Exploring Radio-Frequency Localization for Pedestrian Safety

Justin Goodman
University of Maryland, College Park
jugoodma@umd.edu

Nirupam Roy
University of Maryland, College Park
niruroy@umd.edu

ABSTRACT

Recent work has shown that localizing signal sources is not only possible, but relatively accurate on commodity hardware. With a theme of better informing autonomous vehicle decision-making, we introduce a full localization system that takes advantage of these recent techniques. Then, we present a proof-of-concept mock-up using software-defined radios and discuss design choices to realize this system in full.

1. INTRODUCTION

As human drivers become replaced by assisted and autonomous cars, passenger and pedestrian safety becomes a major concern. Self-driving cars are magnets for philosophical discussions, and ripe for legal issues. Even though self-driving cars are arguably safer than human-driven cars [2, 14], public opinion is highly swayed by autonomous car accidents [4, 5]. In order to make autonomous vehicles more ubiquitous, we have to work harder to improve the human safety factor. One way we can do this is by increasing the information available to an autonomous car such that it can make smarter, and life-saving, decisions faster.

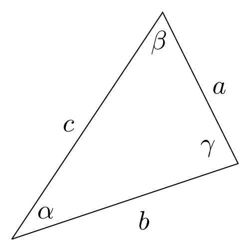
In this paper, we combine approaches outlined in recent literature to enable communication between pedestrians and self-driving cars. A pedestrian, or stationary object (e.g.: a stop sign), can wear a low-cost and low-power radio-frequency (RF) tag that emits a continuous signal to the world. An autonomous car, with the right equipment, can read this signal and triangulate the 2D position of the transmitter. This localization information, in tandem with its computer vision system, can significantly improve the car's spatial awareness. The car can be even more informed if the signal carries identifying information (for example, the road sign name and type).

In this paper, we present a theoretical method for realizing 2D localization. We make use of orthogonal frequency-division multiplexing (OFDM), angle-of-arrival (AoA) estimation techniques, and simple geometry and trigonometry to fully localize a target. We base our AoA estimate using SpotFi [9], which improves on the well-known MUSIC algorithm [12]. While our methods are theoretical, we provide a proof-of-concept implementation on universal software radio peripherals (USRPs) to inform design choices for future work, where we hope to fully realize this system.

The rest of the paper is organized as follows. Section 2 goes over background necessary for the rest of the paper. Section 3 formalizes the problem statement, and section 4 goes over the theoretical solution. Sections 5 and 6 go over our proof-of-concept system. Finally, we discuss future considerations and conclude the work in sections 7 and 8.

2. BACKGROUND

Triangulation is where we use geometric properties of triangles to compute the location of an unknown point based on locations of known points. There are six key values that completely describe a triangle – its three interior angles, and its three perimeter edges. Given three of the six values, depending on the types, we can compute the other missing three values. Recall the law of sines, given in figure 1, which relates a triangle's side lengths to its angles. Further recall that a triangle's



$$\frac{\sin\alpha}{a} = \frac{\sin\beta}{b} = \frac{\sin\gamma}{c}$$

Figure 1: Law of Sines

interior angles must sum to π radians, thus given two of the three angle measures we can compute the missing angle measure. Given two angle measures and one side length, we can compute the missing side lengths with the law of sines. Without loss of generality, the missing side lengths b and c are given by

$$b = a \cdot \frac{\sin\beta}{\sin\alpha}$$
$$c = a \cdot \frac{\sin\gamma}{\sin\alpha}$$

Given two side lengths and one angle measure, we can also compute the missing side length and missing angle measures with the law of sines. We omit this derivation, since we will not use it in this paper – though we note that the known angle measure must be opposite of a known side length, and we generally yield two possible solutions in this case. Given three side lengths, we can use the law of cosines to solve for the three missing angles. This will not be useful for our case, so we omit showing it here. Note that given three angle measures, we cannot solve for the missing side lengths since there are infinitely many similar triangles (all ratio-scaled) with the same three angle measures.

OFDM [1] has become the primary method for signal transmission in numerous domains, like Wi-Fi [7, 13] and cellular networks [6, 16]. It boasts numerous benefits, like robustness to channel effects and inter-symbol interference. In short, OFDM works like traditional frequency-division multiplexing, but requires each frequency bin to be orthogonal to each other. This eliminates subcarriers affecting other subcarriers (crosstalk). OFDM packets start with a preamble sequence, known by the transmitter and receiver, followed by multiple FDM data symbols. The decoder uses the preamble for packet identification and channel estimation, and updates this channel estimate while decoding each symbol. There exist tools to obtain the channel estimate, called the channel state information (CSI), of Wi-Fi signals [3, 18]. These tools do not include CSI for each data symbol. Symbol-level channel estimate granularity may be beneficial though, especially in high-precision methods, so we use an in-house OFDM transceiver.

Channel estimation is the process of estimating how the wireless channel, the environmental path taken by the signal, affected the signal. The wireless channel can alter a signal with phase shift and attenuation, which can be encoded into a single complex number. As such, the channel estimate is just a complex number per OFDM subcarrier. This results in a $K \times 1$ matrix H where K is the number of subcarriers. OFDM includes guard bins, for which the associated subcarrier frequency has zero amplitude. These bins are not included in the channel estimate. The IEEE standard [7] gives the subcarriers for which data is sent depending on the channel bandwidth. Our implementation uses more guard bins than the standard, so we obtain channel estimates for 52 subcarriers.

The channel estimate, paired with multiple receiver antennas, allows for AoA estimation. AoA works by estimating the phase shift of a signal per antenna, and computing the sent-signal angle that most likely yielded this set of phase shifts. Phase shifts are time-dependent, and electromagnetic signals travel at a (mostly) constant speed. As long as we know the distance between our antennas, we can compute the AoA. For simplicity, we arrange our antenna array along a horizontal line, and equally space each antenna with distance $l = \frac{\lambda}{2}$ with λ being the receiver's center frequency wavelength (figure 2). This technique is well-known and well-documented,

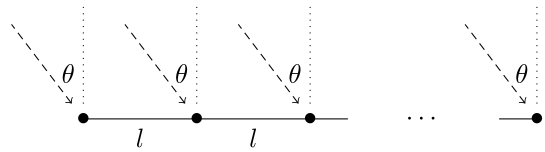


Figure 2: The received signal (dashed) angle θ, relative to the vector perpendicular to the array, is approximately the same per-antenna (dot).

and is essentially the reverse of beamforming [17].

Recent developments in the literature have showed that we can use the channel estimate to perform AoA estimation. The MUSIC algorithm estimates AoA by separating out the noise in a received signal, and computing the AoA from the resulting signal. However, this requires the number of antennas to be greater than the number of signal propagation paths. SpotFi improves on MUSIC by "smoothing" the channel estimate. It uses structurally similar combinations of subcarrier channel estimates per antenna to create virtual antenna estimates. This increases the amount of antennas to typically be greater than the number of propagation paths. Then, MUSIC is applied to the result, and we yield relatively accurate angle estimates at a computational cost still bounded by MUSIC. We can apply these methods to yield a complete localization system.

3. PROBLEM STATEMENT

Consider a moving vehicle with an array of M RF antennas listening at center frequency f equispaced at $l = \frac{\lambda}{2}$ meters where $\lambda = \frac{c}{f}$ is the wave length of the center frequency and c is the speed of electromagnetic waves. Suppose there is a stationary low-powered RF device transmitting a signal at the same center frequency as the vehicle. Suppose that the vehicle's velocity v_i is known at any point in time i. We aim to localize the position, in 2D, of the transmitter with respect to the vehicle.

4. THEORY

Our approach to the problem takes advantage of geometric triangulation. Suppose we take two measurement of the angle between a transmitter and a receiver array, and we know the distance between the measurements (figure 3). Given these data, since a triangle

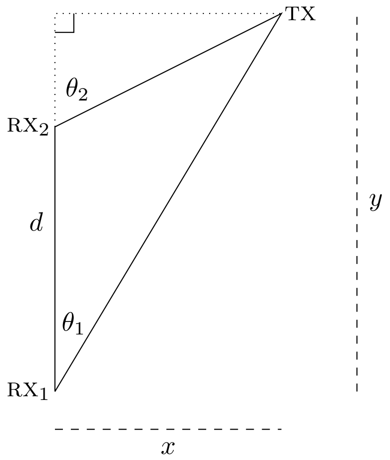


Figure 3: Triangulation.

is entirely determined by two angles connected by one side, we can compute the location (x, y) of the transmitter relative to RX_1. Using straightforward applications of the law of sines and trigonometry,

$$x = d \cdot \frac{\sin(\theta_2)\sin(\theta_1)}{\sin(\theta_2 - \theta_1)} \quad (1)$$

$$y = d \cdot \frac{\cos(\theta_2)\sin(\theta_1)}{\sin(\theta_2 - \theta_1)} + d \quad (2)$$

Since our angle measurements are symmetric, we have the four potential solutions $(\pm x, \pm y)$. AoA techniques yield angles within a π-radian range respective to the horizontal line formed by an antenna array. If we place the receiver array on our vehicle such that the array's horizontal line is perpendicular to the vehicle's direction of motion, we can eliminate the two solutions $(-x, \pm y)$. It suffices to differentiate transmitters that are behind the vehicle versus in front of the vehicle. For simplification, we can assume all transmitters will be in front of the car. This likely will not work in practice though, so we discuss potential solutions later.

By using the methods developed in SpotFi, we can compute the angle of arrivals θ_1, θ_2 while the vehicle moves from position RX_1 to position RX_2. SpotFi also yields an estimate of the time of flight, however this is not necessary to solve our localization problem. The time of flight, though, can be used to compute distance, which can also be used for triangulation. We can use these different methods to potentially reduce error, however we leave this for future work. Since the vehicle will be moving at some velocities v_1 and v_2, Doppler shift will affect the received signal at the respective receiver positions. Note that Doppler shift has been documented in OFDM, and there exist techniques to account for it [8, 10, 11, 15, 19].

In theory, if two antennas have a non-zero distance between them, and both antennas are time synchronized, then it should be possible to estimate AoA from the time difference between the signal reaching one antenna before the other. Consider, though, that for a 20MHz bandwidth we have a 20MHz sampling rate. Then, given a distance l meters between two antennas, we can compute the sampling offset between packets as

$$s = \frac{20 \times 10^6}{c} \cdot l$$

For example, if $l = 6.25$cm then we expect there to be < 1 sample between packets. This is not nearly enough granularity to estimate AoA accurately, which is why we have to use phase shifts in the channel estimate. Though, this gives us a theoretical guarantee that packet i will be received within a few samples on each antenna. This ensures that we are feeding correctly aligned packet channel estimates into SpotFi.

So, how many OFDM packets can we collect at a particular instant in time? Since the vehicle moves continuously, we have to descritize the recording domain to some distance d_i. The vehicle moves at v_i meters per second at a particular instant, so moving d_i meters takes $\frac{d_i}{v_i}$ seconds. Each OFDM packet consists of a preamble, and some amount of data symbols. Recall that we get a channel estimate from the OFDM packet preamble and an estimate per data symbol. Thus, for simplicity, we compute the number of data symbols we can collect within a particular distance interval. If an OFDM data symbol takes 4 microseconds, our recording domain is 10 centimeters (the localization accuracy of SpotFi), and our vehicle velocity is 25 kilometers per hour, then we can collect about 360 data symbols per antenna. Each data symbol (from all antennas) yields an associated AoA estimate, so equivalently we can collect about 360 AoA estimates. This is more than enough to gather a feasibly accurate AoA estimate. However, the number of collected symbols is dependent on vehicle speed and recording domain precision. If the vehicle travels at a higher speed, the number of collected symbols will necessarily decrease. For example, if instead the vehicle travels at 50 kilometers per hour, then we can collect about half as many AoA estimates – though we can account for this by reducing the precision in half (double the distance between recordings). Furthermore, channel estimates on OFDM data symbols are less accurate than those used to estimate the OFDM packet preamble. As such, our number of AoA estimates could decrease by about a factor of 10 (if we only consider the channel estimates from preambles). Still, this is more than enough to yield good AoA estimates.

Altogether, we present a theoretical localization routine in algorithm 1.

```
Algorithm 1: Localization
  Data: vehicle RX array samples stream
  Result: x, y location of TX
  θ1, θ2 ← 0
  while true do
      θ1 ← θ2
      H ← 0
      Δ ← 0    // vehicle's traveled distance
      while Δ < d do
          detect/align OFDM packet in each array
          account for Doppler shift
          decode packets
          if bit-error rate < ϵ then
              append channels to H
          end
          update distance Δ
      end
      θ2 ← AoA from SpotFi using H
      Localize(θ1, θ2, Δ) with parallel processor
  end

  Function Localize(θ1, θ2, d):
      // localize according to (1) and (2)
      x ← d · sin(θ2) sin(θ1) / sin(θ2 − θ1)
      y ← d · cos(θ2) sin(θ1) / sin(θ2 − θ1) + d
      inform vehicle of (x, y)
  end
```

5. SYSTEM DESIGN

To test our theoretical results, we set up a prototype hardware configuration and software analysis scripts. For the hardware setup, we used three universal software radio peripheral USRP devices synchronized by an OctoClock to act as the receiver array. We installed the three receiver antennas onto a laser-cut piece of wood with mount-points arranged in a horizontal line with spacing 6.25cm, corresponding to our center frequency of 2.4GHz. We used a fourth USRP device, unsynchronized with the first three, as a transmitter. Each USRP device was a USRP-N210 with SBX Daughterboard to facilitate a 20MHz bandwidth channel. Each radio was connected to a gigabit Ethernet switch, connected to a computer. For the software, we used MATLAB. Our source code is available on GitHub[1].

6. RESULTS

Our goal here was to create a proof-of-concept of our theoretical design. In our testing, we sent one OFDM packet continuously with a few hundred samples of spacing between each packet. The packet was the same in each test, and is presented in figure 4. Our tests consisted of recording the sent signal synchronously from all three receiver radios for 20 to 30 seconds. Due to how MATLAB interfaces with USRP radios, the first 15 seconds (roughly) would consist of MATLAB establishing a connection to all four radios. Then, the receiver radios had to amplify their local gains, and the transmitter had to start sending the signal. All in all, this meant that only after about 17 seconds would we get consistent data. In the future, we hope to establish known, precise, and quantitative time points for which the transceiver system functions correctly. For now, our tests provide enough insight for a simple proof-of-concept.

We present our testing data in figures located in the appendix. Note that SpotFi yields both AoA and time-of-flight (ToF) estimates. We present both, although theoretically we only need AoA for our localization system.

7. DISCUSSION

We hypothesize two sources of error in our testing. First, there was not enough distance between the transmitter and receiver array. Likely, results can be unpredictable at low ranges, so in the future we recommend testing in the tens of meters of distance. Second, likely there is error in the communications between MATLAB and the radios. It was unclear to us exactly *when* specific code executed. As such, we recommend implementing this system in native `c++` using the UHD API[2]. Furthermore, we recommend running the transmitter code on a separate machine than the receiver code.

Interestingly, channel estimation throughout the received packets seemed to yield consistent results for ToF estimation, but varied for AoA estimation. While we do not know why this happens, we note that channel estimation from the OFDM packet preamble typically yielded good results. We leave per-symbol channel estimation to future work. Likely, it would be fruitful to shorten the OFDM packet data while lengthening the preamble. Or, potentially create a data encoding scheme that incorporates a known transmission signal pattern.

In line with informing vehicle decision-making, we note here that OFDM packets carry bit information. This can prove useful in dynamic driving conditions. Suppose a road is undergoing construction work, and one lane is temporarily closed. If a low-cost RF transmission tag were to be placed on a traffic cone, it could alert an autonomous car of the new conditions. Furthermore, if a construction *worker* is wearing one of these tags, then

[1] https://github.com/jugoodma/rf-loc

[2] https://files.ettus.com/manual/index.html

our vehicle may be able to avoid a collision if the worker were to accidentally miss-step onto the road.

Finally, due to the symmetry of the receiver, we are not guaranteed to resolve transmitters in front of versus behind the receiver. This situation would occur when a car has just passed, and is now moving away from, a transmitter tag. One potential solution is to have two parallel systems – one array perpendicular to the vehicle's direction, and another array parallel to the vehicle's direction. The two arrays would need to have the same center, but this would resolve both 2D directions. Alternatively, we can have a circular array of antennas, which may be a better solution. Finally, it may be possible to use Doppler shifting. When the receiver moves away from a transmitter, then Doppler shift reduces the signal frequency. Since the frequency will only go up or down (depending on if the receiver is moving towards or away from the transmitter), then we can account for both and test which one is correct (maybe the bit-error rate, or a checksum).

8. CONCLUSION

In this work, we have described a theoretical OFDM transceiver system that exploits current literature to enable communication between a self-driving car and a pedestrian. We also provide an introductory code base for implementing this work, and used preliminary data to inform future design choices. We believe work like this is crucial to improving safety in a future age of autonomous vehicles.

REFERENCES

[1] Chang, R. W. Synthesis of band-limited orthogonal signals for multichannel data transmission. *The Bell System Technical Journal 45*, 10 (1966), 1775–1796.

[2] Greenblatt, N. A. Self-driving cars and the law. *IEEE spectrum 53*, 2 (2016), 46–51.

[3] Halperin, D., Hu, W., Sheth, A., and Wetherall, D. Tool release: Gathering 802.11n traces with channel state information. *ACM SIGCOMM CCR 41*, 1 (Jan. 2011), 53.

[4] Holstein, T., Dodig-Crnkovic, G., and Pelliccione, P. Ethical and social aspects of self-driving cars. *arXiv preprint arXiv:1802.04103* (2018).

[5] Howard, D., and Dai, D. Public perceptions of self-driving cars: The case of berkeley, california. In *Transportation research board 93rd annual meeting* (2014), vol. 14-4502, pp. 1–16.

[6] Hwang, T., Yang, C., Wu, G., Li, S., and Ye Li, G. Ofdm and its wireless applications: A survey. *IEEE Transactions on Vehicular Technology 58*, 4 (2009), 1673–1694.

[7] Institute of Electrical and Electronics Engineers. Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless lan medium access control (mac)and physical layer (phy) specifications amendment 5: Enhancements for higher throughput. *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)* (2009), 1–565.

[8] Jeon, W. G., Chang, K. H., and Cho, Y. S. An equalization technique for orthogonal frequency-division multiplexing systems in time-variant multipath channels. *IEEE Transactions on Communications 47*, 1 (1999), 27–32.

[9] Kotaru, M., Joshi, K., Bharadia, D., and Katti, S. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (2015), pp. 269–282.

[10] Robertson, P., and Kaiser, S. The effects of doppler spreads in ofdm(a) mobile radio systems. In *Gateway to 21st Century Communications Village. VTC 1999-Fall. IEEE VTS 50th Vehicular Technology Conference (Cat. No.99CH36324)* (1999), vol. 1, pp. 329–333 vol.1.

[11] Roque, D., and Siclet, C. Performances of weighted cyclic prefix ofdm with low-complexity equalization. *IEEE Communications Letters 17*, 3 (2013), 439–442.

[12] Schmidt, R. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation 34*, 3 (1986), 276–280.

[13] Tan, J. K. *An adaptive orthogonaly division multiplexing baseband modem for wideband wireless communication.* PhD thesis, Massachusetts Institute of Technology, 2006.

[14] Teoh, E. R., and Kidd, D. G. Rage against the machine? google's self-driving cars versus human drivers. *Journal of safety research 63* (2017), 57–60.

[15] Trubuil, J., Le Gall, T., and Chonavel, T. Synchronization, doppler and channel estimation for ofdm underwater acoustic communications. In *OCEANS 2014 - TAIPEI* (2014), pp. 1–6.

[16] Tufvesson, F. *Design of Wireless Communication Systems-Issues on Synchronization, Channel Estimation and Multi-Carrier Systems.*, vol. 19. Signal Processing Group, Department of Applied Electronics, Lund University, 2000.

[17] Van Veen, B., and Buckley, K. Beamforming: a versatile approach to spatial filtering. *IEEE ASSP Magazine 5*, 2 (1988), 4–24.

[18] Xie, Y., Li, Z., and Li, M. Precise power delay profiling with commodity wifi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2015), MobiCom '15, ACM, p. 53–64.

[19] Zhao, Y., Zhao, W., Wang, G., Ai, B., Putra, H. H., and Juliyanto, B. Aoa-based channel estimation for massive mimo ofdm communication systems on high speed rails. *China Communications 17*, 3 (2020), 90–100.

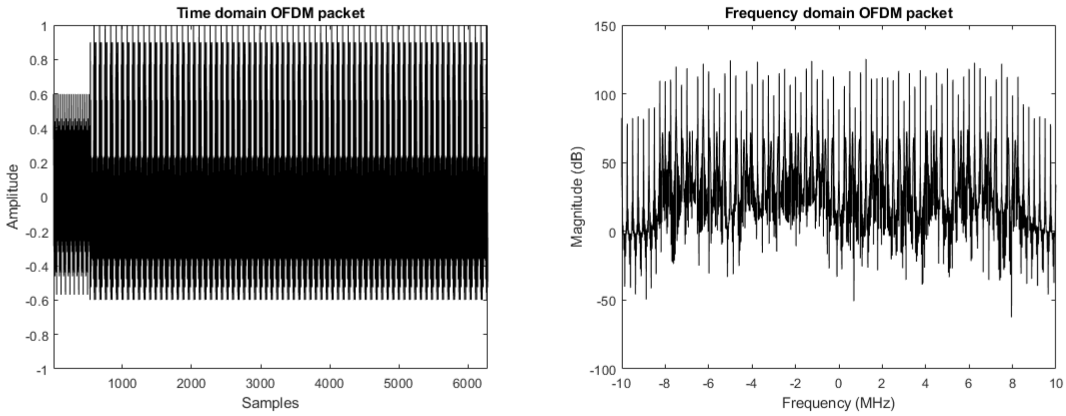


Figure 4: OFDM packet used in our testing setup. There are 8 preamble symbols, each with the bit sequence 01-10010101-11110011-01010000-00110010-10000001-10010100-00. There are 72 data symbols, each with the bit sequence 10001111-11110001-11111100-01111001-10011011-11100000.

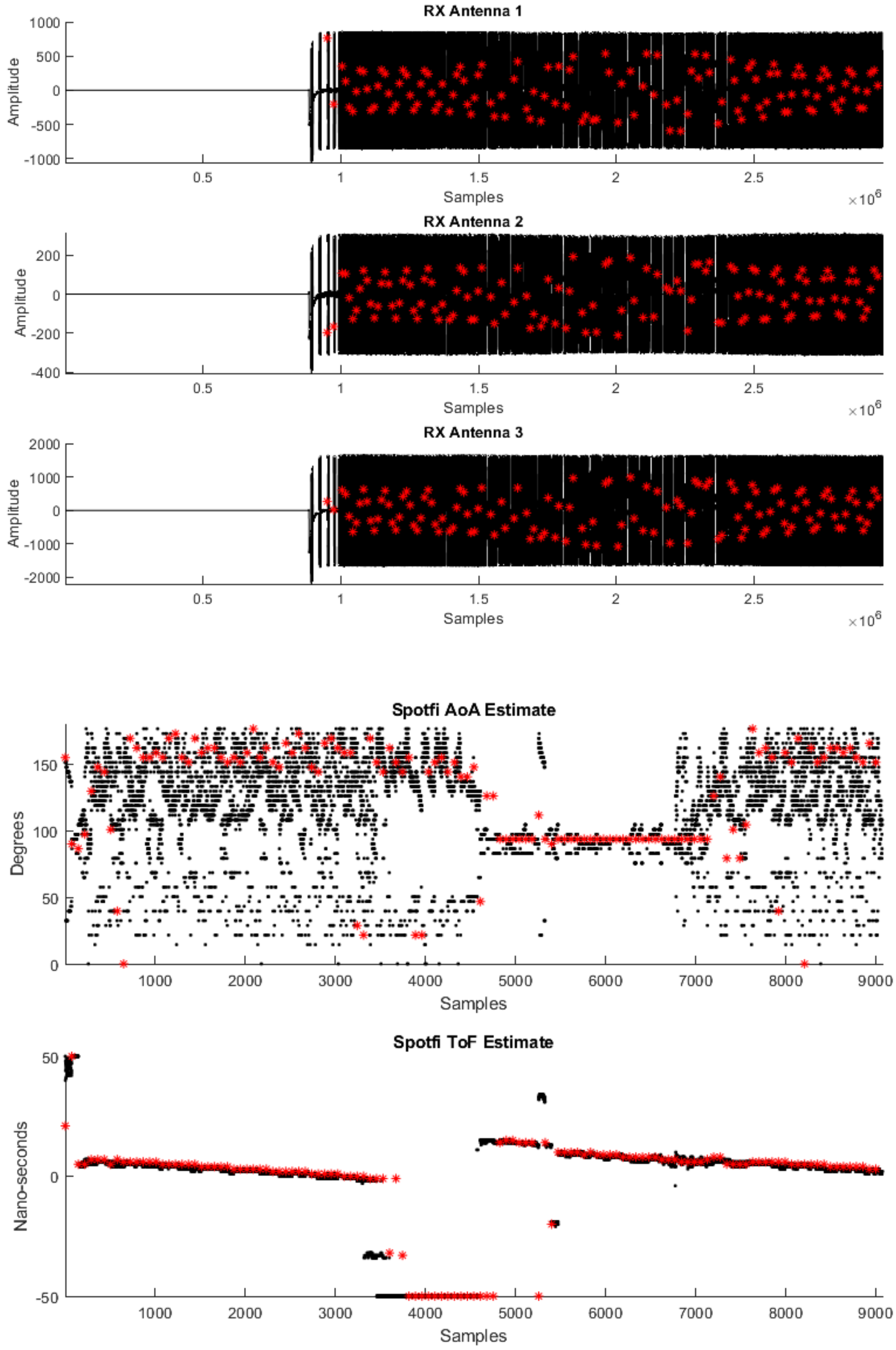


Figure 5: Receiver array is about 86 centimeters from the transmitter. Recording ran for about 20 seconds, but the first 15 seconds (or so) are lost due to USRP quirks. Signal arrives at -85 degrees (negative implies *behind* the array), using 0 degrees as the right-most antenna and 180 degrees as the left-most antenna. The AoA estimate here fluctuates between about 90 degrees and about 160 degrees. For the second plot, red represents the estimate using the channel obtained from the OFDM preamble, and black represents the estimate using the OFDM accumulated channel estimate during data symbol decoding.

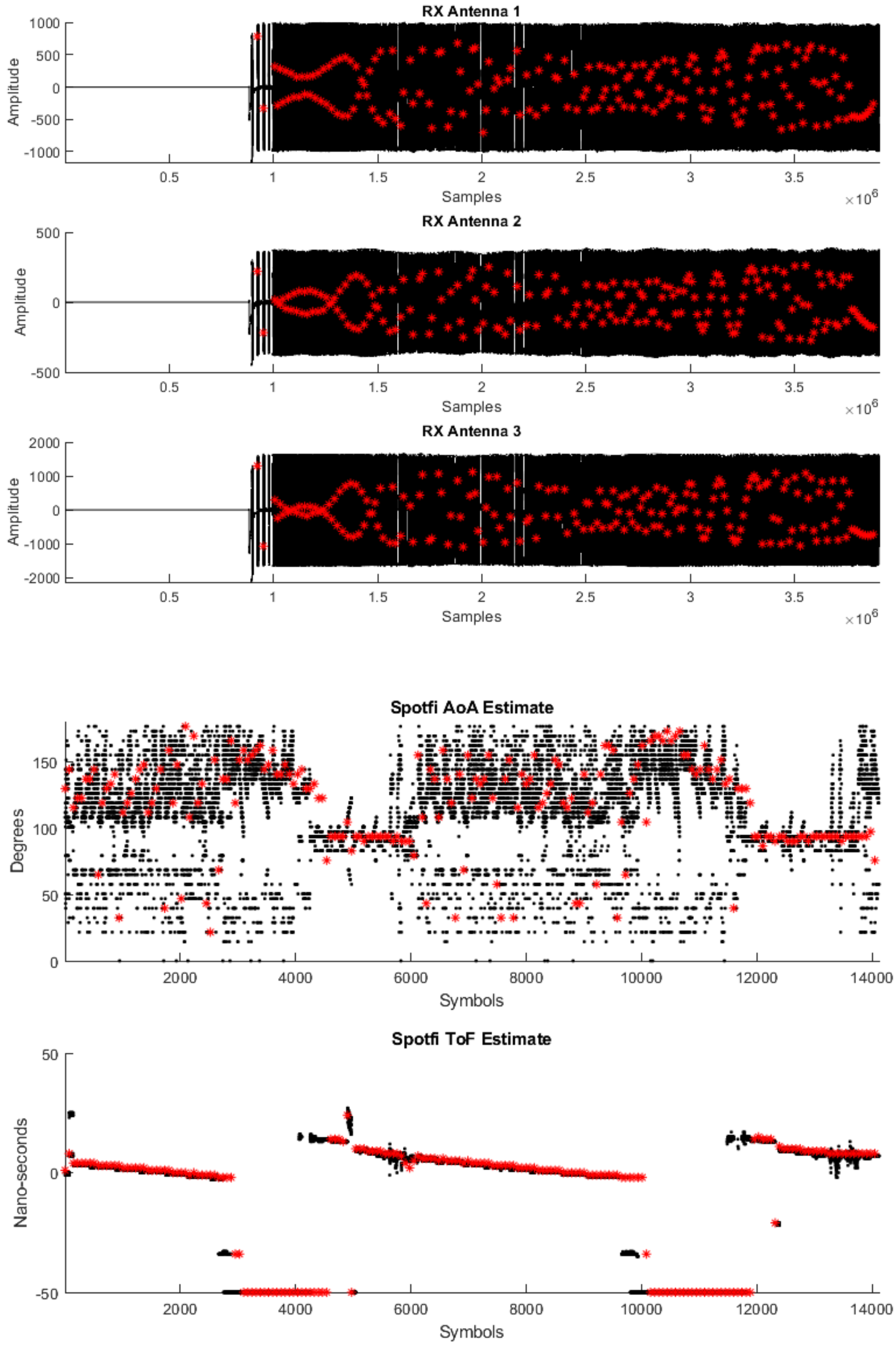


Figure 6: Same setup as before, but signal arrives at 95 degrees. We should expect this result to be similar to the previous. The AoA estimate here fluctuates between about 93 degrees and about 140 degrees.

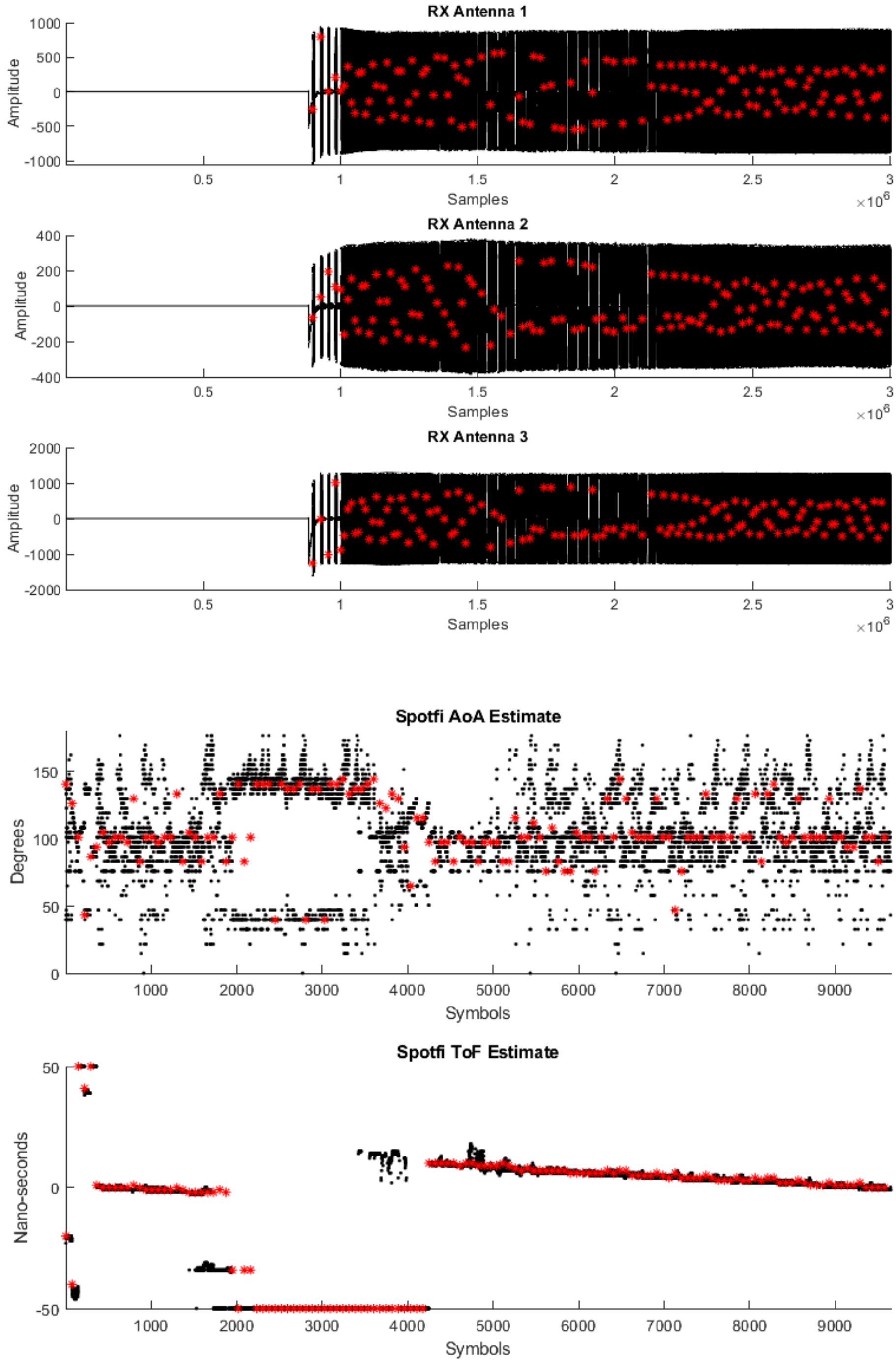


Figure 7: Same setup as before, but signal arrives at 5 degrees. The AoA estimate here fluctuates between about 93 degrees and about 160 degrees The resulting estimation is incorrect.

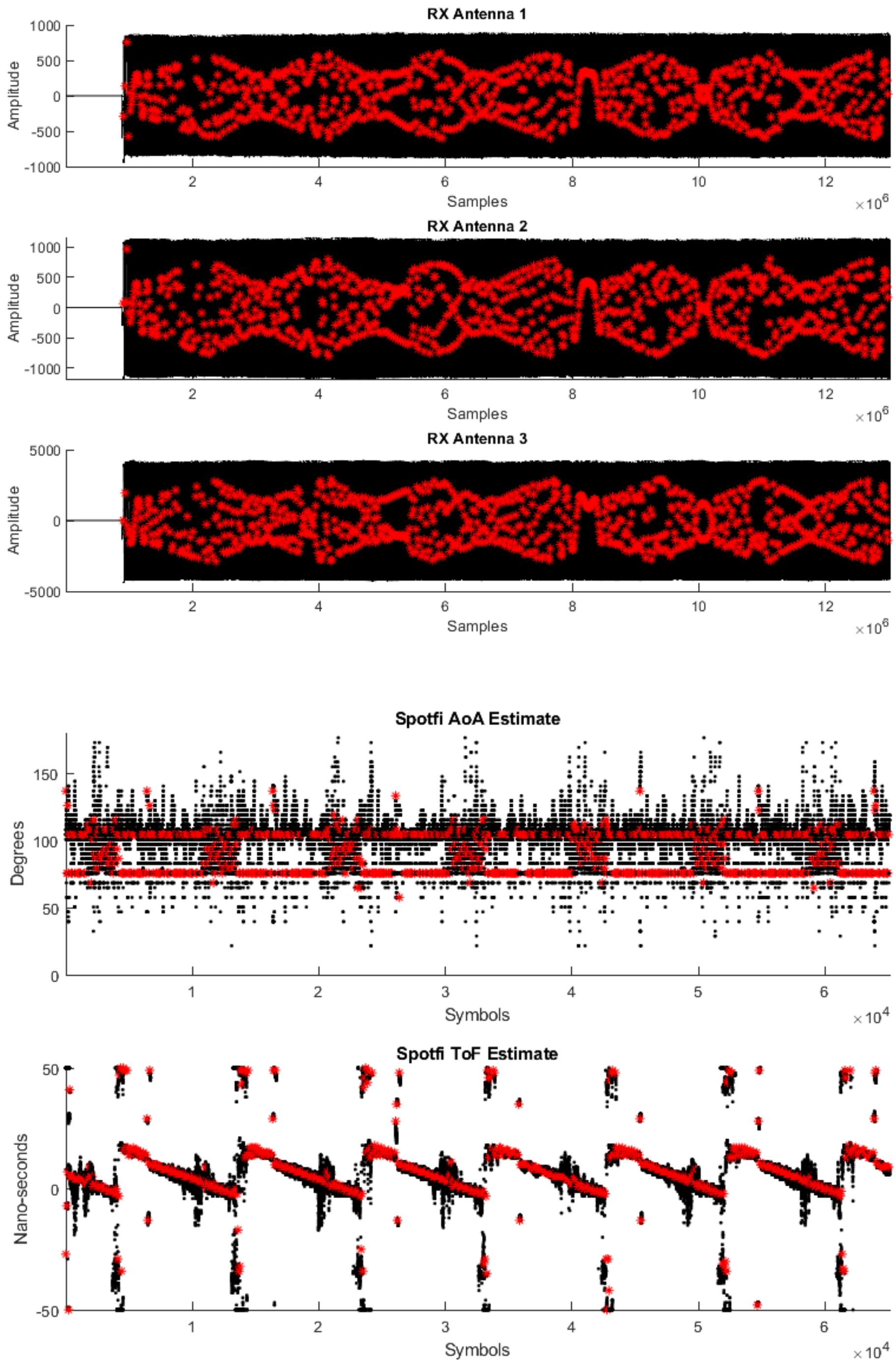


Figure 8: Same setup as before, but signal arrives at 113 degrees at distance about 115 centimeters. This test was ran for 30 seconds (the first 15 being lost), and thus yielded significantly more data. The AoA estimate here fluctuates between about 75 degrees and about 104 degrees. Notice that each preamble-computed channel is roughly the same for each cycle, but the individual data symbol channels are slightly different.

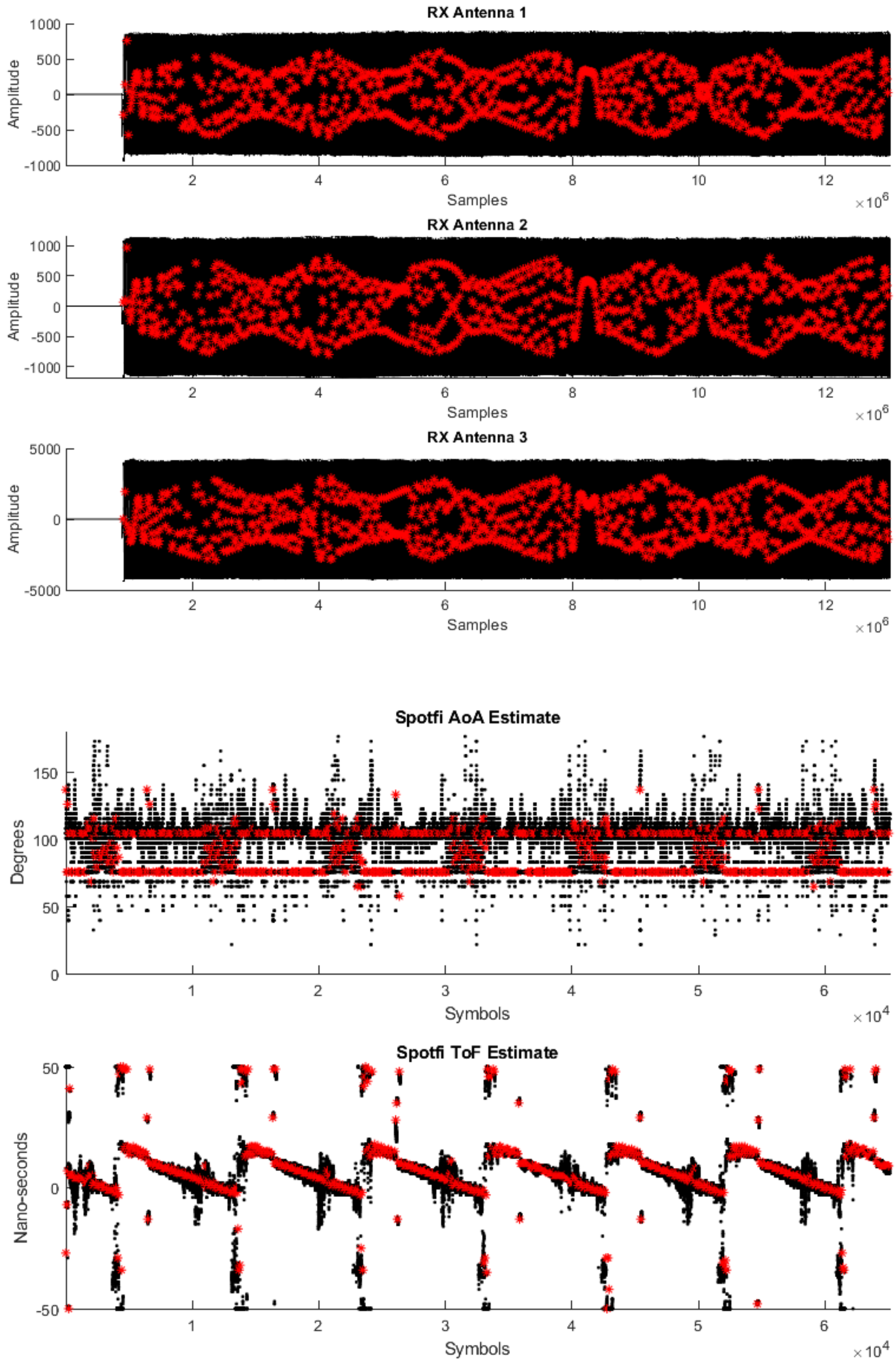


Figure 9: Same setup as before, but we moved the receiver while recording. The start distance is about 115 centimeters, and the end distance is about 50 centimeters. The start angle is about 30 degrees, and the end angle is about 90 degrees. We can see here that our recording is quite inaccurate. The movement only seems to be visible in the first half, where the signal amplitude varies quite a bit. Likely, our testing was inaccurate due to being unsure of the exact timings of when MATLAB and the USRPs transmit data.