the other frequency is completely negated.

The derivation of the Chebyshev filter starts with the low-pass filter equation, as shown below.

$$L_{lp}(s) = \frac{(1+\sqrt{5+4\sqrt{2}})\omega_c^2 + s^2}{\sqrt{6+4\sqrt{2}}J_1 J_2}, \tag{4.3}$$

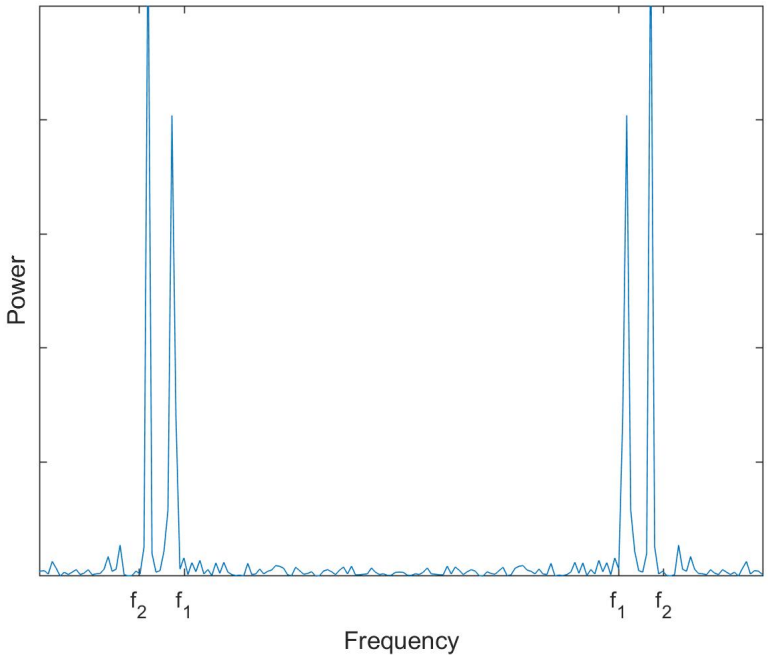
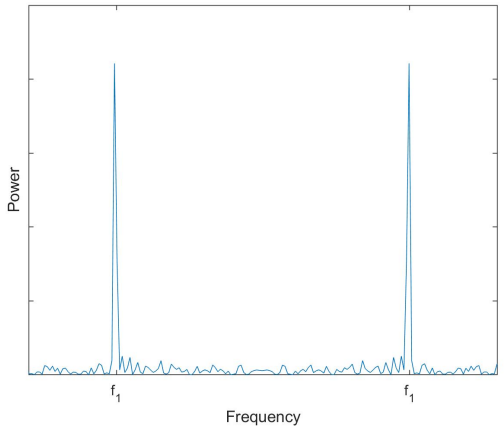
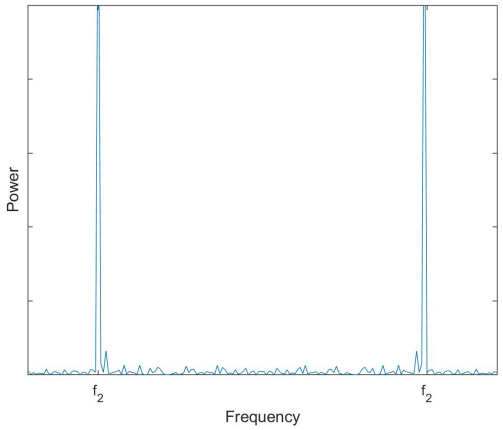


FIGURE 4.2: Power spectral density plot of a $y_r(t)$ signal



(A) Chebyshev filter applied around f_2



(B) Chebyshev filter applied around f_1

FIGURE 4.3: Power spectral density plots after applying Chebyshev filter

$$J_1 = s + i\omega_c \cosh(\frac{1}{2}(\frac{1}{2}(\sqrt{5+4\sqrt{2}}))) \csc(\frac{1}{4}(\pi + 2i(\sqrt{5+4\sqrt{2}}))) \tag{4.4}$$

$$J_2 = s - i\omega_c \cosh(\frac{1}{2}(\frac{1}{2}(\sqrt{5+4\sqrt{2}}))) \sec(\frac{1}{4}(\pi + 2i(\sqrt{5+4\sqrt{2}}))) \tag{4.5}$$

A low-pass to bandpass transformation of $s = \frac{s^2+\omega_f^2}{s}$ is then used, where ω_f is the desired angular frequency to be filtered.

$$L_{bp}(s) = \frac{s^2(\omega_f^4 + \omega_c^2 s^2 + \sqrt{5+4\sqrt{2}}\omega_c^4 s^2 + 2\omega_f^2 s^2 + s^4)}{\sqrt{2(3+2\sqrt{2})}s^2 J_3 J_4} \tag{4.6}$$

$$J_3 = s^2 + \omega_f^2 + i\omega_c \cosh(\frac{1}{2}(\frac{1}{2}(\sqrt{5+4\sqrt{2}}))) \csc(\frac{1}{4}(\pi + 2i(\sqrt{5+4\sqrt{2}}))) \tag{4.7}$$

$$J_4 = s^2 + \omega_f^2 + \cosh(\frac{1}{2}(\frac{1}{2}(\sqrt{5+4\sqrt{2}}))) \sec(\frac{1}{4}(\pi + 2i(\sqrt{5+4\sqrt{2}}))) \tag{4.8}$$

The whole system is then discretized using a bilinear transform, $s = \frac{2(z-1)}{T(z+1}$ to be applied in practical usage.

A power spectral density plot of a generated noisy $y_r(t)$ signal centered around zero can be seen in Figure 4.2. There are two distinct spikes at f_1 and f_2. The Chebyshev filter, as demonstrated through the Bode plot, is capable of removing the frequency information of whatever the zero is chosen to be. This can be seen in Figure 4.3, where the power spectral density of the $y_r(t)$ signal after the filtering is plotted. Figure 4.3a is with a zero at f_2, making it essentially f_{r1}, while 4.3b has the zero at f_1, making it equivalent to f_{r2}.

4.2 The Phase-Locked Loop

The phase-locked loop is an extremely versatile tool that is seen in various communication systems. It plays a vital role for locking on to or synchronizing the instantaneous angle of some external signal and producing a resulting VCO output with that same instantaneous angle.

Effectively, a PLL sees a signal

$$x(t) = \cos(2\pi f_c t + y(t) + \phi_c) \tag{4.9}$$

with carrier frequency f_c. Once this signal is input through the PLL, $y(t)$ is recovered along with phase ϕ_c.

The entirety of the PLL also essentially acts a demodulator in addition to the recovery of the phase. The coherent demodulation scheme used to treat the $x(t)$ signal is described in detail within Chapter 3.3.

4.2.1 PLL Setup

The block diagram for the PLL is outlined as shown.

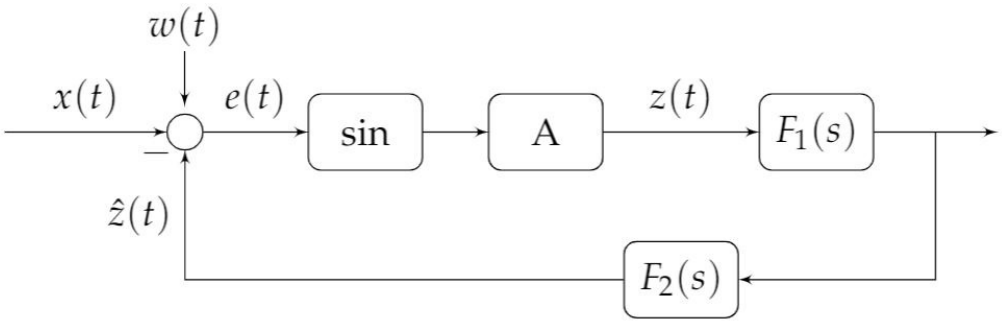


FIGURE 4.4: Block diagram of the PLL

The input signal $x(t)$ is the same modulated signal as (3.6), where

$$x_r(t) = A_1 \cos(2\pi f_c(t - \nu) + y_r(t - \nu)) \tag{4.10}$$

The disturbance $w(t)$, which is equal to $\omega_c d$, is modeled as a disturbance that is to be rejected by the PLL. Ultimately, the signal $z(t)$ is the fully recovered signal

$$z(t) = Asin(x(t) + \omega_c d - \hat{z}(t)) \tag{4.11}$$

The choice of the two filters $F_1(s)$ and $F_2(s)$ will be discussed later.

From the point of view of the receiver, a delayed modulated signal with a carrier frequency f_c is picked up, and this signal, $y_{in}(t)$ is promptly demodulated as the first step in the loop.

$$y_{in}(t) = Acos(2\pi f_c(t - d) + x(t)) \tag{4.12}$$

$$\begin{aligned} y_{in}(t - \frac{\pi}{2})cos(\omega_c t + \phi) - y_{in}sin(\omega_c t + \phi) &= sin(x(t)) - \omega_c d - \phi) \\ &\approx x(t) \end{aligned} \tag{4.13}$$

As seen from (4.13), the $x(t)$ signal is able to be successfully recovered as ϕ converges to $-\omega_c d$. The small angle approximation can be utilized here due to the small amplitude of $x(t)$.

4.2.2 Stability analysis

To ensure that the PLL converges to the desired signal as quickly as possible, various different filters combinations can be looked at. Looking at Figure 4.4, the transfer function of the closed loop system can be modeled as

$$G(s) = (\frac{AF_1}{1 + AF_1F_2})(x - w) \tag{4.14}$$

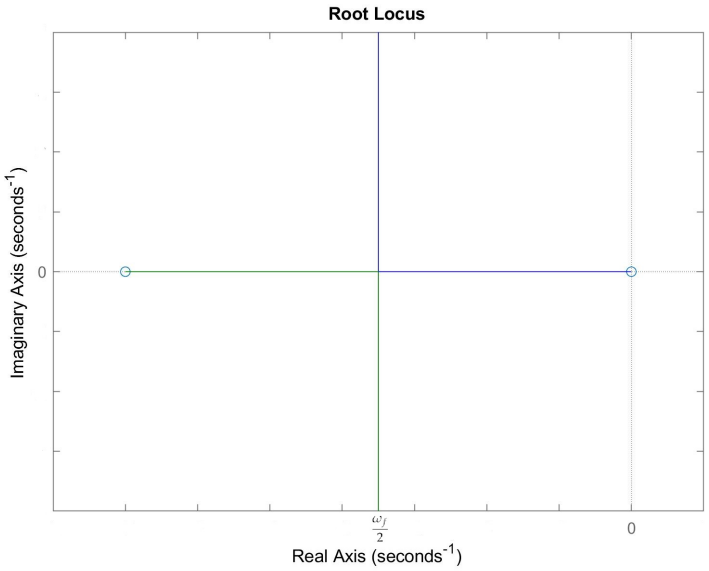


FIGURE 4.5: Root locus plot of $H(s)$

A transfer of such of $F_1(s) = \frac{1}{A}$ is set for the first filter simply just to cancel the amplitude. The second filter $F_2(s)$ is comprised of an integrator, gain, and low-pass filter. This ultimately treats the $\omega_c d$ as an input disturbance and effectively rejects that from $z(t)$, leaving just the desired phase of $\hat{z}(t)$ behind. Complexity in the low-pass filter can be scaled to a higher order if desired, but a first-order filter is capable of doing the job as long as the system is tuned to be optimal. For this filter, $F_2(s) = \frac{K\omega_f}{s(s+\omega_f)}$ is produced, which leads to the resulting closed loop transfer function below.

$$\begin{aligned} H(s) &= \frac{1}{1 + \frac{K\omega_f}{s^2+s\omega_f}} \\ &= \frac{s(s+\omega_f)}{s^2 + s\omega_f + K\omega_f} \end{aligned} \tag{4.15}$$

Quick analysis of the transfer function reveals there to be two zeros at 0 and $-\omega_f$. The poles need to be chosen at $\frac{\omega_f}{2}$ such that the system is stable and optimally

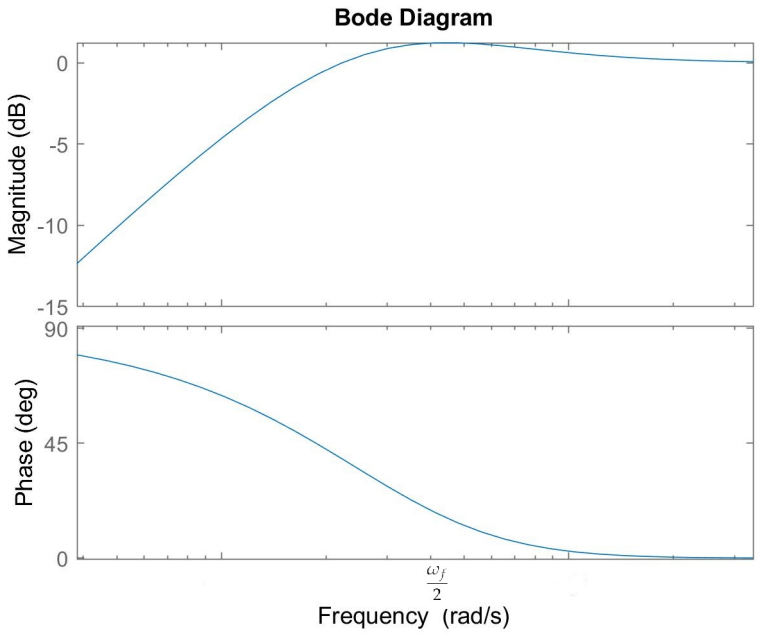


FIGURE 4.6: Bode plot of $H(s)$

converging. By setting the denominator equal to $(s + \frac{\omega_f}{2})^2$, a K value of $\frac{\omega_f}{4}$ is determined. The root locus diagram can be seen in Figure 4.5. The final form of the closed loop transfer function becomes

$$H(s) = \frac{s(s + \omega_f)}{(s + \frac{\omega_f}{2})^2} \tag{4.16}$$

Looking at the Bode plot of $H(s)$, a cutoff frequency at $\frac{\omega_f}{2}$ is successfully achieved.

4.2.3 Envelope Detector

The incoming signal can undergo attenuation from numerous potential sources. To reduce the effects of this attenuation, the signal can be normalized by using an enveloped detector to obtain the amplitude of the signal.

The envelope detector's main component is a low pass filter capable of taking

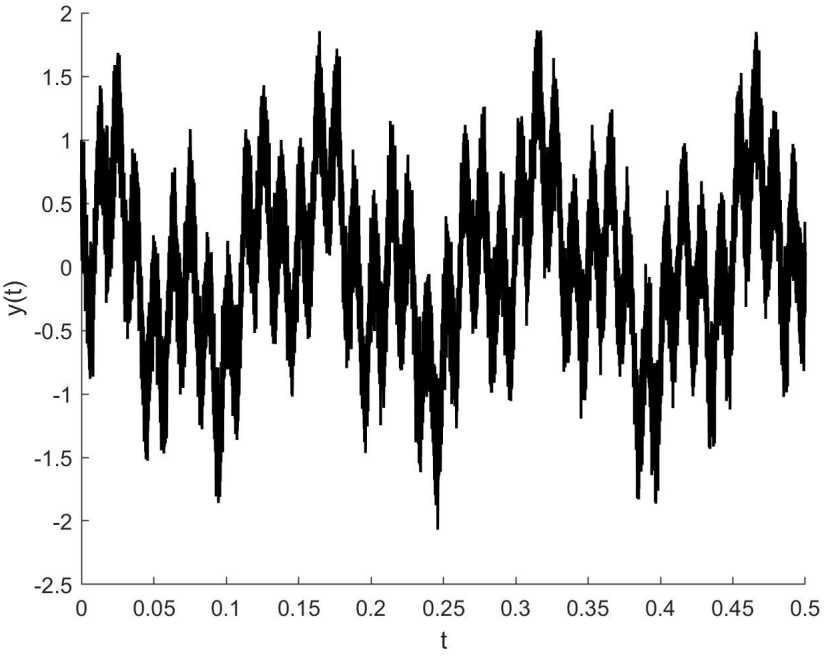


FIGURE 4.7: Randomly generated signal with noise

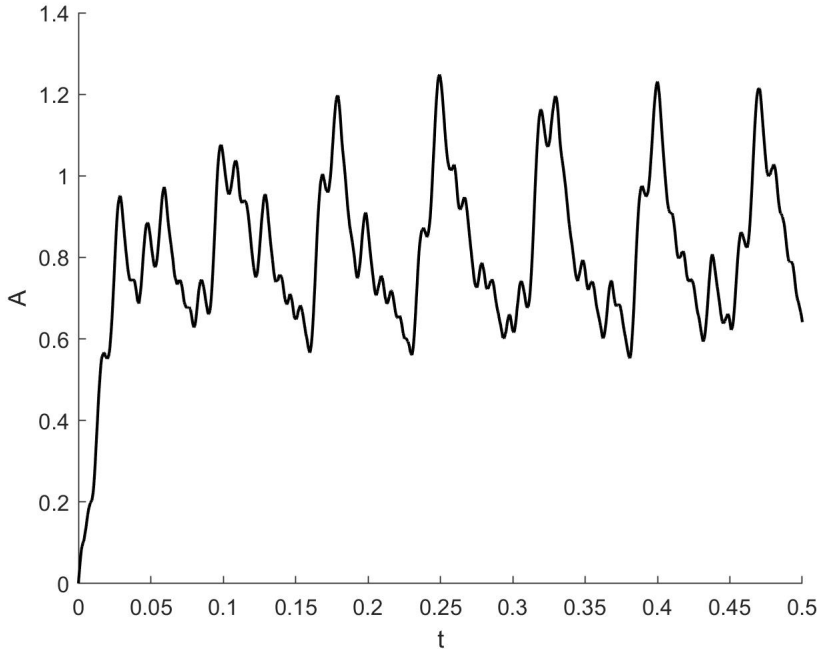


FIGURE 4.8: Envelope obtained from above signal

the high frequency signal and yielding its envelope. Once the amplitude A_e is obtained, the incoming signal can be normalized as such:

$$x_r(t) = \frac{A_e}{A_e^2 + \epsilon} * x_r(t) \tag{4.17}$$

The value of ϵ is taken to be an minuscule value to the order of $1.0 * 10^{-}4$. Effectively, $x_r(t)$ is normalized back to an amplitude of 1 while avoiding division by 0. A example of the envelope detector in action can be seen in Figures 4.7 and 4.8, in which a noisy signal and its resulting envelope can be seen.

4.2.4 Implementation of PLL

As mentioned previously in Section 3.4, the PLL is used once at the start to recover the modulated signal, $x(t)$. This signal is the sum of two waveforms at different frequencies, such that

$$x(t-\nu) = x_1(t-\nu) + x_2(t-\nu) \tag{4.18}$$

$$x_1(t-\nu) = A_1 \cos(2\pi f_1(t-\nu) + \phi_1) \tag{4.19}$$

$$x_2(t-\nu) = A_2 \cos(2\pi f_2(t-\nu) + \phi_1) \tag{4.20}$$

Once $x(t)$ is recovered, the two individual waveforms are each obtained by using the Chebyshev Type II bandpass filter. Both of these signals are then put through another PLL each, demodulating with respect to their own frequencies, f_1 and f_2.

The output of these respective PLL's are such that

$$\begin{aligned} z_1 &= \sin(-\omega d - \psi_1 - (\phi_1 - \phi_2)) \\ z_2 &= \sin(-\omega d + 2\pi\nu\delta - \psi_2 - (\phi_1 - \phi_2)) \end{aligned} \tag{4.21}$$

Eventually, the PLL drives the ψ value such that

$$\begin{aligned}\hat{\psi}_1 &= -\omega d - (\phi_1 - \phi_2) + K_1 2\pi \\ \hat{\psi}_2 &= -\omega d + 2\pi\nu\delta - (\phi_1 - \phi_2) + K_2 2\pi\end{aligned} \tag{4.22}$$

This causes z_1 and z_2 to converge towards zero. These $\hat{\psi}$ values become what are used in the last stage of finding the distance. For practical usage, a PLL Type II is used, in order to handle the noise and demodulate the received signal more effectively [7].

4.3 Distance Estimation Algorithm

Once $\hat{\psi}_1$ and $\hat{\psi}_2$ from (4.22) is obtained, the distance can be thoroughly derived as before in Chapter 1. The key inference from the structure of the phases is with their respective ranges. Both ψ_1 and ψ_2 are restricted between 0 and 2π, but the separation between the them are fixed at $2\pi\nu\delta$. Because δ was chosen for in (3.4) with restrictions based on the maximum range desired, the value of $2\pi\nu\delta$ is comfortably less than 2π as long as ν doesn't exceed the maximum range set.

As stated in 2.9 before,

$$\hat{\psi}_1 - \hat{\psi}_2 = 2\pi\delta_f\nu - \delta_\phi + 2\pi(k_1 - k_2) \tag{4.23}$$

Solving for ν,

$$\nu = \frac{1}{2\pi\delta_f}(\hat{\psi}_1 - \hat{\psi}_2 + \delta_\phi + 2\pi(k_2 - k_1)) \tag{4.24}$$

So now there are two possibilities in the wrapping of the phases. The first is the simple scenario in which both lay within the same period, and K_1 and K_2 are both

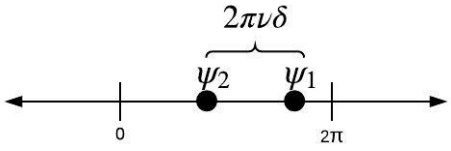


FIGURE 4.9: Phase wrapping, case 1

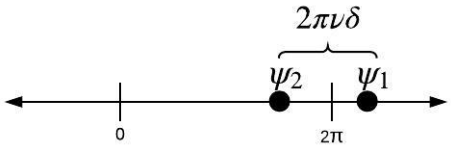


FIGURE 4.10: Phase wrapping, case 2

equal to each other. This occurs when ψ_1 happens to be greater than ψ_2, in which the value between them is established to be $2\pi\nu\delta$.

However, if the two phases happen to be in different periods, then that means ψ_1 has wrapped around to be a smaller value than ψ_2. In this case, ψ_1 is still bigger, but the wrapping causes it to be an order of 2π smaller than it should be. Therefore, K_1 is taken to be equal to $K_2 - 1$ in order to readjust for the wrapping.

Another way to write this process is by taking the modulus of the phase difference with respect of 2π.

$$\nu = \frac{1}{2\pi\delta_f} mod((\hat{\psi}_1 - \hat{\psi}_2 + \delta_\phi), 2\pi) \tag{4.25}$$

Chapter 5

Simulation and Results

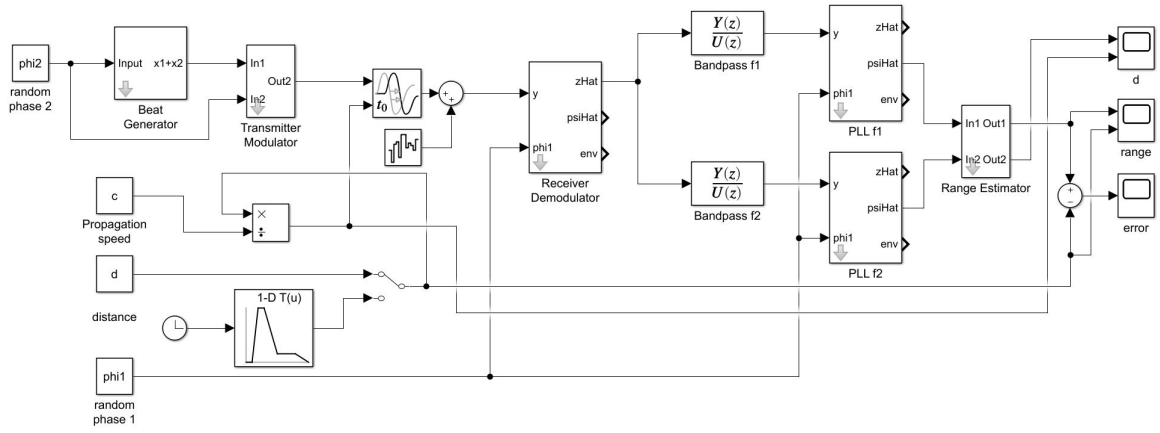


FIGURE 5.1: High level block diagram from Simulink

A model was built in Simulink to demonstrate the effectiveness of the previously outlined principles. The block diagram, as shown in Figure 5.1, contains many of the blocks that have been discussed, such as the PLL and bandpass filter.

5.1 Simulink Layout

From Figure 5.1, the entire process laid out from Figures 3.1 and 3.2 can be seen. All the procedures that take place in the transmitter can be seen on the left side, starting from the beat generator, which produces the signal $y_t(t)$, to the modulator,

which generates the $x_t(t)$ signal. This is all put into a variable time delay block, with a chosen value of ν. The right side of the figure begins with the receiver demodulator, which is essentially a PLL that's exactly the same as the "PLL f1" and "PLL f2" blocks seen to the right. This PLL block can be configured for a unique carrier frequency f_c for demodulation purposes.

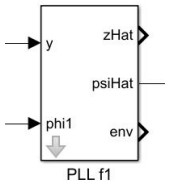


FIGURE 5.2: PLL block

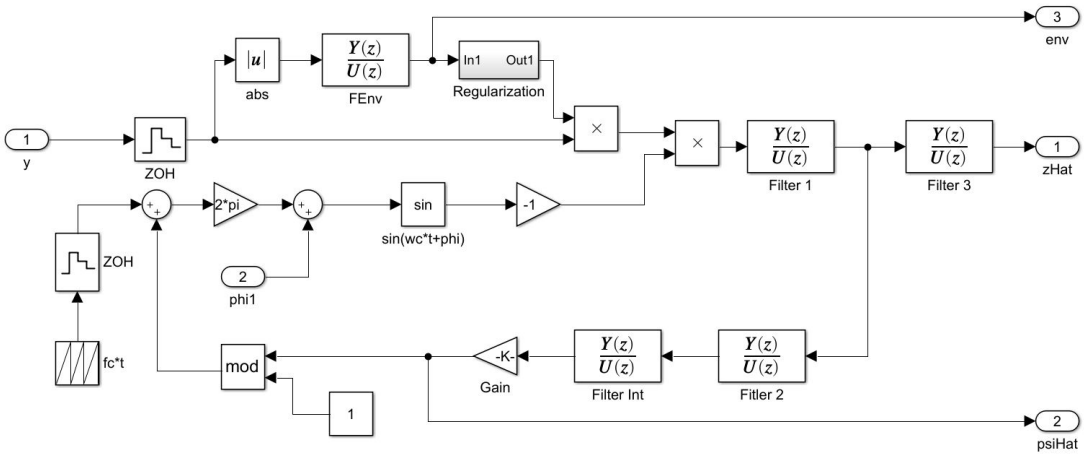


FIGURE 5.3: Inside of the PLL block

The outputs of the PLL block include the demodulated signal, $z(t)$, and the signal phase, ψ. The phase values are what's ultimately taken to be used inside the range estimator block, as shown below.

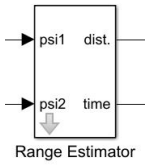


FIGURE 5.4: Range estimator block

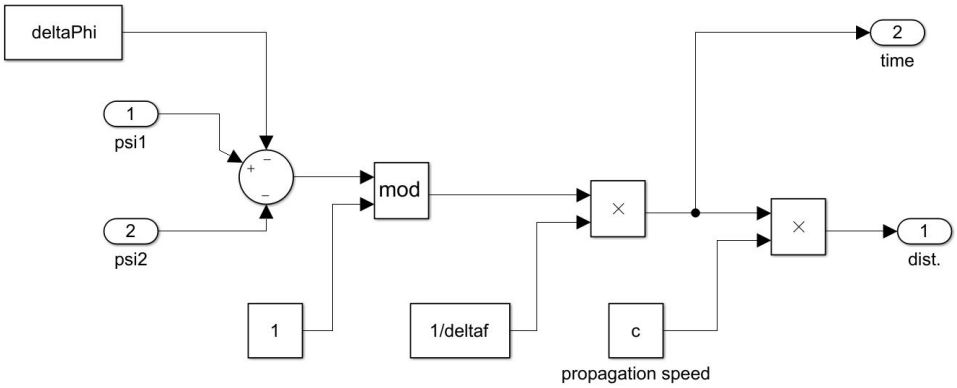


FIGURE 5.5: Inside of the range estimator block

The calculation for the range estimator is examined in detail in Chapter 4.3. The modulus operator is used here in order to account for the wrapping, and it is taken with respect to 1 in order to normalize the value.

5.2 Simulation Results

Numerous simulations were performed with randomly chosen parameters for ν, ϕ_1, and ϕ_2. Filter parameters were finely tuned to optimize performance of the system. A distance range of 0.1m to 50m was tested, and some of the results can be seen below.

Figures 5.6 and 5.7 display two different delays of 0.09 seconds and 0.044 seconds, which are about 30 and 15 meters respectively. Convergence to a steady state

value is achieved no slower than 0.4 seconds in either of the simulations. This settling time is caused mostly by the dynamics within the PLL, which takes some time in order to lock on to the correct phase. The delay estimation starts at $\frac{1}{\delta}$, which is roughly around 0.122 for a δ value of 8.19. As shown by (3.4), this initial value is the maximum delay value possible.

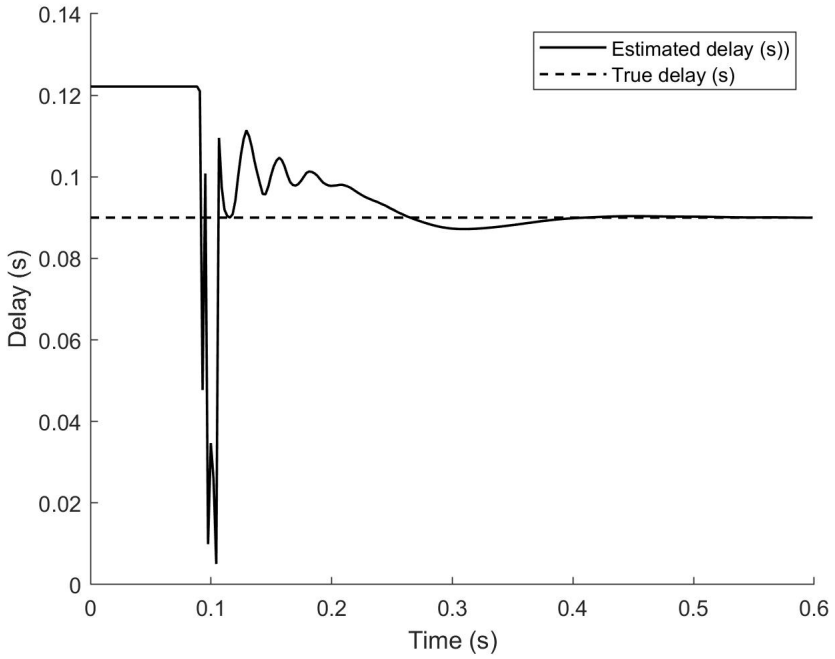


FIGURE 5.6: Simulated system at ν = 0.09 s

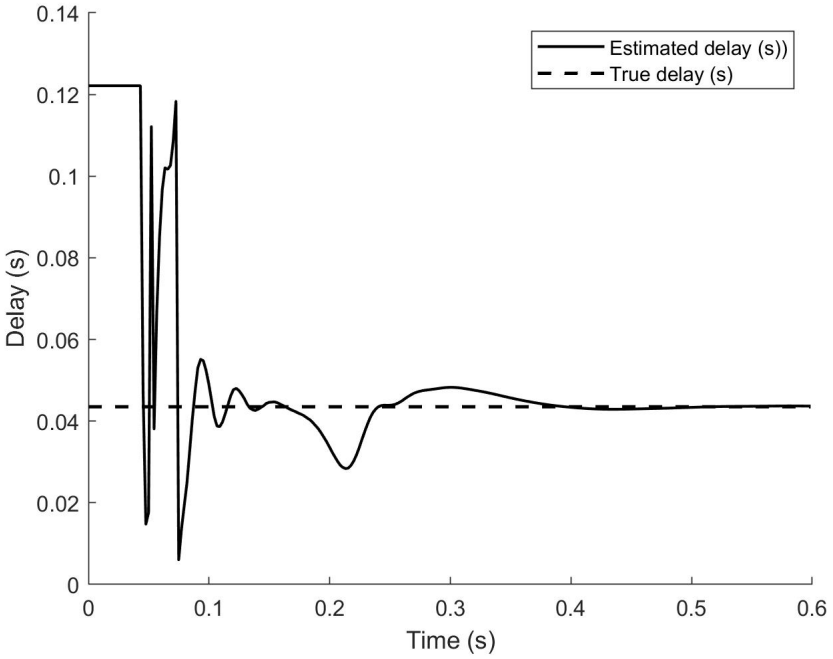


FIGURE 5.7: Simulated system at $\nu = 0.044$ s

Once white Gaussian noise is added to the signal as seen in Figure 5.8 and 5.9, the performance is mostly the same. Even after the noise power is increased by ten times in the latter figure, there aren't any significant offsets in the estimated values.

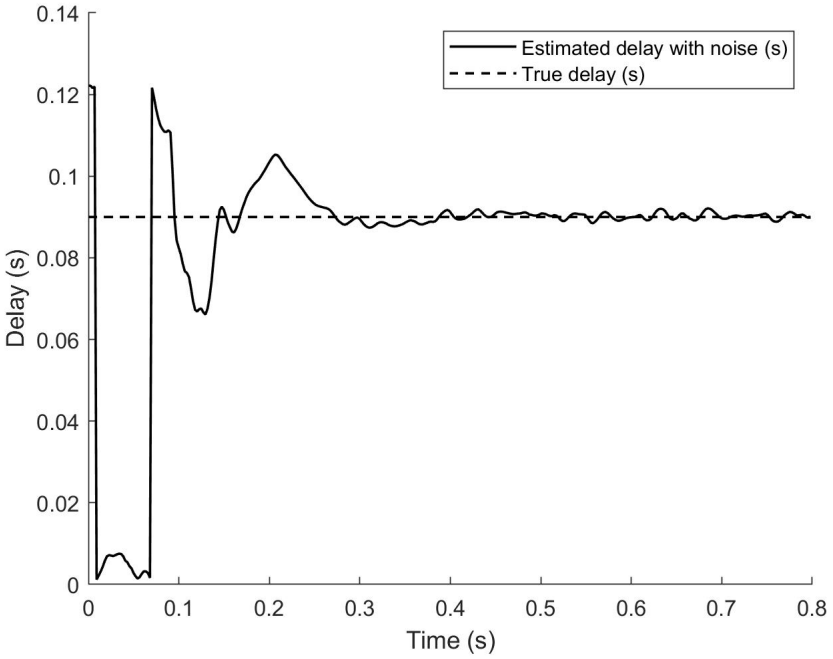


FIGURE 5.8: Effects of noise applied on system, $\nu = 0.09$ s

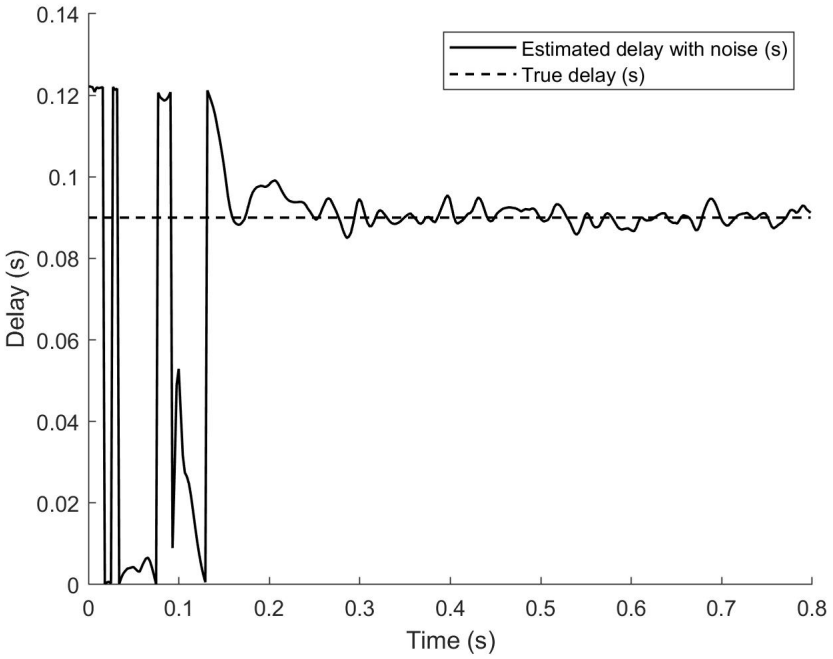


FIGURE 5.9: Even stronger noise applied, $\nu = 0.09$ s

Chapter 6

Pure Data Implementation and Results

This chapter briefly outlines the steps taken to effectively realize the entire system in Pure Data, a graphical programming environment designed by Miller Puckette. Pure Data can operate as a DSP and meet any functional requirements that are needed.

Pure Data uses various object blocks in order to build complex signal systems. These object blocks consists of a myriad of tools, such as basic oscillators to first-order banpdass filters [11]. For example, Figure 6.1 shows a simple signal at 200 Hz being constructed, subsequently mutlipled by an amplitude of 0.5, and then sent to a digital to audio converter.

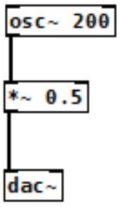


FIGURE 6.1: Example of a Pure Data system

A major constraint of Pure Data is the inability to operate in closed loop systems, such as the PLL. Therefore, in order to design this system in Pure Data, an *external* must be created. Externals are custom made objects that are written in C to fulfill any requirements in Pure Data that doesn't currently exist.

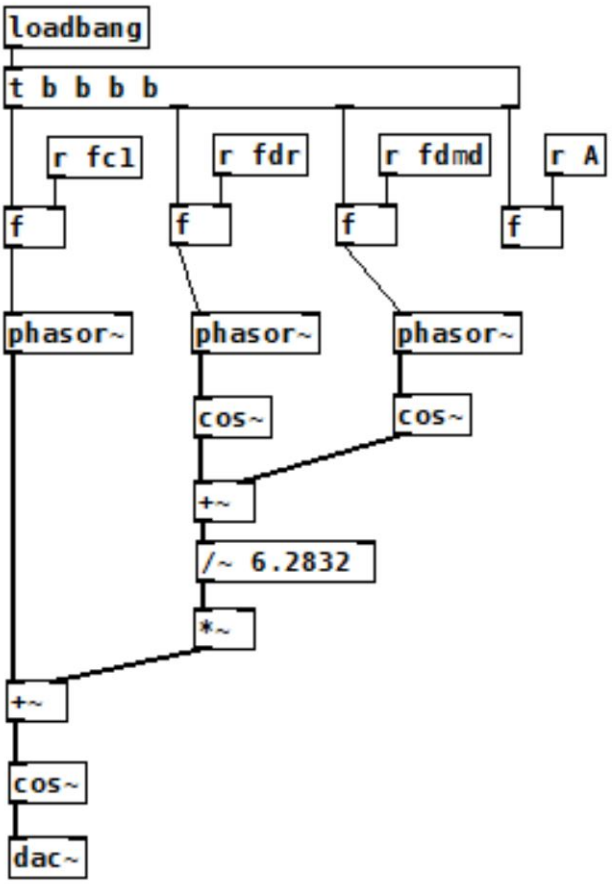


FIGURE 6.2: First half of Pure Data system, the transmitter

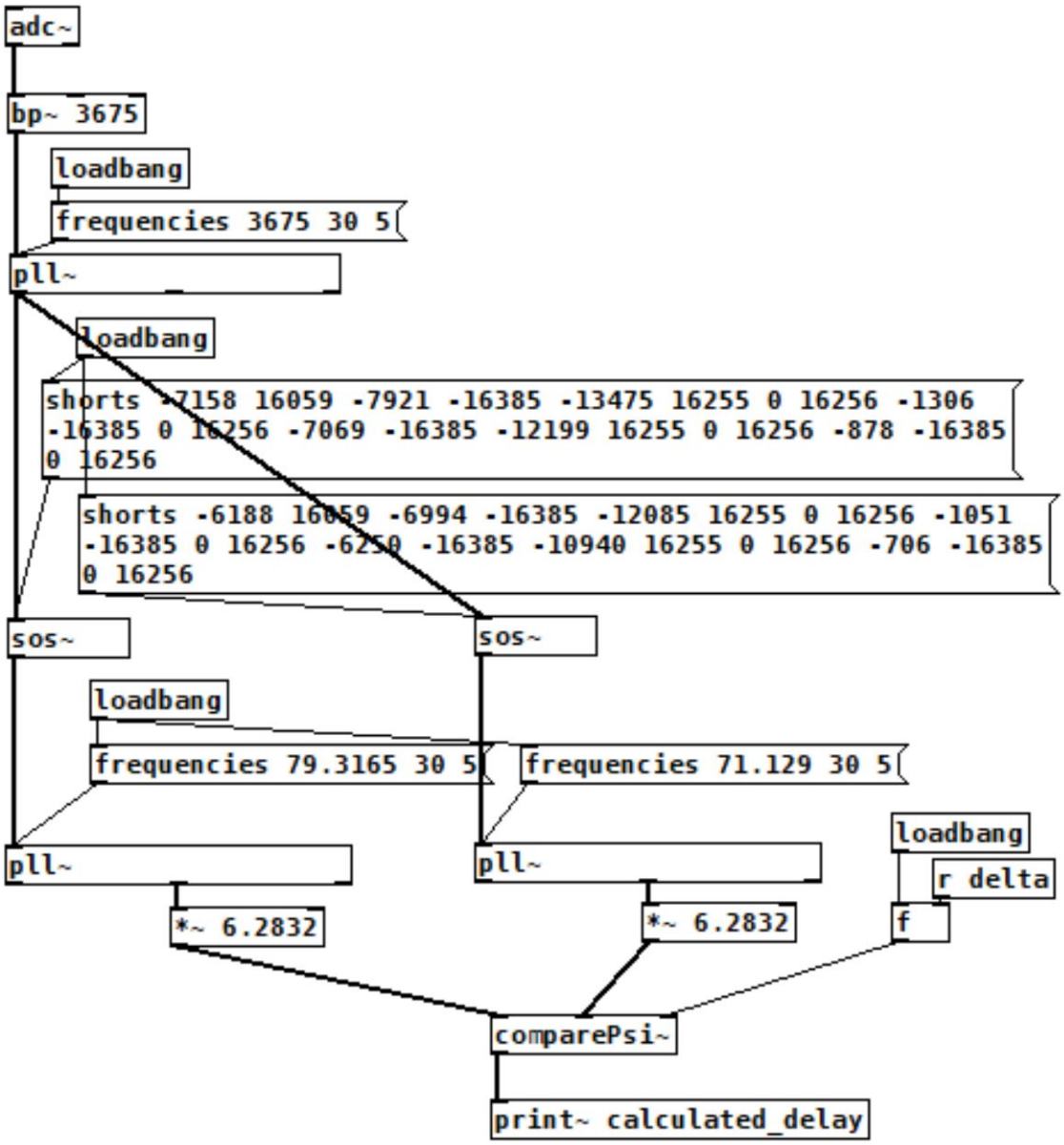


FIGURE 6.3: Second half of Pure Data system, the receiver

The entire Pure Data system can be seen in Figures 6.2 and 6.3. The first figure consists of the transmitter, in which the signal is generated, and the second figure is the receiver, which consists of three PLLs and two bandpass filters.

Experimentation was done through a simple microphone and speaker system. The two pieces are synchronized through Pure Data, such that the signal phases and

the PLL internal phases are all initialized at the same time. While the system is designed to be ultimately transmitted in the ultrasonic range, the testing was done in a frequency that the human ear can hear. There is a lot of room for improvement, as the hardware had a lot of limitations. The microphone used was from a laptop's internal microphone, and the speaker was a wireless bluetooth speaker.

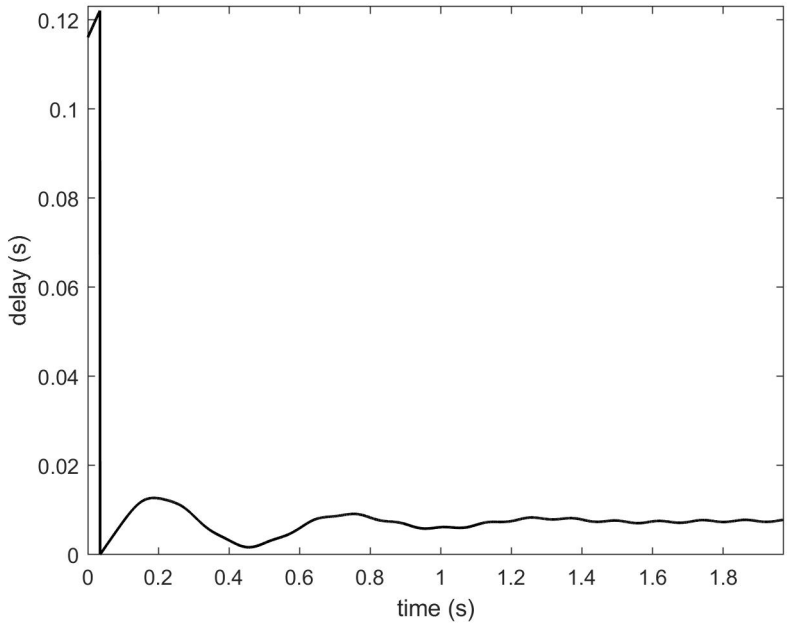


FIGURE 6.4: Pure Data delay at 2 meters

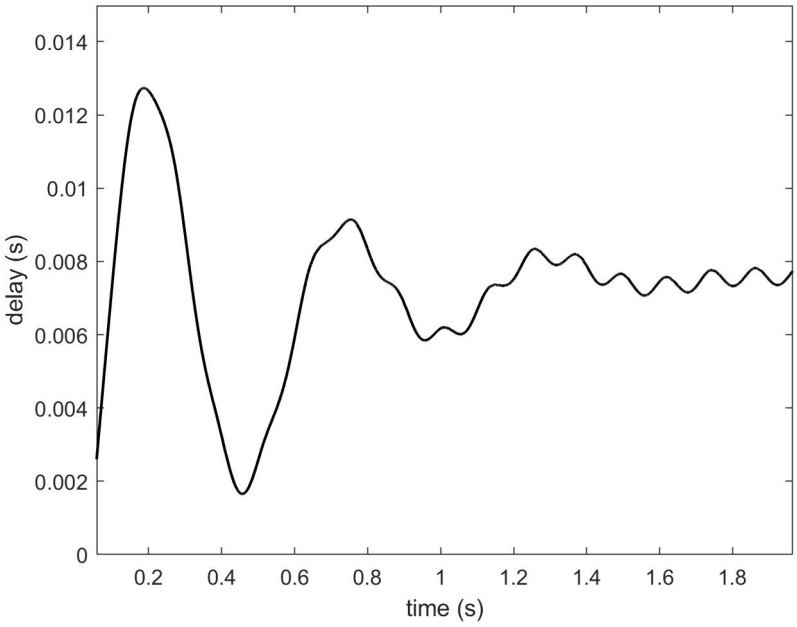


FIGURE 6.5: Pure Data delay at 2 meters zoomed

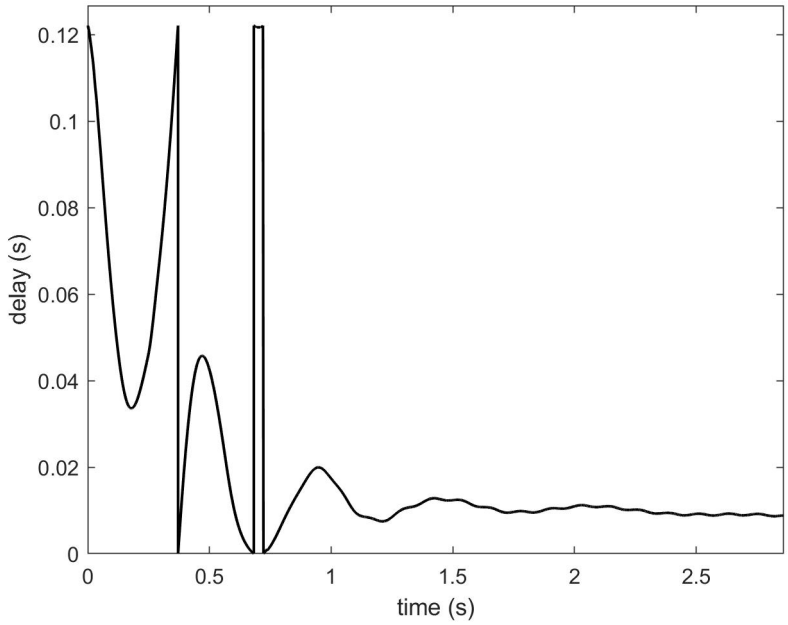


FIGURE 6.6: Pure Data delay at 3 meters

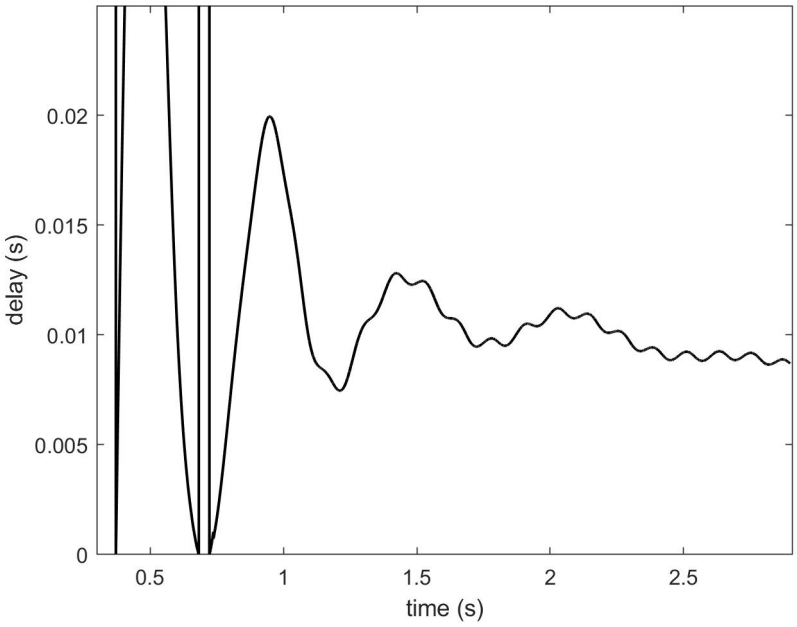


FIGURE 6.7: Pure Data delay at 3 meters zoomed

Examples of two separate runs from 2 meters and 3 meters can be seen in Figures 6.4-6.7. Similar to the simulations, the system takes a bit to converge, but the steady state values are close to the true delay value. The propagation speed of sound was not measured during these measurements, so an estimate of 340 m/s is used. Judging by this estimate, the 2 meter run is off by about 0.001 seconds, which translates to 0.34 meters, and the 3 meter run is off by about the same margin as well. Multiple runs were conducted at 2 and 3 meters, and the results were mostly comparable.

However, transmission at speeds at longer ranges were difficult to pick up using the speaker and microphone. Even when a large gain was imposed, the microphone and the filters had a hard time deciphering the faint signal. For the ranges in which the signals were coherent, the noise was almost completely filtered out as expected.

Chapter 7

Discussion

The proposed distance estimator can be used as a component in a multitude of systems that can be seen in communications, such as oscillators, modulators, phase detectors, and filters [7]. The implementation can be achieved with analog circuits or, depending on the target frequency range, using entirely digital algorithms.

Since the original signal, $y_r(t)$, is being frequency modulated, the transmission and subsequent recovery of the signal is largely unaffected by noise. The phase data is always preserved regardless of any amount of attenuation that occurs as well. In practical usage, the environment in which the signal is transmitted becomes the largest potential source of error. If done indoors, reflections of the signal from various surfaces may compromise the integrity of the signal. These reflections can be modeled as an infinite sum of the main tone, in which each proceeding tone gets increasingly attenuated. Altogether, this acts as a linear filter, altering the performance by yielding a received signal of the same frequency but with a different phase and amplitude. Another interesting note is that because the two y_1 and y_2 signals are so close in frequency, the environment can be expected to have the same effects on both frequencies.

In comparison to some previously listed methods of distance estimation, such as TDoA or ToA, this approach has both similar and unique characteristics that are

equally beneficial. Due to the composition of the system, the range is passively measured. This means that the outgoing signal doesn't have to be sent back in order to make an estimation. In addition, the entire system operates under continuous-wave (CW) rather in pulses. This allows for uninterrupted measurements of target information with a greater resolution [7].

This system was designed to be transmitted using ultrasonic signals, which have various advantages than can be taken advantage of. In practice, ultrasonic sensors are employed in a myriad of electronics, such as robotic navigation, anti-collision systems, and medical diagnostic tools [6]. Within the context of proximity measurement, ultrasonic signals are not affected by the hue or transparency of the objects it reflects off. In addition to this, acoustic waves don't rely on light, so they are able to function perfectly in dark or dimly lit settings as well. However, sound is directly correlated to the temperature and humidity of the medium it travels through, so ultrasonic sensors must constantly be re-calibrated whenever its environment changes. Its dependency on the medium also causes ultrasonic signals to be ineffective in a vacuum and even underwater, depending on how well it's designed. Range for acoustic sensors is also lacking in comparison to other sensors such as radar or lidar, but ultrasonic sensors are never chosen in design practice for range capabilities.

Acoustic waves are just one of many potential propagation methods that can be used. This principle is universal and can be applied with an assortment of options, such as electromagnetic waves or pressure waves, with the same expected effectiveness.

If this is used in conjunction with multiple other transmitters and receivers, a sensor network can be formed to achieve localization. In order for a node inside a wireless sensor network (WSN) to localize itself in 2D space, it needs to know the distance from three separate reference points [8]. Because only one receiver/transmitter pair is

required to calculate the distance, localization can be achieved with a minimal amount of sensors.

This principle can also be expanded to include moving targets or sources as well. Currently, the phase detector and range estimator is designed for both the receiver and transmitter to be stationary. However, if the Doppler shift and additional sources of noise is taken into account, any movement can potentially be adjusted for.

Bibliography

[1] Z. Nemec M. Mandlik and T. Vankat. "Real-Time Ultrasonic Localization Using an Ultrasonic Sensor Array". In: (2012), pp. 106–111.

[2] Miklós Maróti, Péter Völgyesi, and Sebestyén Dóra. "Radio interferometric geolocation". In: *Proceedings of the SenSys'05*. San Diego, California, USA., Nov. 2005.

[3] S. Kumar D. Vasisht and D. Katabi. "Decimeter-level localization with a single wifi access point". In: (2016), pp. 165–178.

[4] Isaac Amundson and Xenofon Koutsoukos. "A Survey on Localization for Mobile Wireless Sensor Networks". In: (2009). Ed. by R. Fuller and X.D. Koutsoukos, pp. 235–254.

[5] Jeril Kuriakose, V. Amruth, and A. G. Sandesh. "A Review on Mobile Sensor Localization". In: *Security in Computing and Communications* (Jan. 1, 2014). DOI: 10.1007/978-3-662-44966-0_4.

[6] Dimitrios Lymberopoulos and Jie Liu. "The Microsoft Indoor Localization Competition: experiences and lessons learned". In: *IEEE Signal Processing Magazine* (Sept. 2017), pp. 125–140.

[7] Janet Rutledge A. Bruce Carlson Paul B. Crilly. *Communication Systems*. McGraw Hill Higher Education, 2001.

[8] A. Varga. "Localization Techniques in Wireless Sensor Networks". In: (2013), pp. 81–90.