

WIRELESS PHASE AND FREQUENCY SYNCHRONIZATION FOR DISTRIBUTED
PHASED ARRAYS

By

Serge R. Mghabghab

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering – Doctor of Philosophy

2022

ABSTRACT

WIRELESS PHASE AND FREQUENCY SYNCHRONIZATION FOR DISTRIBUTED PHASED ARRAYS

By

Serge R. Mghabghab

Distributed microwave wireless systems have the potential to dramatically reshape wireless technologies due to their ability to provide improvements in robustness, transmit power, antenna gain, spatial and temporal resolutions, size, scalability, secrecy, flexibility, and cost compared to single-platform wireless systems. Traditional wireless systems use a platform-centric model, where improving capabilities generally necessitates hardware retrofitting, which in many cases can result in a bulky, expensive, and inefficient system. Nevertheless, distributed microwave wireless systems require precise coordination to enable cooperative operation. The most highly synchronized systems coordinate at the wavelength level, supporting coherent distributed operations like beamforming. The electric states that need to be synchronized in coherent distributed arrays are mainly the phase, frequency, and time; the synchronization can be accomplished using multiple architectures. All coordination architectures can be grouped under two categories: open loop and closed loop. While closed-loop systems use feedback from the destination, open-loop coherent distributed arrays must synchronize their electrical states by only relying on synchronization signals stemming from within the array rather than depending on feedback signals from the target. Although harder to implement, open-loop coherent arrays enable sensing and other delicate communications applications, where feedback from the target is not possible.

In this thesis, I focus on phase alignment and frequency synchronization for open-loop coherent distributed antenna arrays. Once the phase and frequency of all the nodes in the array are synchronized, it is possible to coherently beamform continuous wave signals. When information is modulated on the transmitted continuous waves, time alignment between the nodes is needed. However, time alignment is generally less stringent to implement since its requirements are dependent on the information rate rather than the beamforming frequency, such as for phase and

frequency synchronization. Beamforming at 1.5 GHz is demonstrated in this thesis using a two-node open-loop distributed array. For the presented architecture, the phases of the transmitting nodes are aligned using synchronization signals incoming from within the array, without any feedback from the destination. A centralized phase alignment approach is demonstrated, where the secondary node(s) minimize their relative phase offsets to that of the primary node by locating the primary node and estimating the phase shift imparted by the relative motion of the nodes. A high accuracy two-tone waveform is used to track the primary node using a cooperative approach. This waveform is tested with an adaptive architecture to overcome the performance degradation due to weather conditions and to allow high ranging accuracy with minimal spectral footprint. Wireless frequency synchronization is implemented using a centralized approach that allows phase tracking, such that the frequencies of the secondary nodes are locked to that of the primary node. Once the phase and frequency of all the nodes are synchronized, it is possible to coherently beamform in the far field as long as the synchronization is achieved with the desired accuracy. I evaluate the required localization accuracies and frequency synchronization intervals. More importantly, I demonstrate experimentally the first two-node open-loop distributed beamforming at 1.5 GHz with multiple scenarios where the nodes are in relative motion, showing the ability to coherently beamform in a dynamic array where no feedback from the destination is needed.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor Dr. Jefferey Nanzer for all the support and help that he has given me throughout my Ph.D. Your valuable expertise, encouragement, and guidance helped me overcome all the difficulties and achieve my goals. I am glad that being part of your team made me a part of the Electromagnetics Research Group at Michigan State University as well. I would also like to express my deepest appreciation to my committee which included Dr. Ming Yan, Dr. Prem Chahal, Dr. Shanker Balasubramaniam. You always had a positive attitude, helped me with my questions and concerns, and made my journey easy. In addition to committee, I would like to thank Dr. Edward Rothwell for the helpful advice and mentoring.

Thank you to Defense Advanced Research Projects Agency and Office of Naval Research for funding my education and research, and for providing me with the tools that I needed to make a difference. In addition, I am deeply grateful to all the friends and colleagues that I have met at MSU and in both the Electromagnetics Research Group and the DELTA group. Namely, I would like to thank Stavros Vakalis, Xenofon Konstantinou, Anton Schlegel, Daniel Chen, Jason Merlo, Abdel Alsnayyan, Omkar Ramachandran, Sean Ellison, Neda Nourshamsi, Ahona Bhattacharyya, Amer Abu Arisheh, Will Torres, Jorge Colon-Berrios, Cory Hilton, Michael Craton, Vinny Gjokaj. Without you, my graduate life would have been lonelier and harder. Finally I would like to thank my family, friends, and wife Sarah for believing in me and supporting me.

TABLE OF CONTENTS

| | |
|---|-----------|
| LIST OF TABLES | viii |
| LIST OF FIGURES | ix |
| LIST OF ALGORITHMS | xxi |
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 Coordination of Principal Parameters | 2 |
| 1.1.1 Time Alignment | 3 |
| 1.1.2 Phase Alignment | 5 |
| 1.1.3 Frequency locking | 6 |
| 1.2 Architectures of Coherent Distributed Arrays | 8 |
| 1.3 Main Contribution of this Dissertation | 12 |
| CHAPTER 2 LOCALIZATION REQUIREMENTS FOR OPEN-LOOP COHERENT DISTRIBUTED ARRAYS | 15 |
| 2.1 Coherent Gain | 15 |
| 2.2 Ranging Requirements for Centralized Open-Loop Coherent Distributed Arrays . . | 16 |
| 2.2.1 Ranging Requirements with Wired Frequency Synchronization | 17 |
| 2.2.2 Ranging Requirements with Wireless Frequency Synchronization | 20 |
| 2.3 Localization Requirements for Centralized Open-Loop Coherent Distributed Arrays | 25 |
| 2.3.1 Localization using Range and Angle of Arrival | 25 |
| 2.3.2 Localization using Multilateration | 31 |
| CHAPTER 3 FREQUENCY SYNCHRONIZATION REQUIREMENTS FOR OSCILLATORS AND PHASE-LOCKED LOOPS | 35 |
| 3.1 Phase Noise of and Operation of Oscillators and Phase-Locked Loops | 36 |
| 3.1.1 Oscillator Phase Noise | 36 |
| 3.1.2 General Architecture of PLLs | 38 |
| 3.1.3 PLL Phase Noise | 42 |
| 3.2 Phase Noise Interpretation of Locked and Unlocked Oscillators in Frequency and Time | 45 |
| 3.2.1 Time Jitter | 45 |
| 3.2.2 Allan Deviation and Frequency Drift | 47 |
| 3.2.3 Time Drift | 49 |
| 3.3 Distributed Beamforming Analysis | 50 |
| 3.4 Practical Examples | 52 |
| 3.4.1 Beamforming Performance with Continuous Synchronization | 56 |
| 3.4.2 Beamforming Performance with Periodic Synchronization with Phase and Frequency Errors | 58 |
| 3.4.3 Beamforming Performance with Periodic Synchronization with Phase, Frequency, and Timing Errors | 60 |

| | | |
|-----------|--|-----|
| CHAPTER 4 | WIRELESS FREQUENCY SYNCHRONIZATION FOR CENTRALIZED SYSTEMS | 62 |
| 4.1 | Frequency Synchronization Circuit | 62 |
| 4.2 | Output of the Frequency Synchronization Circuit | 65 |
| 4.3 | Impact of Doppler Frequency | 68 |
| 4.4 | Phase Change Due To Frequency Synchronization Signal Propagation | 70 |
| CHAPTER 5 | ADAPTIVE RANGING THAT SUPPORTS COHERENT BEAMFORMING OPERATIONS | 73 |
| 5.1 | CRLB of Range Estimates | 73 |
| 5.1.1 | CRLB of Typical Two-Tone Waveforms | 76 |
| 5.1.2 | CRLB of Multi-Tone Waveforms | 79 |
| 5.2 | High Accuracy Ranging | 83 |
| 5.2.1 | Minimization of Ranging Biases | 86 |
| 5.2.1.1 | Peak Estimators | 88 |
| 5.2.1.2 | Comparison Between the Peak Estimators | 91 |
| 5.2.2 | Scalability of Two-Tone Waveforms | 94 |
| 5.3 | Adaptive Ranging | 98 |
| 5.3.1 | Using the Standard Deviation of Multiple Range Estimates as Feedback | 100 |
| 5.3.1.1 | Adaptive Ranging Framework | 100 |
| 5.3.1.2 | Adaptive Indoor Ranging Experiment | 107 |
| 5.3.1.3 | Adaptive Outdoor Ranging Experiment | 111 |
| 5.3.2 | Using Only One Ranging Pulse as Feedback | 123 |
| 5.3.2.1 | SNR estimation from One Pulse | 125 |
| 5.3.2.2 | Ranging Standard Deviation Estimation from SNR | 127 |
| 5.3.2.3 | SNR-Based Perception Design | 134 |
| 5.3.2.4 | Adaptive Ranging Results | 137 |
| CHAPTER 6 | BEAMFORMING EXPERIMENTS | 140 |
| 6.1 | Effects of the Displacement of Nodes on the Beamforming Performance | 140 |
| 6.1.1 | Frequency Locking Circuit Receiver Displacement | 144 |
| 6.1.2 | Primary Transmitter Displacement | 145 |
| 6.1.3 | Full Node Displacement | 146 |
| 6.2 | Coherent Beamforming using Frequency Locking Circuit and Ranging | 147 |
| 6.2.1 | Range Estimation | 147 |
| 6.2.2 | Distributed Beamforming Experiment | 151 |
| CHAPTER 7 | TWO-DIMENSIONAL LOCALIZATION TECHNIQUE THAT ENABLES BEAMFORMING | 154 |
| 7.1 | TOA Using Only One Secondary Node | 154 |
| 7.2 | Experimental Results | 156 |
| 7.3 | Discussion | 159 |
| CHAPTER 8 | CONCLUSION | 161 |

| | |
|--------------|---|
| APPENDICES | 163 |
| APPENDIX A | MATLAB RANGING REQUIREMENTS FOR COHERENT DISTRIBUTED ARRAYS 164 |
| APPENDIX B | MATLAB FREQUENCY SYNCHRONIZATION REQUIREMENTS FOR NODES WITH PLL 174 |
| APPENDIX C | MATLAB PEAK ESTIMATORS FOR RANGING 181 |
| APPENDIX D | MATLAB SNR ESTIMATION USING EITHER EIGENVALUE DECOMPOSITION OR MATCHED FILTERING 190 |
| APPENDIX E | MATLAB TIME OF ARRIVAL (TOA) ALGORITHM 194 |
| APPENDIX F | MATLAB CRAMER-RAO LOWER BOUND FOR TIME OF ARRIVAL (TOA) 198 |
| BIBLIOGRAPHY | 204 |

LIST OF TABLES

| | |
|--|-----|
| Table 3.1: SSB phase noise of the selected 100 MHz oscillators. | 53 |
| Table 5.1: Ranging RMSE for the three ranging algorithms. | 95 |
| Table 7.1: Standard deviation (STD) of the estimated locations from experimental measurements versus CRLB in meters. From [1] © 2021 IEEE. | 157 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1.1: General architecture of coherent distributed arrays where the nodes achieve synchronization using inter-node signals and external signals from the targeted location. Once all the nodes are synchronized, the transmitted signals add coherently at the target and provide an increased signal power. | 2 |
| Figure 1.2: Two ideal received signals along with their summation from two perfectly synchronized nodes. | 2 |
| Figure 1.3: Two ideal received signals along with their summation from two nodes performing phase and frequency synchronization without time alignment. | 3 |
| Figure 1.4: Two ideal received signals along with their summation from two nodes performing time and frequency synchronization without phase alignment. | 5 |
| Figure 1.5: Two ideal received signals along with their summation from two nodes performing time and phase synchronization without frequency synchronization. | 6 |
| Figure 1.6: Receiver-coordinated beamforming using explicit-feedback (adapted from [2]). The receiver allocates the channel sounding period time slots along with the beamforming time slot. The nodes transmit an uncompensated carrier in their allocated sounding slot and the target transmits back the feedback signals afterwards to synchronize their phase and frequency. Once synchronized the nodes beamform at the dedicated beamforming slot. From [3] © 2021 IEEE. | 8 |
| Figure 1.7: Retrodirective beamforming where time slots are assigned so that the nodes can synchronize their frequencies to that of the target. The channels are estimated between the nodes and the target, a phase offset is applied so that the secondary nodes are phase aligned to the phase of the primary node. From [3] © 2021 IEEE. | 9 |
| Figure 1.8: Open-loop distributed phased arrays beamforming at an angle θ_0 in the far field. Time, phase, and frequency are synchronized using internode signaling in a centralized architecture. Localizing the primary and/or the secondary nodes is important to align the phases. | 10 |
| Figure 2.1: Two nodes with a hierarchical architecture beamforming at a target in the far field. Frequency synchronization is achieved using a cabled connection, where the primary node transmits a reference frequency signal to lock the PLL of the secondary node. Beamforming is done by accounting for the phase shift produced by the relative motion between the two transmitters. | 18 |

| | |
|--|----|
| Figure 2.2: Probability of the coherent gain exceeding 0.9 versus ranging RMSE σ_d in terms of the signal wavelength λ considering the effects of phase changes where no wireless frequency synchronization is considered. | 20 |
| Figure 2.3: Probability of the coherent gain exceeding 0.9 versus the transmission orientation in the case where frequency synchronization does not contribute to the phase errors. From [4] © 2020 IEEE. | 21 |
| Figure 2.4: Two nodes with a hierarchical architecture beamforming at a target in the far field. Frequency synchronization is achieved wirelessly, where the secondary node locks its oscillator phase to that of the incoming synchronization signals from the primary node. Beamforming is done by accounting for the phase shift produced by the relative motion between the two transmitters and between the synchronization antennas. From [4] © 2020 IEEE. | 22 |
| Figure 2.5: Probability of the coherent gain exceeding 0.9 versus ranging RMSE σ_d in terms of the signal wavelength λ considering the effects of phase changes where wireless frequency synchronization is present (wireless frequency synchronization using phase locking approaches). From [4] © 2020 IEEE. | 23 |
| Figure 2.6: Probability of the coherent gain exceeding 0.9 versus the transmission orientation in the case where wireless frequency synchronization contributes to the phase errors. From [4] © 2020 IEEE. | 24 |
| Figure 2.7: Fifty thousand Monte Carlo simulation showing various threshold values for achieving coherent gain. The probabilities were estimated as a function of angle and localization standard deviation. The strictest requirements for localization uncertainty appear around 90° | 25 |
| Figure 2.8: Probability of the coherent gain exceeding 0.9 versus the transmission orientation in the case where wireless frequency synchronization contributes to the phase errors. These results consider the case where the primary node locks its frequency to that of the secondary node in a two-node array. From [4] © 2020 IEEE. | 26 |
| Figure 2.9: Hierarchical open-loop coherent distributed array beamforming at an angle θ_0 . The angles θ_1 and θ_2 represent the angles of the primary node relative to the planes of the secondary nodes. From [5] © 2021 IEEE. | 27 |
| Figure 2.10: Radial error $\sqrt{\sigma_{x_n}^2 + \sigma_{y_n}^2}$ in terms of λ for maximum and minimum coordinates of ± 10 m for x and y in the case of $\sigma_{d_n} = 0.01\lambda$ and $\sigma_{\theta_n} = 1^\circ/\text{m}$. From [5] © 2021 IEEE. | 28 |

| | |
|--|----|
| Figure 2.11: Probability of the coherent gain to exceed 0.9 versus the maximum allowed x and y coordinates, where $\sigma_{d_n} = 0.01\lambda$ and $\sigma_{\theta_n} = 0.1^\circ/\text{m}$. From [5] © 2021 IEEE. | 29 |
| Figure 2.12: The required σ_θ to maintain a 0.9 coherent gain with 90% probability for 10 nodes with multiple values of σ_d versus the maximum allowed coordinates for x and y . From [5] © 2021 IEEE. | 30 |
| Figure 2.13: Hierarchical open-loop coherent distributed arrays that rely on multilateration using (a) TOA and (b) TDOA measurements with circles (in (a)) and hyperbolas (representing the time difference of arrival between every pair of nodes in (b)) respectively to estimate the location of the primary node and synchronize the phase of the secondary nodes. | 32 |
| Figure 2.14: CRLB of the Transmitter localization in terms of Radial error $\sqrt{\sigma_{x_n}^2 + \sigma_{y_n}^2}$ as a function of λ for TOA systems with (a) 3 nodes and $\sigma_d = 0.01\lambda$, (b) 3 nodes and $\sigma_d = 0.1\lambda$, (c) 3 nodes and $\sigma_d = 0.1\lambda$, (d) 4 nodes and $\sigma_d = 0.1\lambda$, (e) 4 nodes and $\sigma_d = 0.05\lambda$, (f) 5 nodes and $\sigma_d = 0.1\lambda$. The secondary nodes (anchors) locations are marked with red. The RMSE of range (time delay) estimates is considered equal for all the anchors. | 34 |
| Figure 3.1: Phase noise profile of the wirelessly synchronized secondary nodes where the reference oscillator has lower phase noise in comparison to the phase noise on the locked oscillator. From [6] © 2021 IEEE. | 35 |
| Figure 3.2: General shape of the phase noise of an oscillator. The noise power associated with each phase noise type is represented by b_i . From [6] © 2021 IEEE. | 37 |
| Figure 3.3: Typical PLL architecture with frequency synthesizer. F_{ref} represents the input frequency to the PLL, this frequency can be lower version than the input reference frequency, which can be generated using frequency dividers. F_{out} is the output frequency of the VCO, while the $F_{feedback}$ is a divided version of the F_{out} . From [6] © 2021 IEEE. | 39 |
| Figure 3.4: PLL represented using a negative feedback model. | 41 |
| Figure 3.5: A three element LPF with resistive and capacitive components. | 42 |
| Figure 3.6: PLL noise model. | 43 |
| Figure 3.7: PLL phase noise shape base on the selection of the loop bandwidth, where f_{LF} represents the frequency of the loop filter of the PLL. The bandwidth can be: (a) optimum, (b) large, (c) narrow. | 44 |

Figure 3.8: Typical PLL phase noise profile. In the first region, the phase noise of the reference signal dominates the PLL phase noise; on the other hand, the close-in noise dominates region 2 leading to an increase in the PLL noise floor; finally, region 3 is dominated mainly by the phase noise of the locked VCO. f_{LF} , which represents the frequency of the loop filter of the PLL, needs to be selected appropriately as shown in Fig. 3.7 to minimize the phase noise of the PLL. From [6] © 2021 IEEE. 45

Figure 3.9: Example of a clock jitter for the real case compared to the ideal time domain performance of a clock where no jitter is observed. 45

Figure 3.10: AVAR profile for an oscillator. $\tau_{f,min}$ represents the instance where the AVAR is mainly dominated by the random walk FM noise. Later on, when analyzing the RMS frequency drift, it is important to select a value $\tau_f \geq \tau_{f,min}$. From [6] © 2021 IEEE. 48

Figure 3.11: Phase noise of the locked PLLs for architectures: (a) 1, (b) 2, and (c) 3. In architecture 1, VCO A was being locked by the PLLs of the secondary nodes and it was used to generate the reference signals. In architecture 2, VCO A was replaced by VCO B. In architecture 3, VCO B was used to generate the reference signals, while VCO A was being locked by the PLLs of the secondary nodes. In all the architectures, the loop filter was assumed to be designed appropriately to minimize the phase noise of the PLLs. 54

Figure 3.12: Coherent gain for various carrier frequencies for arrays of sizes $N = 2, 10, 20,$ and 100 with continuous locking for architecture 1 (using the high phase noise oscillator as the reference and PLL VCO) with the phase noise profile from Fig. 3.11a. 55

Figure 3.13: Coherent gain for various carrier frequencies for arrays of sizes $N = 2, 10, 20,$ and 100 with continuous locking for architecture 2 (using the low phase noise oscillator as the reference and PLL VCO) with the phase noise profile from Fig. 3.11b. 56

Figure 3.14: Coherent gain for various carrier frequencies for arrays of sizes $N = 2, 10, 20,$ and 100 with continuous locking for architecture 3 (using the low phase noise oscillator as the reference and the high phase noise oscillator as the PLL VCO) with the phase noise profile from Fig. 3.11c. 57

Figure 3.15: Coherent gain for CW and BPSK signals with periodic synchronization, where the jitter and frequency drifts (CW and BPSK), and timing errors (BPSK) were considered for nodes equipped with VCO A. Array size $N = 100$ was selected. 58

| | |
|---|----|
| Figure 3.16: Coherent gain for CW and BPSK signals with periodic synchronization, where the jitter and frequency drifts (CW and BPSK), and timing errors (BPSK) were considered for nodes equipped with VCO B. Array size $N = 100$ was selected. | 59 |
| Figure 3.17: Carrier frequencies supporting a coherent gain of 0.9 for periodic synchronization with array sizes $N = 2$ and 100, where the jitter, frequency drifts and timing errors were considered for nodes equipped with either VCO A or B. | 60 |
| Figure 4.1: The adjunct self-mixing circuit that was used by the secondary nodes to lock their frequencies to that of the primary node. This circuit receives two-tone signals with 10 MHz tone separation and outputs a 10 MHz signal that can be fed to the input of the PLL (REF IN). A splitter was used at the input to split the received signals into ranging and frequency synchronization signals. From [7] © 2021 IEEE. | 64 |
| Figure 4.2: Block diagram of the adjunct self-mixing circuit for wireless frequency synchronization (adapted from [8]). | 64 |
| Figure 4.3: Block diagram of the experimental setup used for testing the spectrum at the output of the frequency locking circuit. | 65 |
| Figure 4.4: Power spectral density of both the reference clock, and the output of the proposed frequency locking circuit. From [9] © 2021 IEEE. | 65 |
| Figure 4.5: Block diagram of the experimental setup used for testing the CW 10 MHz output signals from both frequency locked and frequency unlocked SDRs. | 66 |
| Figure 4.6: Power spectral density of the reference clock, the locked clock, and the unlocked one. From [9] © 2021 IEEE. | 66 |
| Figure 4.7: The effect of mismatch between d_{LO} and d_{RF} on the 10 MHz output from the mixer. | 68 |
| Figure 4.8: The observed Doppler shift for 10 MHz, 1 GHz, and 10 GHz signals in the case where the relative velocity between a primary and a secondary node is varied from 0 to 10 m/s. | 70 |
| Figure 4.9: Block diagram of the setup used to measure the phase shift produced by the frequency synchronization circuit when the antenna receiving the two-tone signal is displaced. From [8] © 2021 IEEE. | 72 |
| Figure 4.10: Experimental results showing the phase shifts between the primary and secondary nodes signals for multiple displacement distances. From [8] © 2021 IEEE. | 72 |

| | |
|--|----|
| Figure 5.1: The ideal spectrum of the two-tone waveforms expressed in (5.10). | 75 |
| Figure 5.2: The minimum achievable standard deviation/RMSE for two-tone CW ranging signals for various values of E/N_0 . From [10] © 2021 IEEE. | 77 |
| Figure 5.3: The ideal spectrum of the two-tone waveforms expressed in (5.17). | 78 |
| Figure 5.4: The ideal spectrum of the four-tone waveforms expressed in (5.18). | 78 |
| Figure 5.5: The ideal spectrum of the three-tone waveforms expressed in (5.18). | 78 |
| Figure 5.6: (a) The in-phase part of a two-tone signal with $f_1 = 53$ kHz, $f_2 = 1.03$ MHz, and a sampling frequency of $f_s = 2.5$ MHz showing discretizations equivalent to two different delays. (b) Output of the matched filter for both scenarios, showing the error due to discretization. | 85 |
| Figure 5.7: (a) Two-tone ranging pulse with $f_1 = 20$ kHz and $f_2 = 7.52$ MHz, along with a disambiguation pulse with $f_d = 1.875$ MHz. These signals were sampled at 25 MSps. (b) Output of the matched filter for both the ranging pulse and the disambiguation pulse. From [10]. | 87 |
| Figure 5.8: RSS of the normalized matched filter outputs for three time delay cases for a ranging waveform with $f_1 = 20$ kHz, $f_2 = 4.5$ MHz, $f_s = 10$ MHz, and $T = 99.8$ μ s. The three selected time delay cases reside within the same two discretization points. Every expected matched filter output h corresponds to a predicted time shift dt_h | 92 |
| Figure 5.9: Experimental SDR setup. Extra cables were added to modify the received signal time delay. | 93 |
| Figure 5.10: Simulated and experimental results showing a comparison between the actual distances traveled by the signals and their estimates. The surface shows the simulated results, and the red dots show the experimental results of interpolation. From [11] © 2021 IEEE. | 94 |
| Figure 5.11: Simulated and experimental results showing a comparison between the actual distances traveled by the signals and their estimates. The surface shows the simulated results, and the red dots show the experimental results of NL-LS. From [11] © 2021 IEEE. | 94 |
| Figure 5.12: Simulated and experimental results showing a comparison between the actual distances traveled by the signals and their estimates. The surface shows the simulated results, and the red dots show the experimental results of MF-LS. From [11] © 2021 IEEE. | 95 |

| | |
|--|-----|
| Figure 5.13: (a) The errors from Figs. 5.10, 5.11, and 5.12 for the various peak estimation algorithms for different cable lengths and bandwidths (average of 100 estimates). (b) Time delay based on the cable length (dotted orange line) and signal bandwidth (solid blue line) per sample. | 96 |
| Figure 5.14: The standard deviation for each ranging method along with the CRLB. The SNR for the first 100 samples was 32 dB and then it decreased by around 2.9 dB for every added cable (every 100 samples), reaching an SNR of 14.6 dB. For some samples, the standard deviation of both the interpolation and NL-LS is smaller than the CRLB, this is due to the bias in the estimates. From [11] © 2021 IEEE. | 97 |
| Figure 5.15: Time domain of a TTFSW with $N_p = 5$ and 50% duty cycle. From [7] © 2021 IEEE. | 98 |
| Figure 5.16: Time-frequency spectrum of a TTFSW with $N_p = 5$ and 50% duty cycle. From [7] © 2021 IEEE. | 98 |
| Figure 5.17: Matched filter output for TTFSW with $N_p = 4$ compared to the matched filter output of pulsed two-tone waveforms (PTTW). From [7] © 2021 IEEE. | 99 |
| Figure 5.18: Adaptive inter-node ranging architecture that relies on the standard deviation of the range estimates. From [10] © 2021 IEEE. | 101 |
| Figure 5.19: A simplified representation of the used PID controller. In this figure, $y[n]$ represents the system output, which is the standard deviation of the time-delay estimates of multiple consecutive ranging pulses. | 104 |
| Figure 5.20: Block diagram representing the hardware used for the indoor adaptive ranging experiment. From [12] © 2021 IEEE. | 106 |
| Figure 5.21: Experimental setup for adaptive ranging system measurements. | 107 |
| Figure 5.22: Ranging system behavior to the proportional gain K_u | 108 |
| Figure 5.23: The behavior of the adaptive ranging system for a varying SNR. | 110 |
| Figure 5.24: The behavior of the adaptive ranging system for a varying reference point. | 111 |

Figure 5.25: (a) Without phase and frequency synchronization, the nodes in the array cannot align their phases, as it is impossible to prevent the frequencies on each node to relatively drift, and even when the frequency seems to be stable, it is not possible to transmit coherent phases. (b) Once both phase and frequency are synchronized, it is possible to transmit a coherent frequency to the target, and ranging is part of localization that enables phase alignment. From [10]. 112

Figure 5.26: Block diagram for the outdoor adaptive ranging experiment, where wireless frequency synchronization was implemented. Node A is the primary node, while node B represents the secondary node. Wireless frequency synchronization two-tone CW signals was transmitted at 910 MHz and 920 MHz, and the ranging two-tone adaptive waveform was transmitted initially at 2.45 GHz carrier and it was received at 5.8 GHz carrier. From [13] © 2021 IEEE. 115

Figure 5.27: The outdoor adaptive inter-node cooperative ranging architecture. The same approach as in Fig. 5.18 was used, with the addition of a wireless frequency locked repeater. From [10]. 116

Figure 5.28: Each nodes was equipped with two antennas, one for transmit (TX) and one for receive (RX). The antennas on every node were separated by around 2 m, and the separation between the antennas of each node was 90 m. From [10]. 116

Figure 5.29: The hardware that was used to operate node A, including two SRDs. 117

Figure 5.30: The hardware that was used to operate node B, including one SDR and the adjunct self-mixing circuit. 117

Figure 5.31: an image showing the weather sensor Davis 6250 Vantage Vue that was attached to the receive antenna of node A. 119

Figure 5.32: (a) Ranging standard deviation for a 24 hours interval, where $f_1 = 20$ kHz and $f_2 = 3.5$ MHz. The effects of SNR changes and wind speed were shown in (b). Also the contribution of humidity and rain rate was presented in (c). 120

Figure 5.33: Achievable coherent frequency using the obtained ranging standard deviation. Three probabilities for achieving coherent gain of at least 0.9 were investigated. The lines with solid colors represent the moving average with a window of 10, while the faded lines represent the individual estimates. 121

Figure 5.34: (a) The adaptively modified second tone f_2 of the transmitted ranging pulses. (b) Ranging standard deviation for a 24 hours interval, where $f_1 = 20$ kHz and f_2 was adaptively modified. The SNR changes and wind speed are shown in (c), and the humidity and rain rate are presented in (d). 122

| | |
|---|-----|
| Figure 5.35: Achievable coherent frequency using the obtained ranging standard deviation in the adaptive framework. Three probabilities for achieving coherent gain of at least 0.9 were investigated. The lines with solid colors represent the moving average with a window of 10, while the faded lines represent the individual estimates. | 123 |
| Figure 5.36: (a) The adaptively modified second tone f_2 of the transmitted ranging pulses. (b) Ranging standard deviation for 1 week interval, where $f_1 = 20$ kHz and f_2 was adaptively modified. The SNR changes and wind speed are shown in (c), and the humidity and rain rate are presented in (d). | 124 |
| Figure 5.37: Achievable coherent frequency using the obtained ranging standard deviation in the adaptive framework. Three probabilities for achieving coherent gain of at least 0.9 were investigated. The lines with solid colors represent the moving average with a window of 10, while the faded lines represent the individual estimates. | 125 |
| Figure 5.38: An example of transmitted (orange) and received (blue) two-tone signal with 11 dB SNR. From [12] © 2021 IEEE. | 128 |
| Figure 5.39: A simulated example of the autocorrelation and cross-correlation of the pulsed two-tone signals that are shown in Fig. 5.38, where the ratio of amplitudes of the received to transmitted pulses is $K = 0.25$. From [12] © 2021 IEEE. | 129 |
| Figure 5.40: Estimation accuracy of the eigenvalue decomposition-based SNR estimator. Three sampling rates were selected to evaluate the change in accuracy. From [12] © 2021 IEEE. | 130 |
| Figure 5.41: Estimation accuracy of the matched filter-based SNR estimator. One of the two tones had a $\pm 90^\circ$ (same results for both cases) phase shift in comparison to the filter; which leads to the lowest accuracy since only the in-phase signal is being considered. Three sampling rates were selected to evaluate the change in accuracy. From [12] © 2021 IEEE. | 131 |
| Figure 5.42: Estimation accuracy of the matched filter-based SNR estimator. The initial phases of both the signal and filter were equal, leading to the highest accuracy. Three sampling rates were selected to evaluate the change in accuracy. From [12] © 2021 IEEE. | 132 |
| Figure 5.43: Experimental setup for testing the proposed SNR estimation method that relies on the matched filter output of the received pulses. The transmitter was connected to the receiver via 2 m coaxial cable with an additional 30 dB attenuator to ensure that the received power is within the acceptable range. | 133 |

Figure 5.44: Measured SNR estimates using the matched filtering approach for a single pulse and 20 sequential pulses in a SDR-based system. The SNR was controlled by modifying the transmitted power. From [12] © 2021 IEEE. 133

Figure 5.45: Adaptive ranging is necessary when high ranging accuracy and low spectral footprint are needed. This can be done by implementing three processes that ensures that only the desired bandwidth is used. A perception process is used to estimate the SNR of the received ranging pulses and convert this SNR to an estimated standard deviation. The controller compares the estimated standard deviation to a desired value and sends the necessary feedback to the action block that generates the desired waveform. This generated waveform is used for ranging, until a new waveform is generated for the next loop. From [12] © 2021 IEEE. 135

Figure 5.46: Measured standard deviation of the time delay estimator versus the scaled CRLB curve, demonstrating that the CRB from (5.16) can be scaled using a constant parameter to match the observed standard deviation. This experiment was done where no vibration or interference were present. From [12] © 2021 IEEE. 136

Figure 5.47: Measured RMSE of ranging accuracy versus SNR (averaged over 100 measurements) showing only a 3.7 ps improvement when more than one pulse is used. From [12] © 2021 IEEE. 136

Figure 5.48: Experimental setup for the adaptive ranging system with the perception process that relies only on the estimated SNR. The two horn antennas were used for transmission and reception, and the corner reflector represented the target. From [12] © 2021 IEEE. 137

Figure 5.49: Performance of the adaptive ranging approach with modified perception: (a) ranging performance; (b) tone separation; (c) SNR. The SNR was modified by changing the transmitted power. The adaptive system was able to keep up with the change in SNR until a higher bandwidth than what is available was needed. From [12] © 2021 IEEE. 138

Figure 6.1: Block diagram of the setup used to measure the performance of the distributed two-node beamforming experiment. From [14] © 2021 IEEE. 141

Figure 6.2: Image of the experimental setup in the semi-enclosed arch range showing the distributed two-node beamforming experiment. From [14] © 2021 IEEE. 142

| | |
|--|-----|
| Figure 6.3: Beamformed signals from two transmitting nodes at a 1.5 GHz carrier with a 100 kHz CW baseband signal. The secondary node was wirelessly frequency synchronized to the primary node using the adjunct self-mixing circuit, and the received phases were manually adjusted to achieve the highest possible coherent gain at the receiver. | 143 |
| Figure 6.4: Experimental results showing the coherent summation of the primary and secondary node signals for the case where the two-tone transmitter M2 was displaced. The transmitted signals were connected to the oscilloscope via cable and the data was recorded for different displacement distances. From [14] © 2021 IEEE. | 144 |
| Figure 6.5: Coherent beamforming results showing the combined signal power of the two transmitted signals for multiple displacement distances of the primary node transmitter M1. From [14] © 2021 IEEE. | 145 |
| Figure 6.6: Coherent beamforming results showing the combined signal power of the two transmitted signals for multiple displacement distances of the primary node. From [14] © 2021 IEEE. | 146 |
| Figure 6.7: (a) Time domain representation of the localization waveform. (b) Frequency content of the localization waveform. | 148 |
| Figure 6.8: Schematic of the distributed array used in this study, consisting of a primary and secondary node, beamforming to an arbitrary angle. | 149 |
| Figure 6.9: 50,000 Monte Carlo simulation showing various threshold values for achieving coherent gain. The probabilities were estimated as a function of angle and localization standard deviation. The strictest requirements for localization uncertainty appear at 90°. | 150 |
| Figure 6.10: Schematics showing the connections and setup of the distributed nodes and target. This setup was used for the wireless beamforming experiment. From [15] © 2021 IEEE. | 150 |
| Figure 6.11: Open-loop coherent distributed array setup showing the target, primary, and secondary nodes. The antennas in this setup were connected to multiple SDRs, an Oscillator, and a frequency locking circuit. From [15] © 2021 IEEE. | 151 |
| Figure 6.12: Experimental results showing the coherent beamforming results for the case where the secondary node was displaced. In this scenario, the signals were transmitted using log-periodic antennas, and their were received using a horn antenna. From [15] © 2021 IEEE. | 153 |

Figure 7.1: Diagram showing the proposed TOA experiment, where the secondary node localizes the primary node using TTSFW signals. From [1] © 2021 IEEE. . . . 155

Figure 7.2: Experimental setup. From [1] © 2021 IEEE. 158

Figure 7.3: Square root of the summation of lower bounds for estimating x and y coordinates of the target at multiple locations. From [1] © 2021 IEEE. 158

Figure 7.4: Ten thousand Monte Carlo simulations showing the average achievable coherent gain for 1 GHz coherent signals using the CRLB results for the localization estimates. Arrays of 2 and 6 nodes were analyzed. From [1] © 2021 IEEE. 159

LIST OF ALGORITHMS

| | |
|--------------------------------|----|
| Algorithm 5.1: NL-LS | 89 |
| Algorithm 5.2: MF-LS | 90 |

CHAPTER 1

INTRODUCTION

Wireless technologies have advanced dramatically in the past few years allowing distributed cooperation between separate subsystems, leading towards enabling coherent distributed operations. Cooperation at the wavelength level of the wireless operation enables a multitude of subsystems to transmit and/or receive coherent signals, forming a coherent distributed array. Coherent distributed arrays, such as the one shown in Fig. 1.1, operate in essence as distributed phased arrays where the signal power gain is relative to the number of transmitters squared multiplied by the number of receivers. In contrast to single-platform phased arrays [16, 17], receiving coherent distributed arrays operate comparably to element-level digital phased arrays, where signals are collected by the analog-to-digital converter (ADC) at each element and then are combined in processing [18, 19]. Coherent distributed arrays introduce a paradigm shift to the current focus on wireless systems that rely on platform-centric antennas [20, 21]. Improvements to platform-centric systems focus on enhancing the characteristics of the system rather than scaling it by combining multiple independent elements, this leads to bulky, costly, inefficient systems that are prone to a single element failure and are hard to conceal or relocate. Coherent distributed arrays are scalable in that the power gain can be adjusted by adding or removing elements from the array [22, 23]. They also prevent single point-of-failure problems, where it is possible to remove or shut down one element without perturbing the operation of the array. In addition, selected nodes can be relocated to adapt to any changes and to cover various areas and various applications, where the nodes can be disaggregated into multiple sub-categories that perform different tasks when more than one operation is required. Applications of distributed beamforming include satellite-satellite and satellite-terrestrial communications [24], surveillance and information transfer using a drone swarm [25–27], indoor and outdoor fifth generation (5G) communications [28, 29], among others.

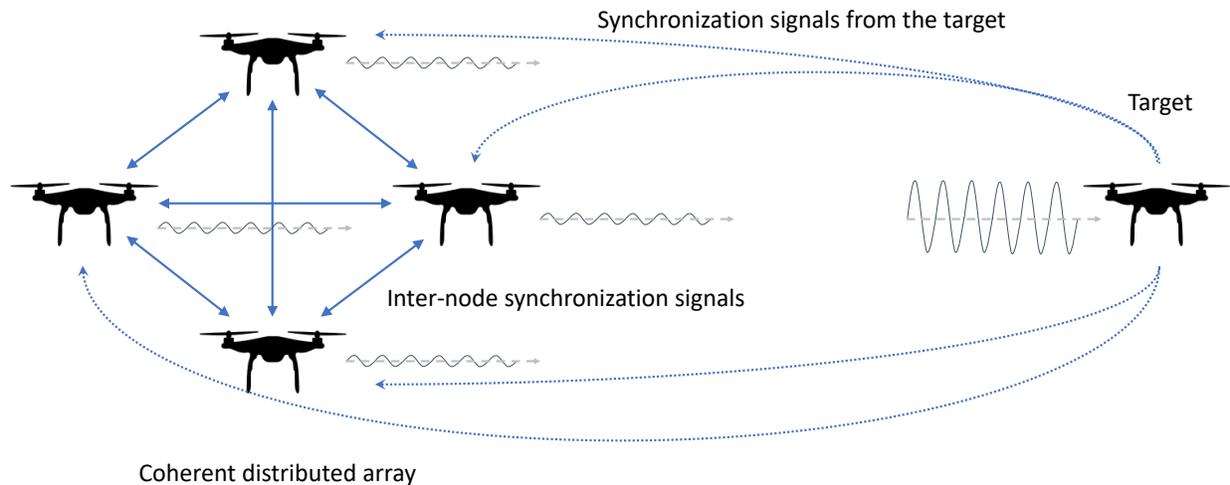


Figure 1.1: General architecture of coherent distributed arrays where the nodes achieve synchronization using inter-node signals and external signals from the targeted location. Once all the nodes are synchronized, the transmitted signals add coherently at the target and provide an increased signal power.

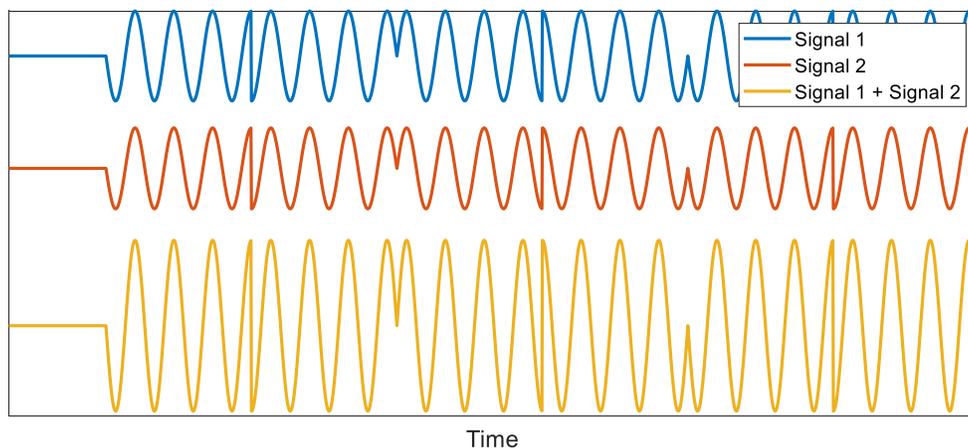


Figure 1.2: Two ideal received signals along with their summation from two perfectly synchronized nodes.

1.1 Coordination of Principal Parameters

Accurate coordination of disaggregated nodes is crucial to the operation of the distributed array. The principal electrical states that require synchronization are: time [30–33], phase [2, 34–39] and frequency [9, 14, 34, 35, 40]. Fig. 1.2 shows the individual representation along with the summation of two ideal received signals from two perfectly synchronized nodes. This shows how the amplitude of the received signals doubles which leads to a multiplicative gain for the received

power.

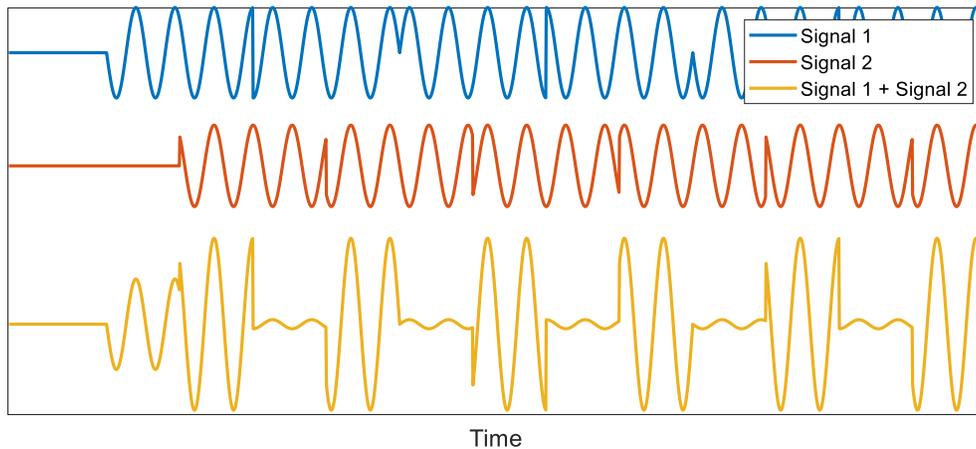


Figure 1.3: Two ideal received signals along with their summation from two nodes performing phase and frequency synchronization without time alignment.

1.1.1 Time Alignment

Coherent distributed antenna arrays require appropriate time alignment so that the pulses or symbols overlap sufficiently in order to maximize the transmitted or received power. Generally, time alignment requires the least precision compared to phase alignment and frequency synchronization since the requirements for time alignment are imposed by the information or symbol rate rather than the transmitted frequency. Nevertheless, timing errors of no more than nanoseconds or sometimes picoseconds are required for wideband signals with hundreds of MHz of bandwidth [41]. Fig. 1.3 shows the individual representation along with the summation of two ideal received signals from two nodes with only phase and frequency synchronization. Timing is significantly misaligned in this case, resulting in a distorted signal summation that can prevent the receiver from correctly estimating the received symbols.

The timing on every node is generated using internal clocks that are controlled by one or multiple oscillators. Oscillators, if not well synchronized, tend to drift over time. Even for synchronized oscillators, time alignment is necessary to remove any constant phase shift that could arise from the startup of the system, the addition of a node, or from motion induced time shifts. A simplistic

model describing the clock shift in a 2 node system is given by [42]

$$C_1(t) = a \cdot C_2(t) + b, \quad (1.1)$$

where C_1 and C_2 represent the clock time of nodes 1 and 2, a represents the relative time drift, and b is the relative offset between the two clocks. In an ideal scenario, we should have $a = 1$ and $b = 0$.

Time alignment in general is feasible using Global Navigation Satellite System (GNSS) and Network Time Protocol (NTP). Nevertheless such systems cannot always offer the desired accuracy, especially for the applications of coherent distributed arrays. In addition, time alignment using GNSS signals is not always a viable option since satellite signals are not detectable in many locations such as in indoor applications.

Various time alignment techniques have been developed to improve the synchronization accuracy and reduce the cost, hardware, and complexity observed in the previously mentioned techniques. Initially, the time alignment algorithms focused on transmitting timestamps from one node to the other such as in The Timing-Sync Protocol for Sensor Networks (TPSN) [43], Tiny-Sync (TS) and Mini-Sync (MS) [44], among others. Timestamp free approaches have been recently introduced to improve the accuracy, simplicity, and flexibility. Also, the approaches that rely on timestamps exchange use the medium access control (MAC) layer to process the synchronization signals which provide timing errors on the order of microseconds [42]. Moving the functionality to the physical layer (PHY) increases the synchronization speed and accuracy dramatically, which could be done for timestamp-free approaches such as in [45–47]. Standard deviations of less than 100 ps were observed in [41] while implementing a timestamp-free synchronization approach that uses linear frequency modulated (LFM) pulses with 1 μ s pulse width and 40 MHz bandwidth over a 1.44 GHz carrier.

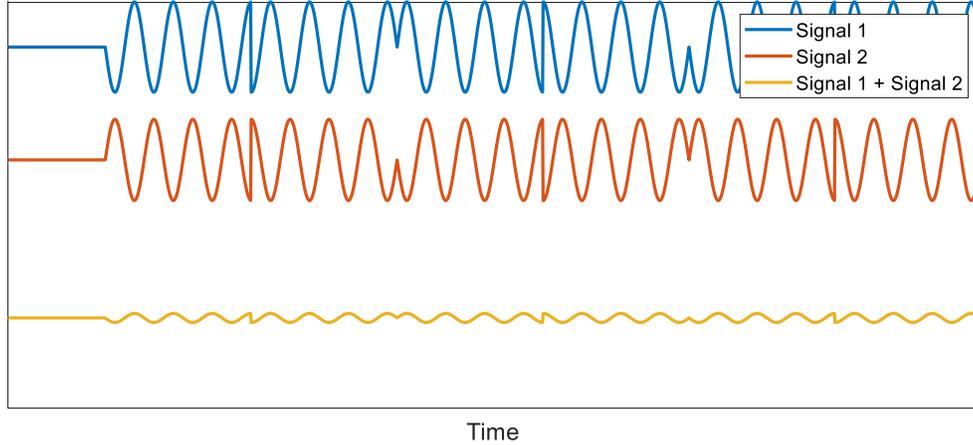


Figure 1.4: Two ideal received signals along with their summation from two nodes performing time and frequency synchronization without phase alignment.

1.1.2 Phase Alignment

Phase alignment is the most challenging task in a coherent distributed array compared to the synchronization of the other electrical states since it is in general harder to achieve in open-loop systems and it requires the errors to be more than one order of magnitude less than the transmitted wavelength. It was shown in [48] that no more than 18° root-mean-square error (RMSE) for phase offset of the received signals is permitted in order to have a high probability to achieve 90% coherent gain, where the coherent gain represents the received power at the target over the maximum achievable power. The effect of the signal summation with phase offset is shown in Fig. 1.4. As can be seen, the phase offset between the two transmitted signals resulted in signal summation with low amplitude/power.

Phase alignment has been achieved using multiple architectures, the majority relying on feedback signals from the target [2, 36–39], while others were achieved using inter-node synchronization signals [14]. In general, before implementing phase synchronization, the nodes need to be frequency locked/synchronized so that relative phase shifts stay constant throughout the calibration process. Furthermore, less complicated synchronization approaches are needed for static nodes, on the other hand, in a dynamic array, which is the general case, fast algorithms are needed so that the phases can continuously align.

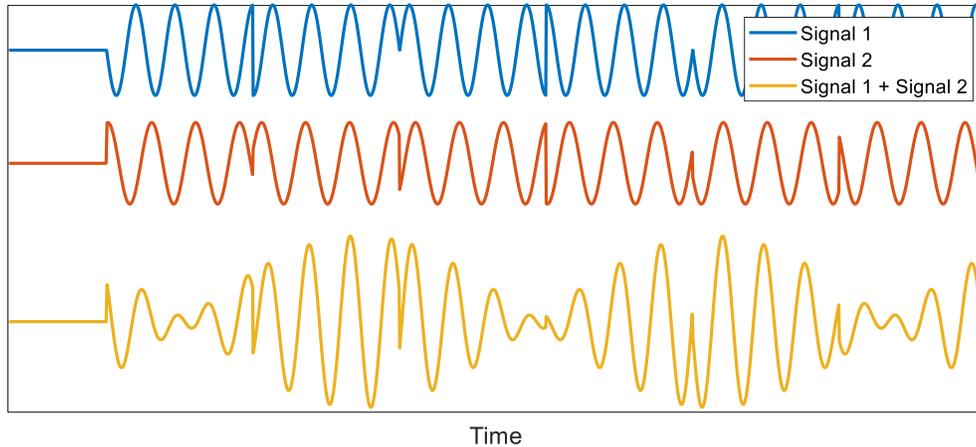


Figure 1.5: Two ideal received signals along with their summation from two nodes performing time and phase synchronization without frequency synchronization.

1.1.3 Frequency locking

Frequency synchronization is essential for distributed systems since the nodes transmit signals stemming from local oscillators. Oscillators in general tend to operate at a slight offset from the nominal frequency, drift over time, and might have high phase noise that degrades the performance of a coherent distributed array. Fig. 1.5 depicts the effect of having two local oscillators operating with a frequency mismatch. By combining the two transmitted signals with frequency offset, the amplitude of the combined signals was changing sinusoidally, where at some instances the amplitude was equal to the summation of both transmitted signal amplitudes and at other the amplitude was close to zero. Thus frequency synchronization is essential to ensure a coherent summation of the signals at the targeted location. knowing that the nodes are scattered and have no physical connection, frequency synchronization has to be implemented wirelessly. Wireless frequency synchronization is common for wireless systems and it is commonly used for orthogonal frequency division multiple access (OFDMA) to ensure orthogonality and suppress multiple access and inter-channel interferences [49].

In coherent distributed arrays frequency synchronization is necessary as no time or phase alignment is possible unless all the nodes operate on the same frequency. A common frequency synchronization approach is to assign to each node a time slot which is used to compare the relative

phase [50] or frequency [51] to a reference. In [50], the measured phases were fed to a Kalman filter to predict the phase and frequency shifts and correct them. For instance, in [52], frequency and phase were monitored based on the received power at the target, necessary frequency and phase adjustments were made to ensure that the power stays above a certain threshold.

Other forms of frequency synchronization where no time slots are required include the use of coupled-oscillators [53]. Nevertheless, coupled-oscillators under-perform when nodes are widely separated and they are limited to close range operation. Another option would be to use optically-locked voltage controlled oscillators [54], which would suffer as well from long range synchronization where pointing, tracking, and acquisition require special hardware on each node. On the other hand, long range frequency synchronization is achievable using GNSS signals such as Global Positioning System (GPS) [55]. Yet, as mentioned earlier, GNSS signals are not always available and even when available they tend to introduce a high phase noise that prevents the distributed array from operating coherently, especially at high frequencies.

Centralized approaches where frequency synchronization is achievable using inter-node signals was realized in [56]. Secondary nodes that were represented by either software-defined radios (SDRs) or other similar hardware were equipped with phase-locked loops (PLLs) that can lock the frequency of the local oscillator using an incoming frequency reference signal. In this approach the primary node had to transmit to the secondary nodes a continuous wave (CW) with a frequency equal to that of the desired reference frequency signal in order to lock their local oscillators. This approach was deemed impractical since the transmitted frequency was limited to only one option, which is almost always a very low frequency. This approach was improved in [57] by cascading two PLLs where the first one was used to scale the received frequency to the desired reference frequency using multipliers and dividers. This approach improved the flexibility but still transmitting one tone was prone to high interference levels. In [9, 14, 40] the secondary nodes were synchronized using two-tone signals and an adjunct self-mixing circuit that extracts the reference frequency before feeding it the PLL. In general, the discussed centralized approaches make it easier to align the phase of the distributed nodes for multiple beamforming architectures, however,

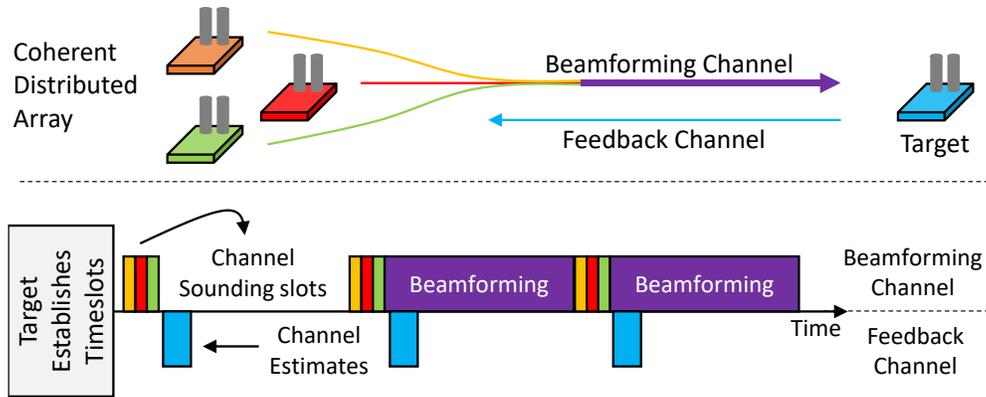


Figure 1.6: Receiver-coordinated beamforming using explicit-feedback (adapted from [2]). The receiver allocates the channel sounding period time slots along with the beamforming time slot. The nodes transmit an uncompensated carrier in their allocated sounding slot and the target transmits back the feedback signals afterwards to synchronize their phase and frequency. Once synchronized the nodes beamform at the dedicated beamforming slot. From [3] © 2021 IEEE.

they are prone to one-element failure and they might be inefficient for widely spread nodes.

A decentralized wireless frequency approach was introduced in [58], where the nodes in an array synchronize their frequency by reaching a final average frequency among all the nodes. In this approach not all nodes have to be connected, and even with one or multiple element failures, it is always possible to synchronize the frequency of the elements in an array as long as every node has at least one connection to the remaining nodes. Yet, decentralized approaches might present some impediments for phase and time synchronization. For instance, the initial phase and frequency synchronization duration tend to be higher than the duration required to synchronize centralized systems, as many synchronization cycles are needed (compared to one cycle for many centralized architectures) to reach a consensus on the desired phase and frequency.

1.2 Architectures of Coherent Distributed Arrays

Coherent distributed arrays are divided into either closed-loop or open-loop architectures. This classification is dependent on the approach used to synchronize the nodes in the array; when a feedback from an external system, typically the target, is used to synchronize the array, the architecture is referred to as closed loop. This category dominates the coherent distributed array literature as it is relatively easy to apply. Multiple closed-loop approaches, where a coherent feedback from

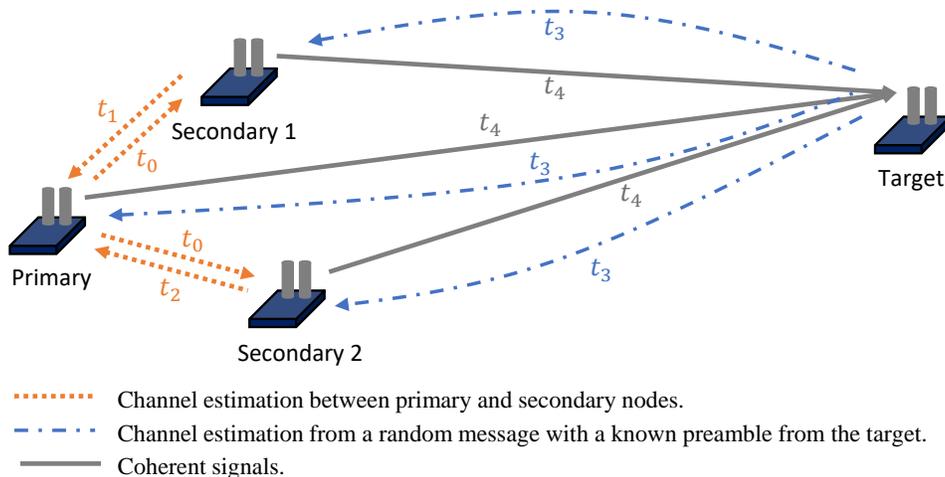


Figure 1.7: Retrodirective beamforming where time slots are assigned so that the nodes can synchronize their frequencies to that of the target. The channels are estimated between the nodes and the target, a phase offset is applied so that the secondary nodes are phase aligned to the phase of the primary node. From [3] © 2021 IEEE.

the target is used, were developed, such as: one-bit feedback [36], roundtrip synchronization [59], primary-secondary synchronization [56], and two-way synchronization [60], receiver-coordinated explicit-feedback [2], to name a few. For instance, in [2] a time slotted approach was implemented to synchronize the carrier phases of the transmitting nodes using a feedback from the receiver as illustrated in Fig. 1.6. At first, the target establishes the channel sounding and beamforming time slots. Every node transmits its uncompensated carrier at its dedicated sounding time slot and the receiver determines the phase offset at the carrier frequency. The receiver transmits back the estimated phases and every node estimates its phase and frequency offsets using a Kalman filter. Afterwards, in the beamforming time slot, phase and frequency offsets are added for the transmitters of every node before beamforming at the target/receiver. This approach was tested using a CW signal at 910 MHz over a 1 km distance of separation between the coherent distributed array and the target/receive.

Furthermore, a closed-loop system can be sub-categorized as a retrodirective distributed beamformer. Retrodirective beamforming covers the techniques that rely on a feedback from the target to synchronize the array, but the received signal is not cooperative as is the case for [2,36,56,59,60]. By analyzing incoming signals from the target or external devices, the channel gains or frequency

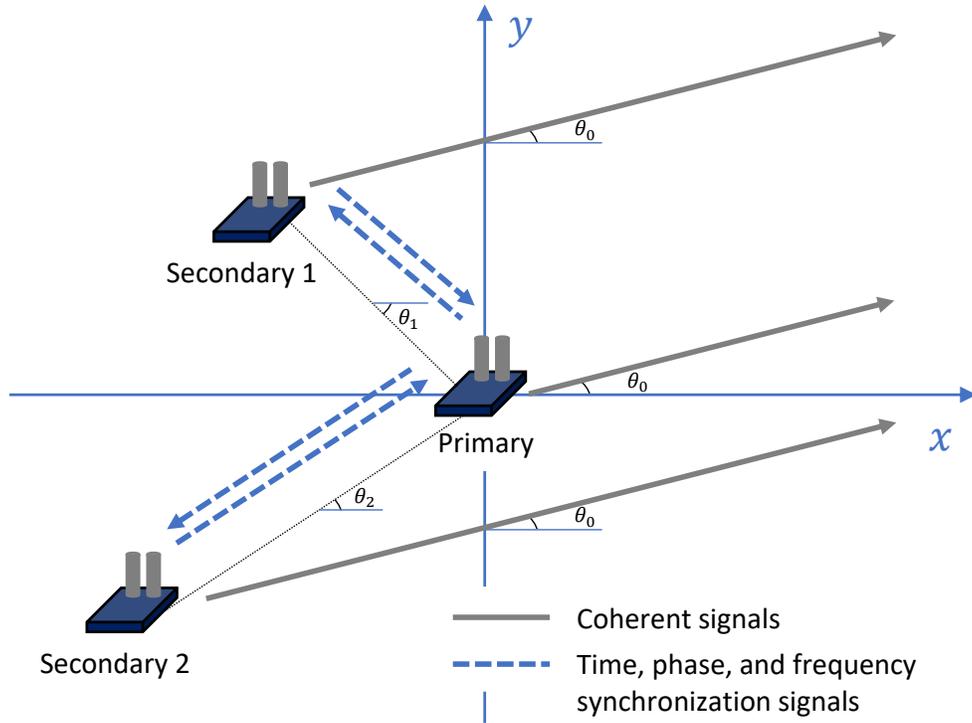


Figure 1.8: Open-loop distributed phased arrays beamforming at an angle θ_0 in the far field. Time, phase, and frequency are synchronized using internode signaling in a centralized architecture. Localizing the primary and/or the secondary nodes is important to align the phases.

is adjusted. As an example, retrodirective array would synchronize its frequency by synchronizing the local oscillators to a strong signal incoming from an external device. Pon array [61] is a form of retrodirective coherent distributed array where reciprocity is used to mitigate the propagation phase delays and frequency offsets. In [37], stationary nodes were synchronized using a hierarchical model, where at first, the primary node had a dedicated time-slot, afterwards the secondary nodes were each assigned a time slot with the target having the final time slot before beamforming could take place as shown in Fig. 1.7. Initially, the secondary nodes synchronize their frequency to that of the primary node and afterwards they estimate their channel parameters. Afterwards, all the nodes calibrate their phases using the non-cooperative broadcast signal from the target. Once all the nodes are calibrated, beamforming is possible. It was possible to achieve a beamforming gain higher than 90% of the ideal beamforming gain after 10 s of synchronization period. Other examples of retrodirective coherent distributed arrays are available in [62, 63].

Closed-loop architectures, whether feed-back enabled or retrodirective, are limited to communications applications where an external system, usually the target/receiver, has to assist the nodes to synchronize their local electrical states. Furthermore, closed-loop architectures cannot coherently beamform to arbitrary directions since they require feedback from the desired transmission direction.

Contrary to closed-loop approaches, open-loop coherent distributed arrays [14,48] have no dependency on the target or any external systems. This expands the application space of open-loop topologies to include radar and remote sensing applications and other communications applications where the receiver does not have the means to transmit feedback signals. Also, open-loop coherent distributed arrays can beamform to arbitrary directions since they have no dependency on the location of the target. As open-loop coherent distributed arrays synchronize their electrical states within the array, accurate node localization is crucial since phase alignment relies on the relative positions of the nodes in the array. Coherent distributed arrays in general and open-loop coherent distributed arrays specifically can be grouped as either centralized or decentralized systems. Centralized open-loop systems, such as what is shown in Fig. 1.8, are easier to synchronize as all the electrical states can be synchronized based on a reference node, which is usually referred to as primary or master node [14,64]. Nevertheless, such architectures can suffer from single-point failures, unless there is a mechanism that allows other secondary nodes to be reassigned as a primary node. Decentralized architectures are an alternative to centralized systems, where the operations in the array are divided equally among all the nodes. On the other hand, phase synchronization in decentralized open-loop coherent distributed arrays is challenging, and so far it was not achieved yet. Nevertheless, synchronization of other electrical states such as frequency locking has been demonstrated in a decentralized approach [58,65] which opens the door for future decentralized open-loop coherent distributed arrays.

1.3 Main Contribution of this Dissertation

This dissertation focuses on analyzing and implementing centralized open-loop coherent distributed antenna arrays. First, I analyze the localization requirements for phase synchronization in Chapter 2. As mentioned earlier, phase synchronization in open-loop approaches is mainly achieved by localizing the nodes in the array. In a centralized topology, the nodes synchronize their phases relatively to that of the primary node. Phase alignment is possible by tracking the relative movement of the nodes; the secondary nodes should at least localize the primary node to track the phase shift induced from relative motion or other factors. Based on the desired frequency of operation, the ranging accuracy must meet certain thresholds to achieve a high coherent gain. Chapter 3 focuses on the frequency synchronization intervals that allow the coherent operation of the array. The analysis is focused on frequency synchronization for nodes equipped with PLLs. Chapter 4 describes the centralized frequency synchronization approach that was implemented using adjunct self-mixing circuits and PLLs internal to the SDRs representing the nodes in the array. The phase shift produced by the self-mixing circuit in a dynamic array is analyzed. Accurate range estimation using two-tone waveforms is analyzed and demonstrated in Chapter 5. An adaptive framework that minimizes the spectral footprint (bandwidth) while maintaining the desired ranging accuracy to support beamforming operations is shown. Proof-of-concept open-loop beamforming experiments are shown in Chapter 6 along with the achieved coherent gain. Chapter 7 discusses a localization technique that enables two-dimensional tracking of the primary node and relates the obtained localization accuracies to the achievable coherent gain.

The principal contributions of this work include the following:

- Synchronization of open-loop coherent distributed arrays is challenging, moreover, the synchronization performance has a direct effect on the coherent gain of the system. In this dissertation, I analyze thoroughly the desired RMSE for ranging to achieve certain thresholds of coherent gain for the case where wireless frequency synchronization induces phase shifts in a dynamic array. Furthermore, I extend the analysis to localization where the an-

gle of arrival needs to be estimated. In addition to the analysis on ranging and localization, I analyze the desired frequency and phase stability for multiple generic voltage controlled oscillators (VCOs), and I present a generalized analysis that helps identifying the desired frequency synchronization rate for a multitude of case scenarios.

- I present an adaptive ranging approach that supports distributed beamforming over a long range. The adaptive ranging waveform is based on a spectrally sparse two-tone waveform with a variable bandwidth. A proportional–integral (PI) controller, which is a special case of the proportional–integral–derivative (PID) controller [66, 67] is employed to adaptively modify the ranging waveform. Cooperative ranging was performed between two nodes over a 90 m separation in an outdoor setup. The two cooperative nodes were wirelessly frequency locked and it was possible to achieve and maintain a ranging standard deviation of 1 cm for a one week period. This adaptive ranging approach minimized the spectral footprint, where, the bandwidth of the ranging waveform was minimized while achieving the desired ranging accuracy. The maximum tone separation of the two-tone ranging signals was set to only 7.5 MHz.
- The first fully wireless open-loop coherent distributed array proof-of-concept is demonstrated experimentally in this dissertation. Coherent beamforming of CW signals was possible by synchronizing the phase and frequency of the distributed nodes in a centralized architecture. The secondary node(s) can synchronize its/their frequency using the presented adjunct self-mixing circuit. Using the presented ranging approach, the phase shifts produced from both moving the node transmitters and moving the wireless frequency synchronization antennas were tracked and corrected for. Beamforming at 1.5 GHz was possible with at least 0.9 coherent gain for most of the captured data.
- An accurate two-dimensional localization approaches based on time of arrival (TOA) that could enable beamforming in the future was implemented. The achieved localization accuracies were analyzed in terms of the maximum achievable coherent frequency with high

coherent gain.

CHAPTER 2

LOCALIZATION REQUIREMENTS FOR OPEN-LOOP COHERENT DISTRIBUTED ARRAYS

In this chapter I discuss the coherent gain, a common metric for evaluating the performance of coherent distributed arrays, and I analyze the localization accuracies needed to achieve high levels of power gain. First, the requirements for ranging accuracy are analyzed separately from angle estimation in centralized open-loop topologies, which would be the case if the angles of the secondary nodes relatively to the primary node are given. Afterwards, an analysis on the requirements for localization is presented.

2.1 Coherent Gain

The performance of distributed beamforming operation can be characterized by evaluating the signal-to-noise ratio (SNR) or bit error rate (BER) of the received signal; other application-specific metrics can be used for evaluation as well. Nevertheless, evaluating the power gain in the main-beam of the beamformed signals where phase, frequency, and timing errors are present, in comparison to the ideal case where no degradation in the signals summation is observed, is a straightforward and general approach that can be used to evaluate the general performance of the distributed antenna array. In particular, I evaluate the coherent gain G_c with the definition from [48] using

$$G_c(t) = \frac{|s_r(t)s_r(t)^*|}{|s_i(t)s_i(t)^*|}, \quad (2.1)$$

where $s_r(t) = \sum_{n=0}^{N-1} s_{r,n}(t)$ is the summation of the N received signals with instantaneous phase and timing errors, and $s_i(t) = \sum_{n=0}^{N-1} s_{i,n}(t)$ represents the ideal summation of the signals where no phase or timing errors are present. Although $G_c(t)$ is time dependent, in the remainder of this dissertation it is represented by G_c . The coherent gain is bounded by $0 \leq G_c \leq 1$, where the lower limit represents the case where a totally destructive addition of the signals is observed, and the upper limit 1 represents a transmission with perfect phase and time alignment. A general

representation of $s_{i,n}(t)$ is as follows

$$s_{i,n}(t) = h_n(t)A_n(t)e^{j2\pi f_c t}, \quad (2.2)$$

where $h_n(t)$ is the complex valued coefficient for the n th propagation channel, $A_n(t)$ is the amplitude and modulation of the n th signal, and f_c is the carrier frequency used by the transmitting nodes. $s_{r,n}(t)$ is expressed similarly to $s_{i,n}(t)$ with additional phase, frequency, and timing errors. The general expression of $s_{r,n}(t)$ is represented by

$$s_{r,n}(t) = h_n(t)A_n((t - \tau_n(t))e^{j2\pi[f_c + \delta f_n(t)]t + j[\delta\phi_n(t) + \phi_{s,n}(t) - \phi_{c,n}(t)]}, \quad (2.3)$$

where $\tau_n(t)$ represents the time delay error that can have both a constant shift produced by initial time offsets in the system or the calibration process, and a variable shift produced by the time drift of internal clocks. In the majority of the cases, the time delay errors are not considered (set to 0) since this dissertation focuses on phase and frequency synchronization rather than time alignment. In practice, the timing errors are suppressed by transmitting to the target CW signals with no underlying modulation. $\delta f_n(t)$ represents the frequency offset, $\delta\phi_n(t)$ combines the instantaneous frequency and phase errors resulting from multiple software-, and hardware-related factors, $\phi_{s,n}(t)$ is the phase shift produced by the motion of the nodes, constant synchronization errors, multipath, among other factors, and $\phi_{c,n}(t)$ represents the correction applied to the phase shift which can be obtained through initial calibration and node localization for open-loop topologies so that a continuous correction for the observed phase shifts can be applied. For closed-loop topologies, $\phi_{c,n}(t)$ is generated using the feedback from the target. Other sources of error can be added to (2.3) depending on the desired application. The majority of the parameters that were introduced or are going to be introduced in the future are time dependent, however, the time variable t is going to be omitted from these parameters from now on for the sake of simplicity.

2.2 Ranging Requirements for Centralized Open-Loop Coherent Distributed Arrays

As seen from (2.3), there are multiple compounded errors that contribute to the degradation of the coherent gain. In the case where time and frequency synchronization are perfectly achieved,

hardware induced errors are not considered, and weather and external errors such as multipath are suppressed, one can evaluate the accuracy of phase alignment approaches on the coherent gain by considering the phase changes $\phi_{s,n}(t)$ in a dynamic array and the applied phase correction $\phi_{c,n}(t)$ which includes errors.

Since this dissertation focuses on centralized open-loop coherent distributed arrays, in such topologies, phase alignment can be achieved by localizing the nodes in the array and then applying the desired phase correction $\phi_{c,n}(t)$ that cancels the observed phase shift $\phi_{s,n}(t)$. Localization is achieved in a centralized manner where the secondary nodes need to localize themselves relatively to the location of the primary node. This way, it is possible to correct for the anticipated relative phase shift between the primary node and a secondary node n . Localization of the primary node can be achieved using both ranging [68–72] and angle of arrival (AOA) [73–76] or using multilateration approaches such as TOA [77–79], time difference of arrival (TDOA) [80–84], and frequency difference of arrival (FDOA) [85–87]. As a first step, one can evaluate the effect of ranging accuracy on the coherent gain by assuming that the AOA is given. This can relate in practice to an array with nodes distributed in one dimension.

2.2.1 Ranging Requirements with Wired Frequency Synchronization

To evaluate the impact of errors in a direct manner, frequency synchronization can be achieved using wired connections such as in Fig. 2.1. This way, the requirements for ranging accuracy are relaxed in comparison to wireless frequency synchronization as it will be shown in Section 2.2.2. Wired frequency synchronization can be achieved by connecting a signal with the desired reference frequency to the PLL of the nodes in the array. The reference signal can be generated from the oscillators of the primary node, and it can be either a sinusoidal wave or a square wave with typical frequencies of 10 MHz or 100 MHz.

In the case where no wireless phase locking/frequency synchronization is implemented, the main source for relative phase offset between the transmitted coherent signals from every primary/secondary pair is the relative displacement of the transmitters on these two nodes. In a hier-

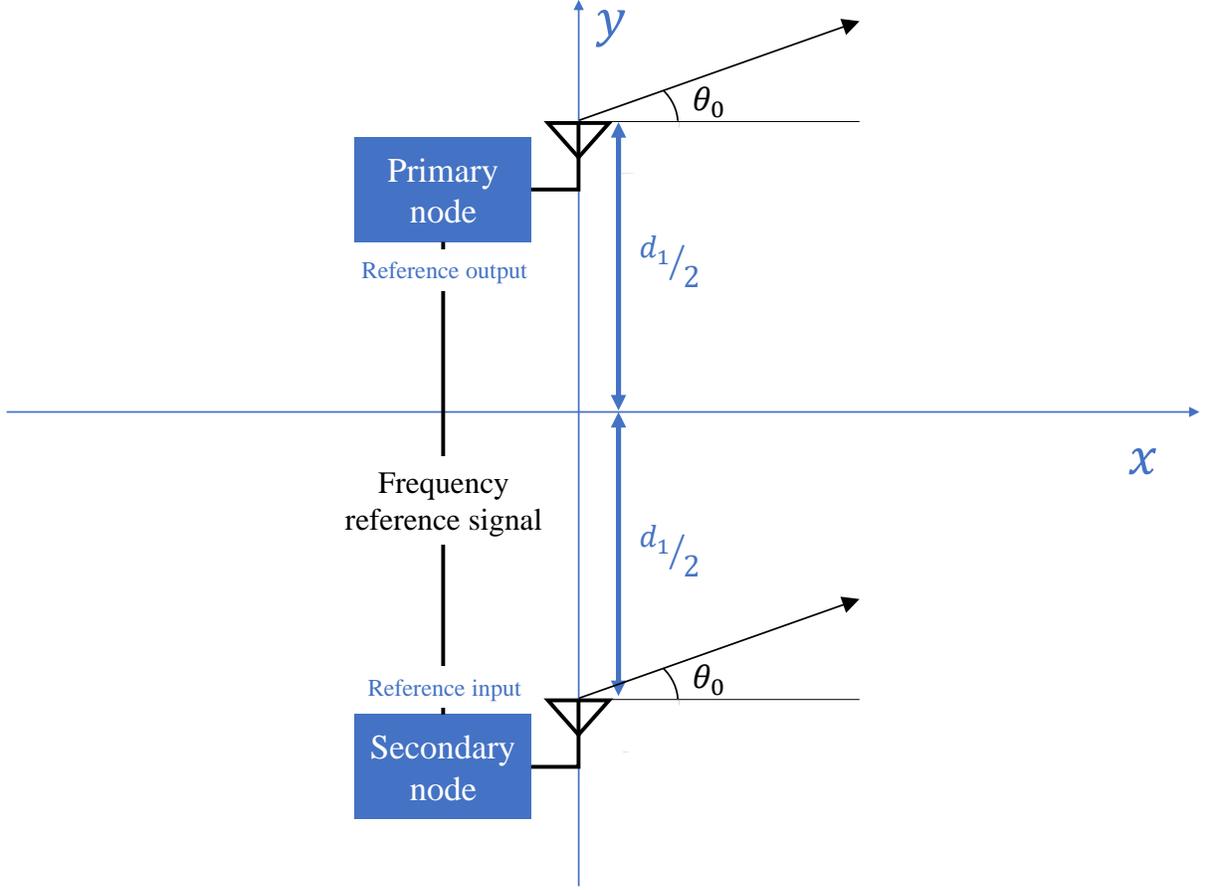


Figure 2.1: Two nodes with a hierarchical architecture beamforming at a target in the far field. Frequency synchronization is achieved using a cabled connection, where the primary node transmits a reference frequency signal to lock the PLL of the secondary node. Beamforming is done by accounting for the phase shift produced by the relative motion between the two transmitters.

archical topology, the phase synchronization can be achieved in a pairwise manner, where every secondary node estimates its relative phase shift due to any relative motion between the primary/secondary pair. The motion-induced phase shift observed at the target from the transmitted signals of the secondary node n is obtained from

$$\Delta\phi_{s,1,n} = -\frac{2\pi\Delta d_{1,n}\sin(\theta_0)}{\lambda_c}, \quad (2.4)$$

where $d_{1,n}$ is the distance separating the coherent transmitter of the secondary nodes from the coherent transmitter of the primary node, $\Delta d_{1,n}$ is the relative displacement of the coherent transmitters, θ_0 is the beamsteering angle at which beamforming is achieved in the far field, and λ_c is

the wavelength of the coherent signal. The phase shift $\Delta\phi_{c,1,n}$, which represents the estimated version of the actual phase shift $\Delta\phi_{s,1,n}$, is used to adjust the transmitted phase. In general, the total estimated phase shift is represented with $\phi_{c,n}$; $\phi_{c,n}$ is a combination of all estimated phase shifts due to the motion of the secondary node n along with the initial phase shift at the calibration stage. The desired output phase from node n is obtained by subtracting the phase $\phi_{c,n}$ from the transmitted signals. For a given relative angle between every primary/secondary pair, and a desired θ_0 , it is possible to evaluate the desired ranging accuracy in order to ensure a certain level of coherent gain. The probability to achieve a certain level of coherent gain $P(G_c \geq X)$ is evaluated in terms of ranging accuracy, where $0 \leq X \leq 1$ is the fraction of the ideal coherent signal power.

The probability $P(G_c \geq X)$ can be estimated using Monte Carlo simulations, where $\Delta d_{1,n}$ and the angle of the primary node in comparison to the secondary node are uniformly distributed over a wide distance and all the possible angles. Ten thousand Monte Carlo simulations were generated where h_n , $A_n((t-\tau_n))$, δf_n , and $\delta\phi_n$ were set to 1, 1, 0, and 0 respectively. By doing so, it is possible to evaluate the effect of ranging accuracy while suppressing all the other error factors. The ranging accuracy is evaluated in terms of root-mean-square error (RMSE) and it is expressed in terms of λ_c . Expressing the ranging RMSE in terms of λ_c makes it possible to evaluate $P(G_c \geq X)$ for any frequency by just multiplying the RMSE value by the wavelength of interest. In this analysis, X was set to 0.9, which represents only 0.5 dB degradation in the maximum achievable power at the targeted location, making it a reasonable threshold to target. Fig. 2.2 shows the Monte Carlo simulation results for 2, 3, 10, and 1,000 nodes in a coherent distributed array. As it can be seen, by increasing the number of nodes in the array, the separation between coherent and incoherent transmission becomes more clear and if a probability of 90% to achieve at least 0.9 coherent gain is considered ($P(G_c \geq 0.9) = 90\%$), the ranging RMSE σ_d has to be at most 0.0667λ ($\lambda_c/15$) for large arrays.

As seen in Fig. 2.2, errors in range estimates are the most forgiving for two-node arrays. Analysis on the coherent operations of two nodes can be of interest for multistatic or bistatic systems. A further analysis on the desired ranging accuracy for two-node arrays is shown in Fig.

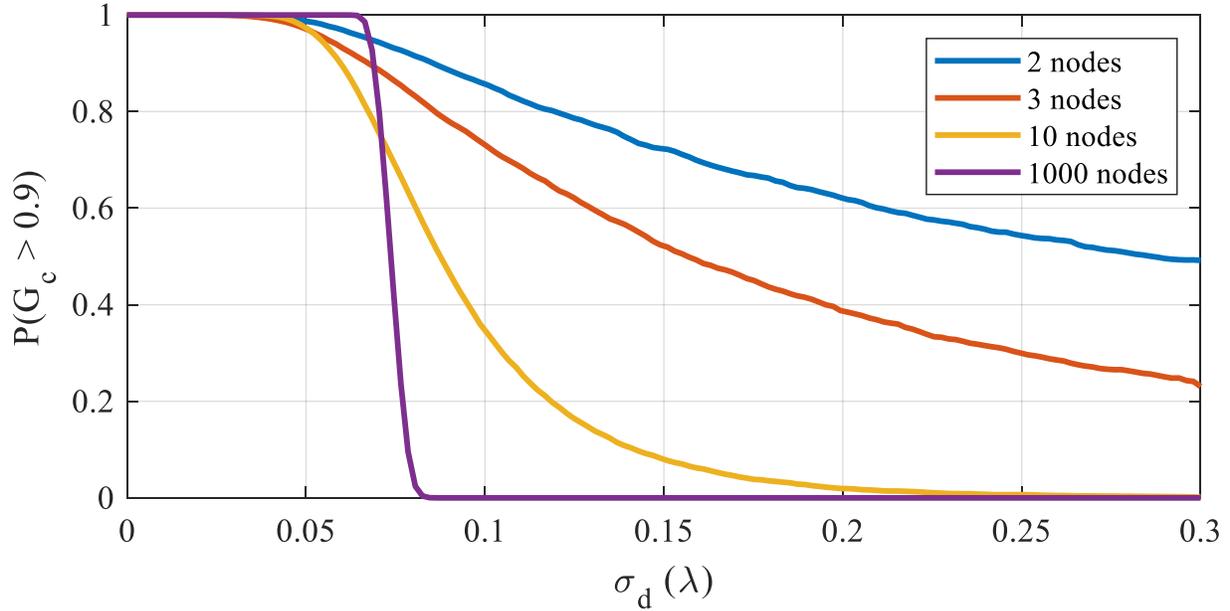


Figure 2.2: Probability of the coherent gain exceeding 0.9 versus ranging RMSE σ_d in terms of the signal wavelength λ considering the effects of phase changes where no wireless frequency synchronization is considered.

2.3, where the ranging RMSE is evaluated for all the transmission angles θ_0 and multiple values for $P(G_c \geq 0.9)$. The same settings as earlier were used for the Monte Carlo simulations. As it can be seen, and as it can be analyzed from (2.4), the angles 0° and 180° are the least affected by the degradation in ranging accuracy, while the areas around the angles 90° and 270° have more stringent constraints on σ_d .

2.2.2 Ranging Requirements with Wireless Frequency Synchronization

So far, the analysis on the desired ranging accuracy was focused on systems that do not require wireless frequency synchronization. In practice, coherent distributed arrays have no physical connections, and frequency synchronization is only achievable wirelessly. As it will be seen in the next chapters, phase tracking is challenging for open-loop architectures, and it is necessary to implement a wireless frequency synchronization approach that enables the secondary nodes to track their relative phase shift to that of the primary node, such as in [9, 14, 40]. In general, instantaneous frequency shifts produce constant phase shifts, this issue can be prevented by relying on phase

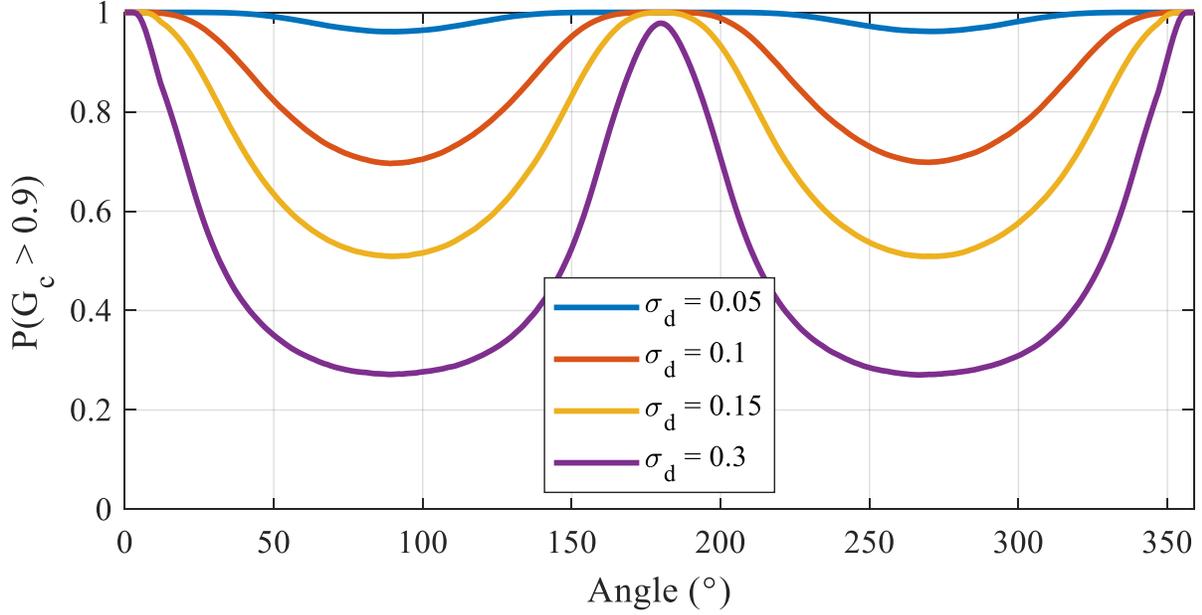


Figure 2.3: Probability of the coherent gain exceeding 0.9 versus the transmission orientation in the case where frequency synchronization does not contribute to the phase errors. From [4] © 2020 IEEE.

locking approaches such as PLLs, where the phase shifts of the oscillators of a node are a result of the phase shifts observed at the input reference frequency signals. As it will be demonstrated later, the phase shift at every secondary node is relative to the distance of displacement of the secondary node relative to the primary node. For a secondary/primary pair such as in Fig. 2.4, the phase shift is expressed as

$$\Delta\phi_{s,2,n} = -\frac{2\pi\Delta d_{2,n}}{\lambda_c}, \quad (2.5)$$

where $d_{2,n}$ is the distance separating the synchronization antennas of the secondary node and primary nodes, $\Delta d_{2,n}$ is the relative displacement of the synchronization antennas. $\Delta\phi_{s,1,n}$ and $\Delta\phi_{s,2,n}$ are both additive phase shifts and the total phase shift that needs to be accounted for in case of any relative motion is expressed as

$$\Delta\phi_{s,n} = \Delta\phi_{s,1,n} + \Delta\phi_{s,2,n}. \quad (2.6)$$

The effect of (2.6) is analyzed in Fig. 2.5, where 10,000 Monte Carlo simulations were evaluated for 2, 3, 10, and 1,000 nodes in a coherent distributed array. The same analysis and settings as in Section 2.2.1 were used. The same distance of displacement was selected for $\Delta d_{2,n}$ and $\Delta d_{2,n}$.

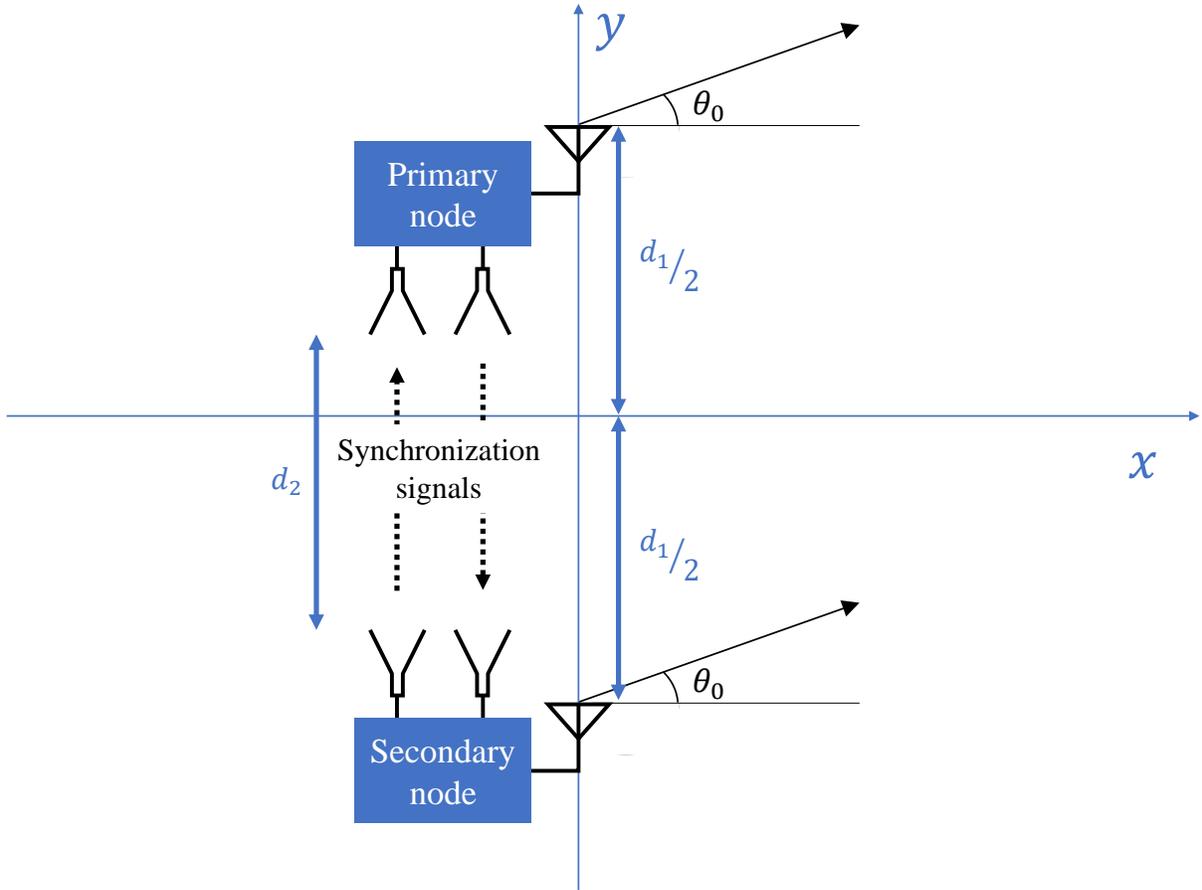


Figure 2.4: Two nodes with a hierarchical architecture beamforming at a target in the far field. Frequency synchronization is achieved wirelessly, where the secondary node locks its oscillator phase to that of the incoming synchronization signals from the primary node. Beamforming is done by accounting for the phase shift produced by the relative motion between the two transmitters and between the synchronization antennas. From [4] © 2020 IEEE.

The effect of adding wireless frequency synchronization is clear in that the curves representing the probability to achieve at least 0.9 coherent gain were pushed to the left side; this shows that more stringent ranging accuracies are needed in order to transmit coherently. As can be seen, for the case of 1,000 nodes (more stringent constraints), a 90% probability to achieve at least 0.9 coherent gain is reached when σ_d is at most 0.0385λ ($\lambda_c/27$).

An analysis on the desired ranging accuracy for beamforming at a specific angle θ_0 is studied. This analysis is similar to what was analyzed in Section 2.2.1, Fig. 2.3, however, the extra phase shift produced by moving the synchronization antennas is evaluated in this section. The results

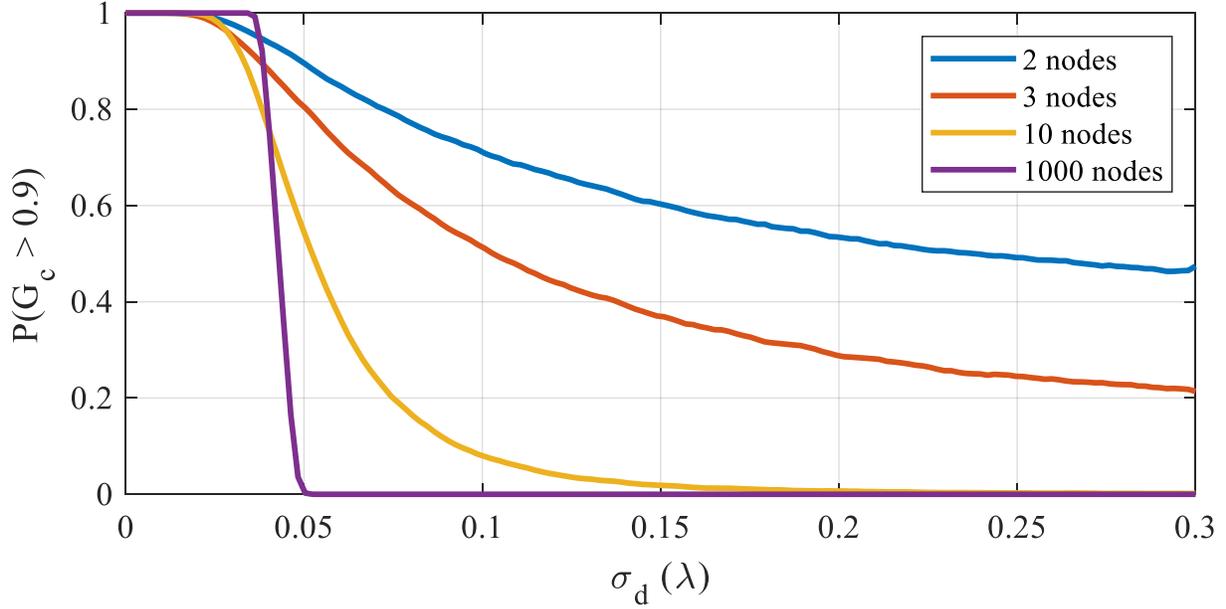


Figure 2.5: Probability of the coherent gain exceeding 0.9 versus ranging RMSE σ_d in terms of the signal wavelength λ considering the effects of phase changes where wireless frequency synchronization is present (wireless frequency synchronization using phase locking approaches). From [4] © 2020 IEEE.

are shown in Fig. 2.6. It can be seen that in contrast to the case where no wireless frequency synchronization is considered, the area around 90° requires more stringent ranging accuracies. This is caused by the added phase shift $\Delta\phi_{s,2,n}$. For the angles around 90° , the value of the sine function in (2.5) is close to $+1$, which leads to an additional phase shift in (2.6). Thus, any change in relative distance has a higher effect on the phase shift for the transmitted signals once (2.5) is considered. On the other hand, in the region around 270° , the value of the sine function in (2.5) is close to -1 . This effect negates (fully at $\theta_0 = 270^\circ$) the phase shift observed in (2.4), making any phase shift due to the small displacement distances negligible.

A similar analysis that shows the desired ranging accuracy versus both, the transmission angle and multiple $P(G_c \geq X)$ options is shown in Fig. 2.7. The curves representing a 90% probability to achieve at least 0.6, 0.7, 0.8, and 0.9 coherent gain are shown. The results are correlated to what is observed in Fig. 2.6. By knowing the transmission angle and the achievable ranging accuracy, one can estimate the achievable coherent gain. Nevertheless, these analyses, consider perfect frequency and time alignment, and exact knowledge of the orientation of every node, which

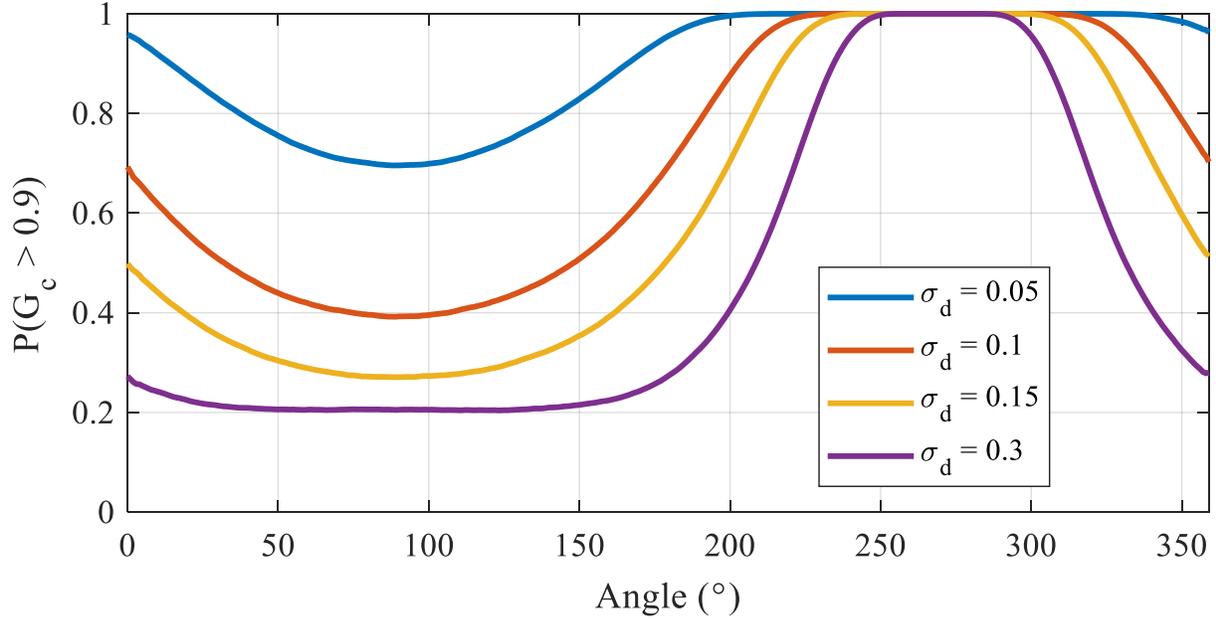


Figure 2.6: Probability of the coherent gain exceeding 0.9 versus the transmission orientation in the case where wireless frequency synchronization contributes to the phase errors. From [4] © 2020 IEEE.

is not the case in practice. Nevertheless, this allows us to know the maximum achievable coherent gain, and the other error sources can be added as compounded errors in the future.

Now that we have seen that the phase shift produced by the motion of the wireless frequency synchronization antennas has both an additive and a subtractive effect on the phase shift produced by the motion of the coherent transmitters, one can expect to have an inverted effect if the primary node was locking its frequency to that of the secondary node (usually the secondary nodes lock their frequency to that of the primary node). This effect is depicted in Fig. 2.8, where it can be seen that a mirrored effect to the one shown in Fig. 2.6 is obtained. Thus, the wireless frequency synchronization in a two-node system can be performed adaptively where the wireless frequency circuit can be activated either on the first or on the second node depending on the desired transmission orientation to maximize the beamforming performance and minimize the ranging requirements.

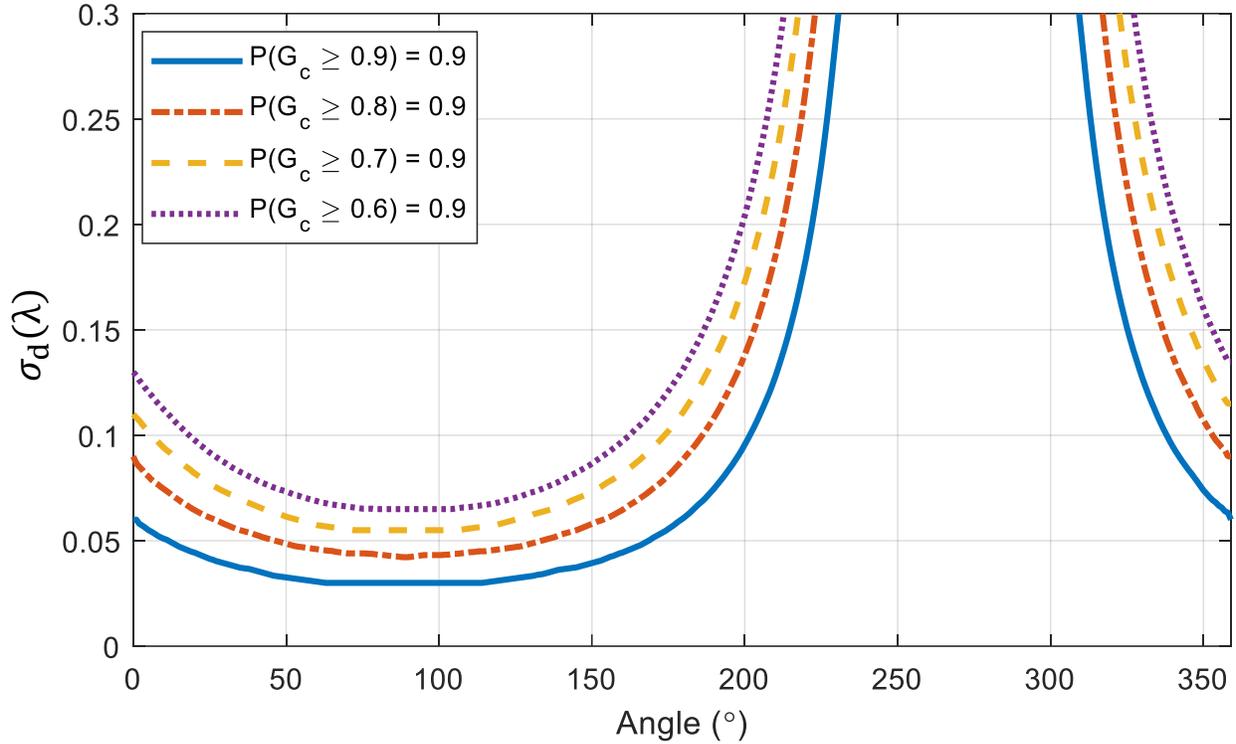


Figure 2.7: Fifty thousand Monte Carlo simulation showing various threshold values for achieving coherent gain. The probabilities were estimated as a function of angle and localization standard deviation. The strictest requirements for localization uncertainty appear around 90° .

2.3 Localization Requirements for Centralized Open-Loop Coherent Distributed Arrays

So far, it was always assumed that the orientation of the primary node in comparison to that secondary node n is perfectly known. In practice, estimating the AOA accurately can be much more challenging than having accurate range estimates. Furthermore, the errors in angle estimation have a multiplicative effect for far nodes as it is going to be shown later in this section.

2.3.1 Localization using Range and Angle of Arrival

To evaluate the effect of angle estimation, the setup in Fig. 2.4 is updated to the one in Fig. 2.9, where the angles between the primary node and other secondary nodes is clearly shown. Also,

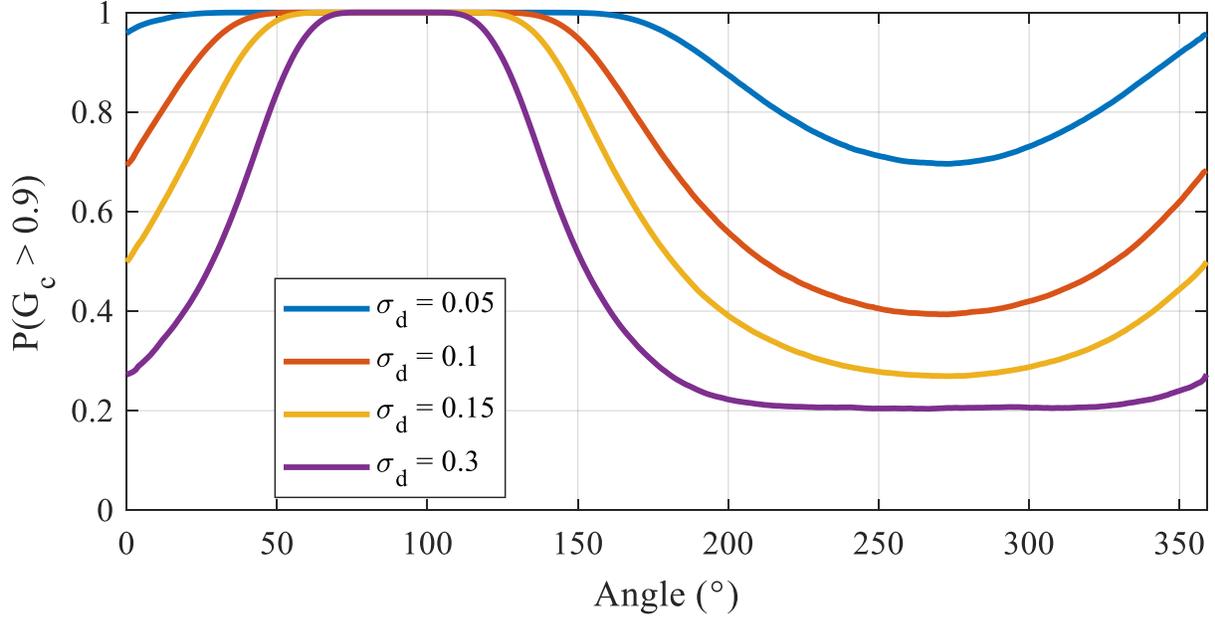


Figure 2.8: Probability of the coherent gain exceeding 0.9 versus the transmission orientation in the case where wireless frequency synchronization contributes to the phase errors. These results consider the case where the primary node locks its frequency to that of the secondary node in a two-node array. From [4] © 2020 IEEE.

(2.4) is updated to reflect this change as follows

$$\Delta\phi_{s,3,n} = -\frac{2\pi\Delta d_{1,n}\sin(\theta_0 + \theta_n)}{\lambda_c}. \quad (2.7)$$

It is important to mention that the angles θ_n were already present in the earlier sections, nevertheless their value was embedded with the value of θ_0 . For this new problem formulation, the RMSE of the angle estimates is represented with σ_θ .

To easily visualize the effect of angle errors on localization in general, it is possible to perform a variable transformation for the RMSE of range and angle to RMSE of x and y coordinates. The x and y coordinates of a primary node in comparison to the coordinates of a secondary node n are obtained from

$$x_n = d_n \cos(\theta_n), \quad (2.8)$$

$$y_n = d_n \sin(\theta_n), \quad (2.9)$$

where d_n is the distance between the center of the primary node and the center of a secondary node n . Transformation of the RMSE is done by assuming that the estimated values of d_n and θ_n

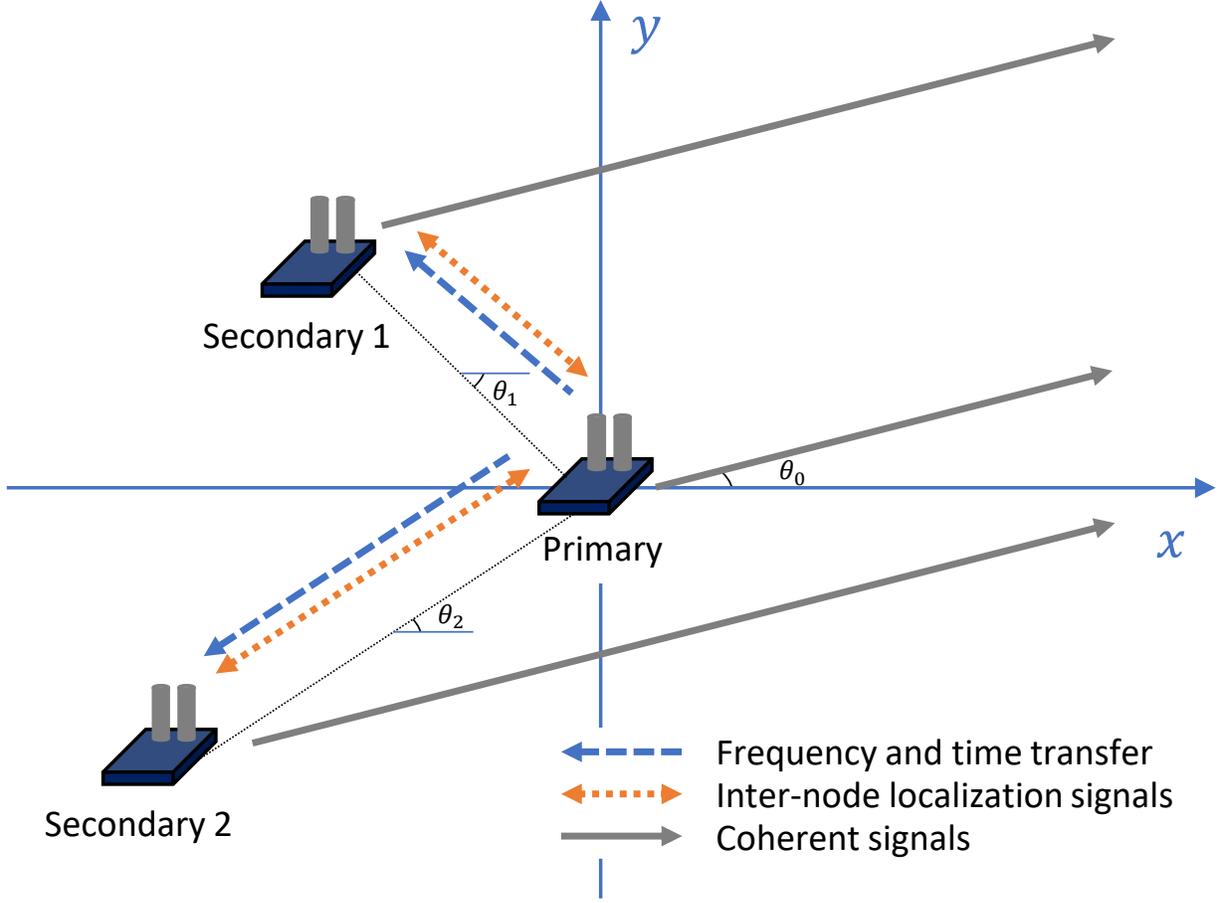


Figure 2.9: Hierarchical open-loop coherent distributed array beamforming at an angle θ_0 . The angles θ_1 and θ_2 represent the angles of the primary node relative to the planes of the secondary nodes. From [5] © 2021 IEEE.

are close to their mean, such that a function $f(d_n, \theta_n)$ can be approximated using the expansion in first order Taylor's series of the random variables d_n and θ_n [88]. It is then possible to express the deviations of each sample $f_i(d_n, \theta_n)$ of a random variable $f(d_n, \theta_n)$ in terms of $d_{n,i}$ and θ_n :

$$f_i(d_n, \theta_n) - \bar{f}(d_n, \theta_n) \approx (d_{n,i} - \bar{d}_n) \frac{\partial f(d_n, \theta_n)}{\partial d_n} + (\theta_{n,i} - \bar{\theta}_n) \frac{\partial f(d_n, \theta_n)}{\partial \theta_n}, \quad (2.10)$$

where $\bar{\cdot}$ represents the mean of a variable. Variance transformation (and consequently RMSE) is done using

$$\sigma_{f(d_n, \theta_n)}^2 \approx \sigma_{d_n}^2 \left(\frac{\partial f(d_n, \theta_n)}{\partial d_n} \right)^2 + \sigma_{\theta_n}^2 \left(\frac{\partial f(d_n, \theta_n)}{\partial \theta_n} \right)^2 + 2\sigma_{d_n \theta_n}^2 \left(\frac{\partial f(d_n, \theta_n)}{\partial d_n} \right) \left(\frac{\partial f(d_n, \theta_n)}{\partial \theta_n} \right), \quad (2.11)$$

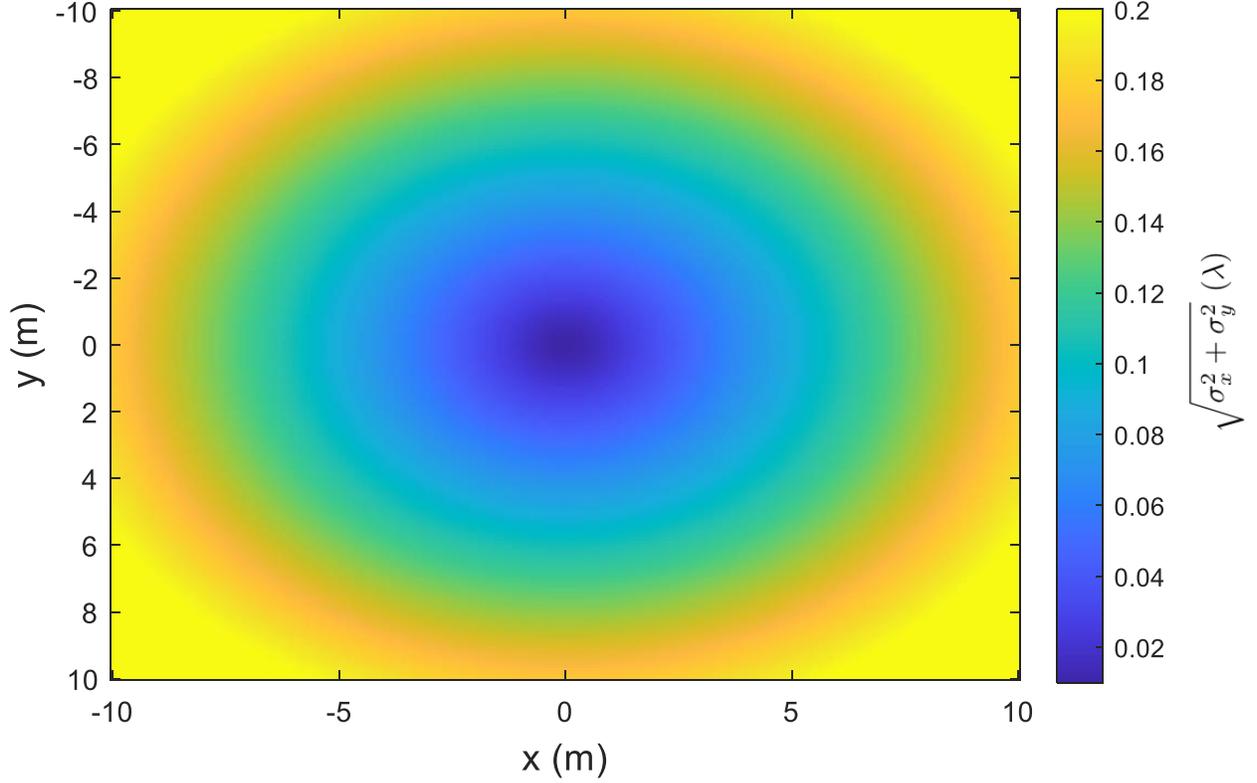


Figure 2.10: Radial error $\sqrt{\sigma_{x_n}^2 + \sigma_{y_n}^2}$ in terms of λ for maximum and minimum coordinates of ± 10 m for x and y in the case of $\sigma_{d_n} = 0.01\lambda$ and $\sigma_{\theta_n} = 1^\circ/\text{m}$. From [5] © 2021 IEEE.

where $\sigma_{d_n\theta_n}^2$ is the covariance between the variables d_n and θ_n . In our case, we consider that d_n and θ_n are uncorrelated, making $\sigma_{d_n\theta_n}^2 = 0$. Similarly, the covariance between two functions $f_2(d_n, \theta_n)$ and $f_1(d_n, \theta_n)$ can be calculated using [89]

$$\sigma_{f_1(d_n, \theta_n)f_2(d_n, \theta_n)}^2 \approx \sigma_{d_n}^2 \left(\frac{\partial f_1(d_n, \theta_n)}{\partial d_n} \right) \left(\frac{\partial f_2(d_n, \theta_n)}{\partial d_n} \right) + \sigma_{\theta_n}^2 \left(\frac{\partial f_1(d_n, \theta_n)}{\partial \theta_n} \right) \left(\frac{\partial f_2(d_n, \theta_n)}{\partial \theta_n} \right). \quad (2.12)$$

Using (2.11), σ_{x_n} and σ_{y_n} , the variances of x_n and y_n , are obtained from

$$\sigma_{x_n}^2 = \sigma_{d_n}^2 \cos^2(\theta_n) + \sigma_{\theta_n}^2 d_n^2 \sin^2(\theta_n), \quad (2.13)$$

$$\sigma_{y_n}^2 = \sigma_{d_n}^2 \sin^2(\theta_n) + \sigma_{\theta_n}^2 d_n^2 \cos^2(\theta_n). \quad (2.14)$$

Also, the covariance between x_n and y_n can be calculated from

$$\sigma_{x_n y_n}^2 = (\sigma_{d_n}^2 - \sigma_{\theta_n}^2 d_n^2) \sin(\theta_n) \cos(\theta_n). \quad (2.15)$$

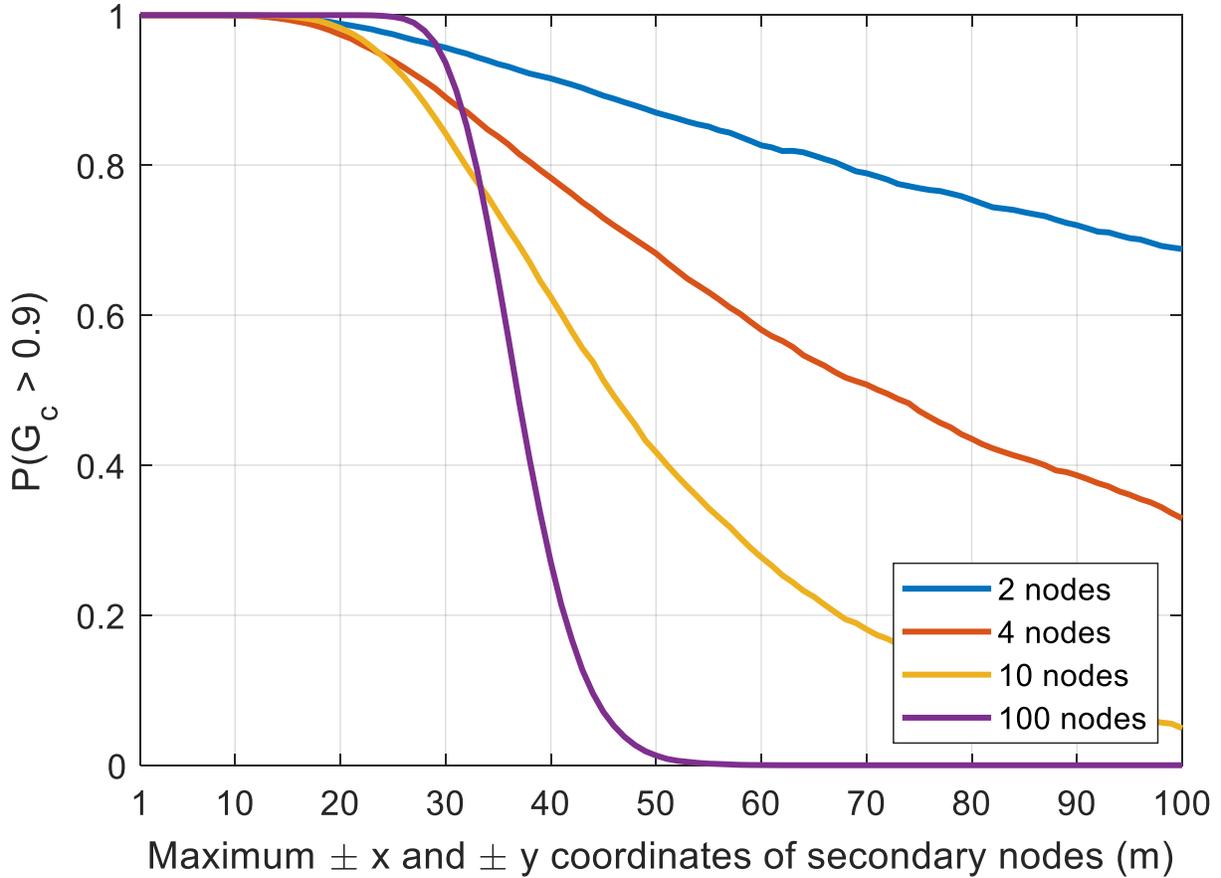


Figure 2.11: Probability of the coherent gain to exceed 0.9 versus the maximum allowed x and y coordinates, where $\sigma_{d_n} = 0.01\lambda$ and $\sigma_{\theta_n} = 0.1^\circ/\text{m}$. From [5] © 2021 IEEE.

The effect of angle RMSE is shown in Fig. 2.10, where it can be seen that at larger distances, the accuracy of x and y estimates worsens dramatically. This effect is due to the RMSE of the angle estimates, as when the nodes are far apart the error becomes multiplicative. In Fig. 2.10, the RMSE of range and angle estimates is expressed in terms of the wavelength of the transmitted coherent signals, this is done to help the reader link the earlier findings to the figures in this section.

As it can be deduced from Fig. 2.10, evaluating the required RMSE for range and angle estimation is going to be dependent on the spread area of the nodes; angle errors translate to larger positional errors at longer distances; the effect of the nodes spreading on the desired ranging accuracy is minimal in comparison to angle estimation, as it is going to be shown later in this section. The maximum achievable coherent gain for N nodes, where $N-1$ secondary nodes localize the primary node to adjust their relative phases, was illustrated in Fig. 2.11, where $\sigma_{d_n} = \sigma_{d_{1,n}} = \sigma_{d_{2,n}} = 0.01\lambda$

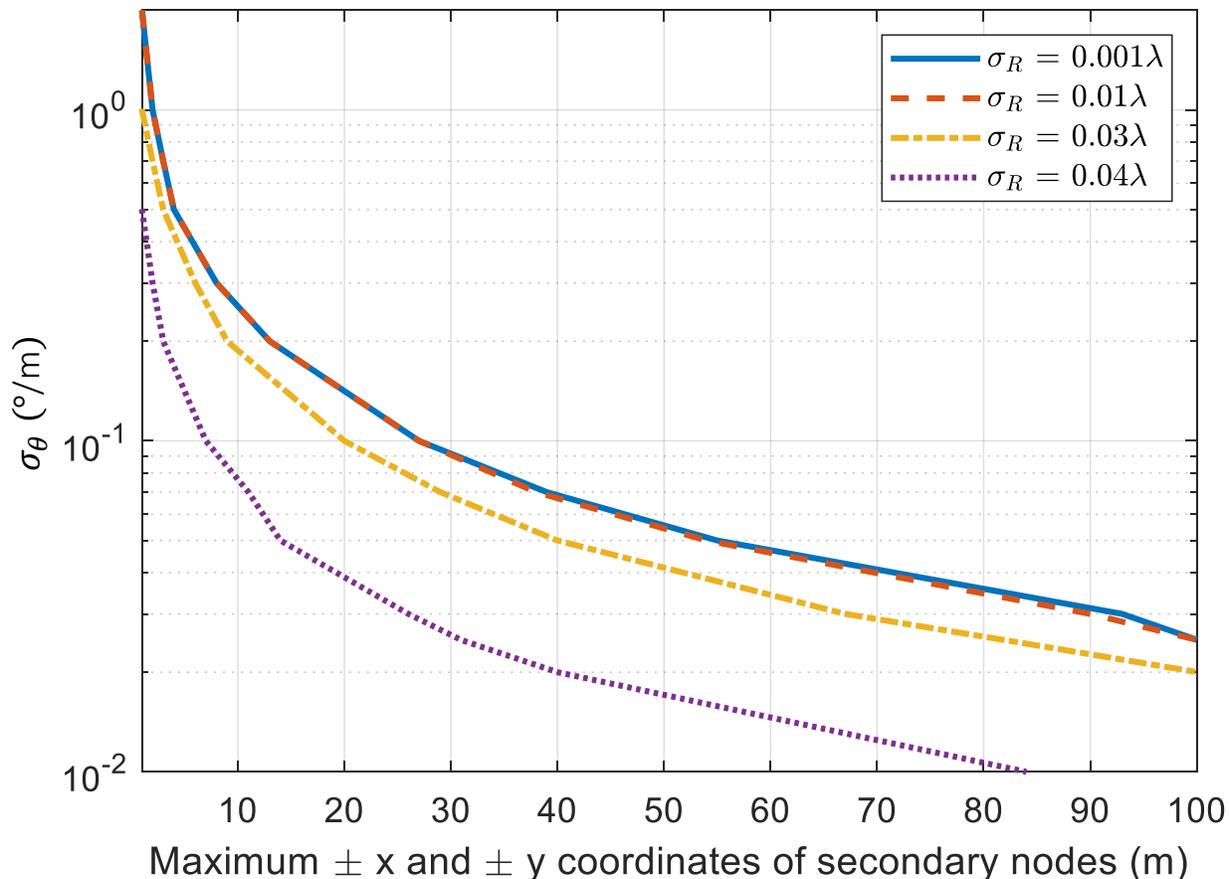


Figure 2.12: The required σ_θ to maintain a 0.9 coherent gain with 90% probability for 10 nodes with multiple values of σ_d versus the maximum allowed coordinates for x and y . From [5] © 2021 IEEE.

and $\sigma_{\theta_n} = 0.1^\circ/m$. A minimum of 0.5 m separation between the primary node and any secondary node in x and y was imposed, and a maximum area of $100 \times 100 \text{ m}^2$ for the spread of the nodes was investigated. 10,000 Monte Carlos Simulations were generated where $\Delta d_{1,n} = \Delta d_{2,n}$, θ_n , and θ_0 were selected as random variables with uniform distribution over the entire area.

Fig. 2.12 expands on Fig. 2.11, where it shows the required angle estimation RMSE values for multiple RMSE values of range estimates in a 10 node array. the metric $P(G_c \geq 0.9) = 90\%$ is chosen. An area up to a maximum of $100 \times 100 \text{ m}^2$ was considered. It can be seen that with $\sigma_{d_n} \leq 0.01\lambda$, the requirements for σ_{θ_n} are similar since the bottleneck in this case is σ_{θ_n} . On the other hand, for $\sigma_{d_n} \geq 0.045\lambda$ it was not possible to maintain a coherent gain of 0.9 with a 90% probability for any value of σ_{θ_n} . This shows that angle estimation accuracy is generally the limiting

factor for the beamforming performance, especially when the nodes are scattered over a large area. For very large areas, the desired values for the RMSE of σ_{θ_n} are very small in that they cannot be achieved in practice. Thus, if the nodes are sparse, a localization approach that does not rely on angle estimation is needed.

2.3.2 Localization using Multilateration

Multilateration is a localization technique that is based on comparing the estimated distances or time of arrival of the localization pulses from multiple receivers (or receivers and transmitters) to localize the target. TOA and TDOA are examples of multilateration where the time of arrival or time difference of arrival of certain signals is used to localize the target. The advantage of such systems over the systems represented in Section 2.3.1 is that extremely accurate angle estimates, which are in practice hard to achieve, are not required. Nevertheless, the localization accuracy is linked to the time delay estimation accuracy and the spread area of the transmitters/receivers or the geometry of the array. Distributed Arrays where the secondary nodes localize the primary node using both TOA and TDOA approaches are shown in Fig. 2.13. Generally, for TOA/TDOA systems, the receivers (secondary nodes in our case) which are usually referred to as anchors, should have known locations. The transmitter (primary node in our case) is localized by processing its transmitted localization signals and comparing the time of arrival at the secondary nodes. The same transmitted signal can be used for frequency synchronization, time alignment, and localization (subsequently phase alignment) which offers a great advantage. Nevertheless, in practice, the locations of the secondary nodes are unknown as well. In Chapter 7, A modified version of TDOA/TOA is shown where the locations of the secondary nodes do not need to be known. The general time delay equation for target localization using TOA is [80]

$$t_i = \frac{[(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2]^{1/2}}{c}, \quad (2.16)$$

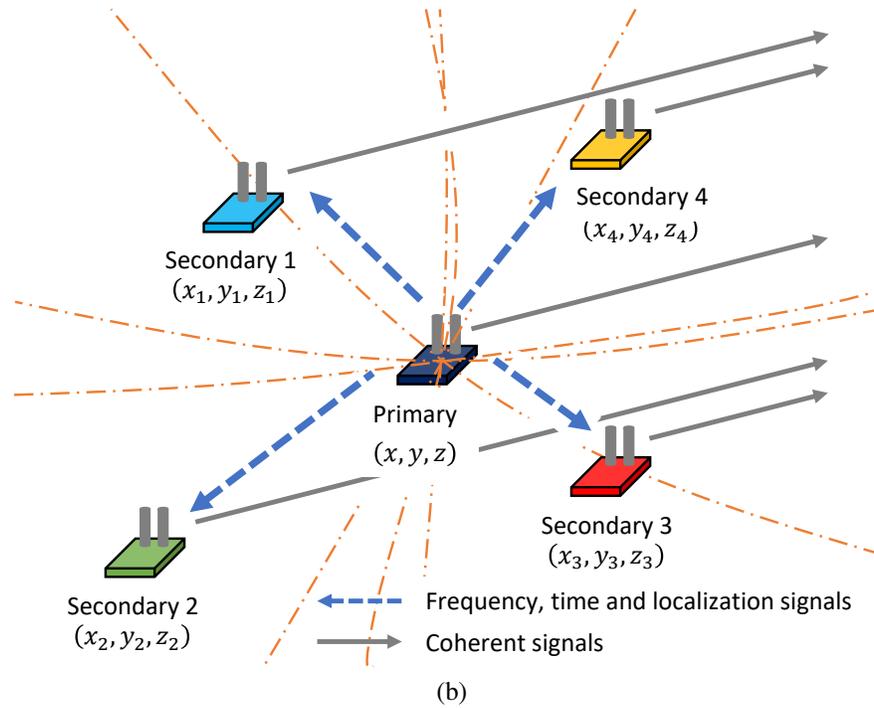
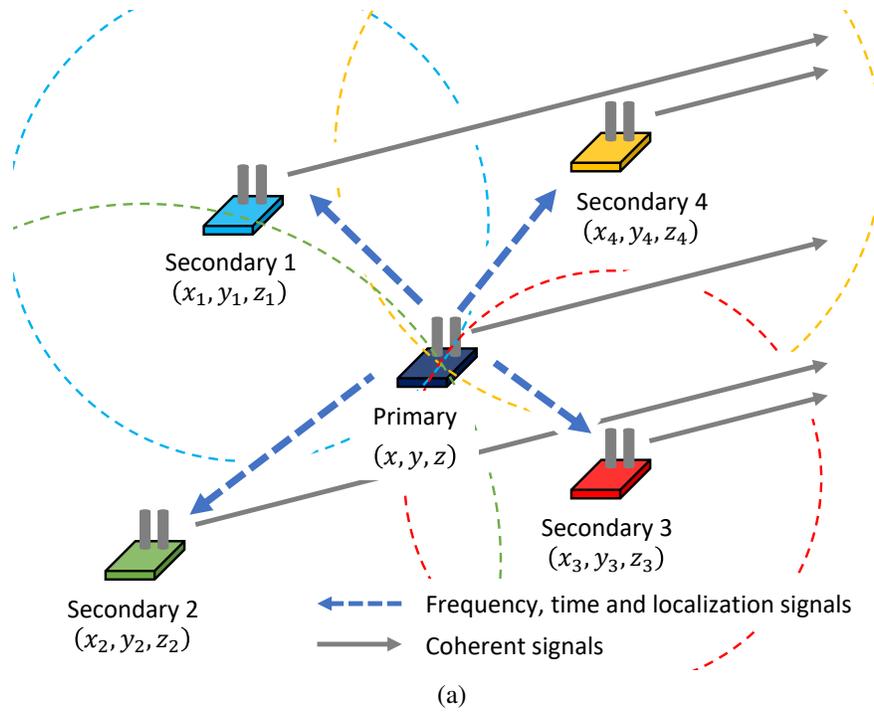


Figure 2.13: Hierarchical open-loop coherent distributed arrays that rely on multilateration using (a) TOA and (b) TDOA measurements with circles (in (a)) and hyperbolas (representing the time difference of arrival between every pair of nodes in (b)) respectively to estimate the location of the primary node and synchronize the phase of the secondary nodes.

and the equation for time difference in TDOA is

$$\Delta t_{ij} = \frac{[(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2]^{1/2}}{c} - \frac{[(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2]^{1/2}}{c}, \quad (2.17)$$

$$i = \{1, \dots, N\}, j = \{2, \dots, N\} \wedge j > i.$$

The 2D Cramer-Rao Lower Bound (CRLB) for TOA (which is similar to that of TDOA if all the combinations of time difference of arrival are evaluated), which represents the minimum achievable localization variance (and subsequently RMSE) for non-biased estimators, is given in [82, 83]. The CRLB is dependent on the time delay estimation accuracy and the location of the receivers. Figure 2.14 shows examples of the CRLB for multiple case scenarios, this gives the reader an idea about the effects of different configurations, and make it easier to compare this localization method to the one shown in Section 2.3.1. As it can be seen a much lower CRLB is obtained in Fig. 2.14 in comparison to the CRLB in Fig. 2.10, nevertheless, the presented conventional TOA approach cannot be implemented for coherent distributed antenna arrays as the secondary nodes are dynamic and it will never be possible to have an exact knowledge of their location. A modified TOA approach that can align the phase of coherent distributed arrays is presented in Chapter 7.

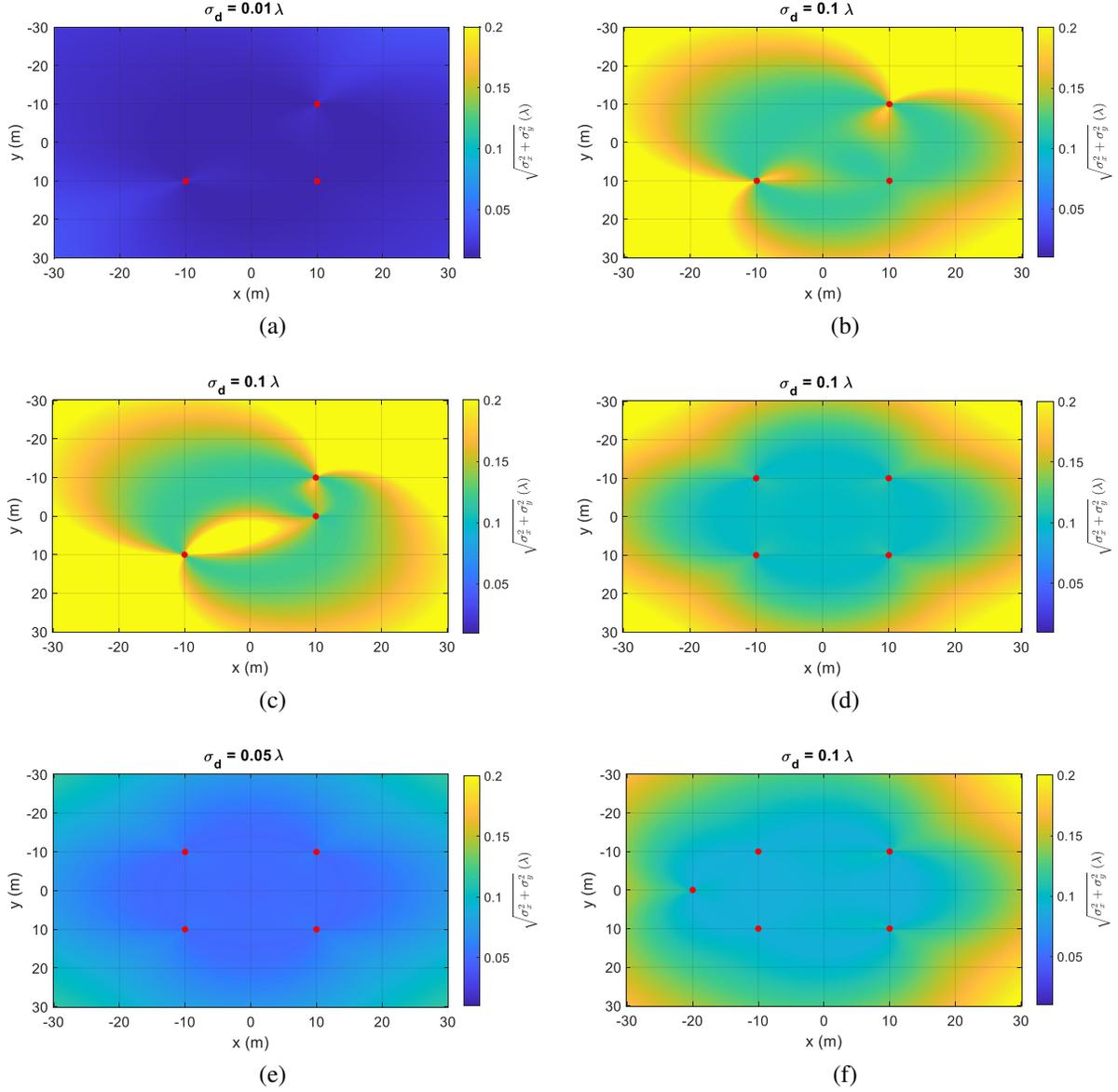


Figure 2.14: CRLB of the Transmitter localization in terms of Radial error $\sqrt{\sigma_{x_n}^2 + \sigma_{y_n}^2}$ as a function of λ for TOA systems with (a) 3 nodes and $\sigma_d = 0.01\lambda$, (b) 3 nodes and $\sigma_d = 0.1\lambda$, (c) 3 nodes and $\sigma_d = 0.1\lambda$, (d) 4 nodes and $\sigma_d = 0.1\lambda$, (e) 4 nodes and $\sigma_d = 0.05\lambda$, (f) 5 nodes and $\sigma_d = 0.1\lambda$. The secondary nodes (anchors) locations are marked with red. The RMSE of range (time delay) estimates is considered equal for all the anchors.

CHAPTER 3

FREQUENCY SYNCHRONIZATION REQUIREMENTS FOR OSCILLATORS AND PHASE-LOCKED LOOPS

Distributed beamforming between wireless systems requires significant coordination, a principal factor affecting the coherence in a distributed array is the relative stability of the local oscillators on every node. A coherent state is reached once all the nodes are sharing the same reference frequency, which typically needs to be shared wirelessly, where the secondary nodes lock their frequencies to a primary source using signal processing or adjunct and internal circuits such as PLLs, among others. Large arrays or crowded spectrum may prevent continuous frequency synchronization. Thus, frequent frequency updates might be required during which the nodes are frequency locked and between which the frequency of the oscillators drifts. The stability of the oscillator, the PLL parameters (in case PLLs are used for frequency synchronization), and the used carrier, among others, all affect the synchronization requirements. The effect of PLL phase noise on the locked local oscillator is illustrated in Fig. 3.1. I analyze in this chapter the effect of varying the update time for various oscillator characteristics and synchronization models. A framework for analyzing the distributed beamforming performance from oscillator and PLL parameters is presented.

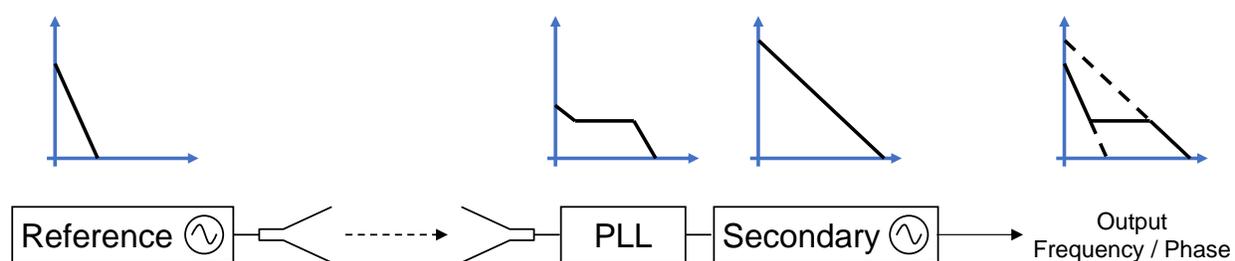


Figure 3.1: Phase noise profile of the wirelessly synchronized secondary nodes where the reference oscillator has lower phase noise in comparison to the phase noise on the locked oscillator. From [6] © 2021 IEEE.

3.1 Phase Noise of and Operation of Oscillators and Phase-Locked Loops

3.1.1 Oscillator Phase Noise

One of the important metrics for an oscillator is its spectral purity. Ideally, an oscillator can be described using a sine wave with nominal amplitude v_0 and nominal angular frequency w_0 :

$$v(t) = v_0 \sin(w_0 t). \quad (3.1)$$

In practice, the amplitude and phase fluctuations of the output signal pushes the spectrum of an oscillator to frequencies around w_0 . These imperfections also lead to extra spurs and harmonics at the frequencies $2w_0$, $3w_0$, $4w_0$, etc. The instantaneous output of the oscillator can be represented by

$$v(t) = v_0 [1 + A(t)] \sin [w_0 t + \alpha(t)] + v_{\text{harmonics}}(t) + v_{\text{spurs}}(t), \quad (3.2)$$

where $A(t)$ represents the amplitude fluctuations, and $\alpha(t)$ refers to the phase fluctuations. Phase noise includes phase fluctuations, spurious signals ($v_{\text{spurs}}(t)$), and harmonics ($v_{\text{harmonics}}(t)$). Generally, spurious and harmonics signals are caused by the use of non-linear active devices such as MOSFETs and BJTs. Spurious signals are also caused by external noise sources such as noise on power supply and bias currents. The phase noise of an oscillator is quantified and represented by the single-sideband (SSB) phase noise, which is defined by the ratio of power at an offset f_m with 1 Hz bandwidth over the total power for a SSB. The phase noise is obtained from [90]

$$L(f_m) = 10 \log \left(\frac{P_{\text{noise}}(f_m)}{P_{\text{sig}}} \right) = 10 \log \left(\frac{v_{n,RMS}^2(f_m)}{v_{c,RMS}^2} \right), \quad (3.3)$$

where P_{sig} is the signal power, $v_{c,RMS}^2$ is the root mean square (RMS) amplitude of the carrier, and $v_{n,RMS}^2(f_m)$ is the RMS amplitude of the signal representing the phase noise at an offset frequency f_m .

The phase noise in (3.3) is a result of both amplitude modulated and phase modulated noise, thus the oscillator phase noise can be expressed as

$$L(f_m) = 10 \log \left(\frac{S_\phi(f_m)}{2} + \frac{S_a(f_m)}{2} \right), \quad (3.4)$$

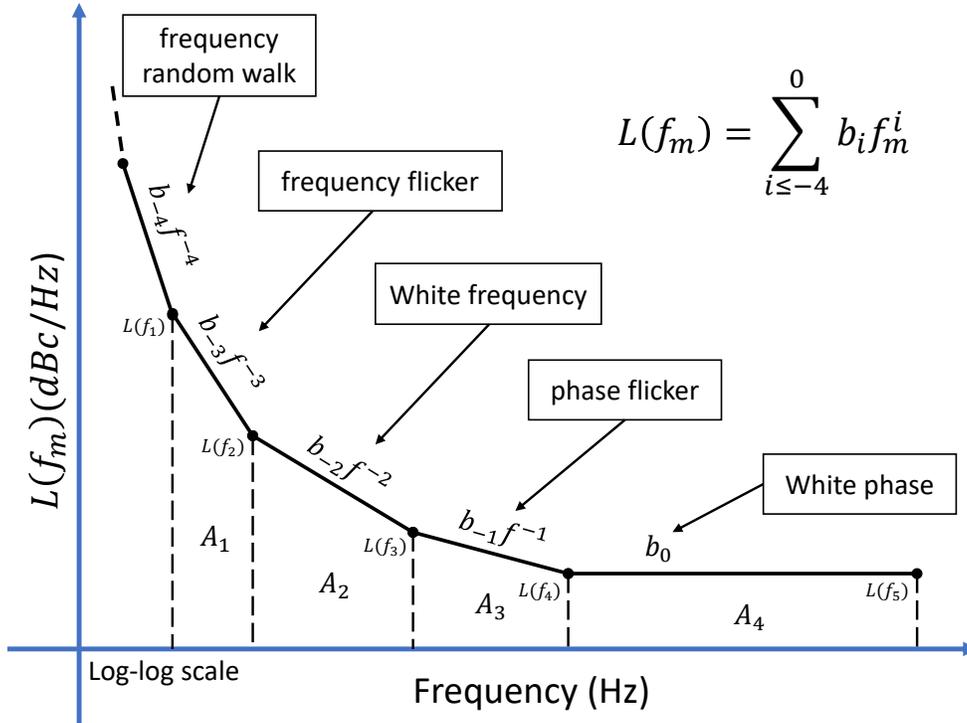


Figure 3.2: General shape of the phase noise of an oscillator. The noise power associated with each phase noise type is represented by b_i . From [6] © 2021 IEEE.

where $S_\phi(f_m)$ represents the double-sideband (DSB) phase noise spectral density and $S_a(f_m)$ represents the DSB amplitude noise spectral density. The contribution of phase modulated noise around the carrier dominates the contribution of amplitude modulated noise, making it the main contributor to the phase noise power. However, at far offsets, both phase modulated and amplitude modulated noises contribute equally.

The spectral profile of the phase noise of an oscillator can be modeled using linear line segments in dBc/Hz each with different slope, where each segment represents different type of noise as shown in Fig. 3.2. The physical sources of noise in each region derive from the following [91]:

- Random walk frequency modulated (FM) noise ($1/f^4$) is usually within 10 Hz of the carrier. This type of noise is related to the physical environment, such as temperature and vibration, and it causes the frequency of the oscillator to drift slowly over time.
- Flicker FM noise ($1/f^3$) is mainly related to the resonance of the oscillator. This noise is

usually observed for high quality oscillators where the $1/f^2$ and $1/f$ noise components are low, and thus do not mask it.

- White FM noise ($1/f^2$) is common for oscillators with passive resonators.
- Flicker phase modulated (PM) noise ($1/f$) is observed generally in active components such as amplifiers. Flicker PM noise is always observed in oscillators, and the $1/f$ corner frequency is typically between 500 KHz and 1 MHz for CMOS technology, and between 1 KHz and 10 KHz for bipolar transistors. Beyond these frequencies, the noise profile is dominated by the white PM noise [92].
- White PM (f^0) noise is present for all the active component and can be kept to low values with appropriate circuit design.

The phase noise of oscillators has been characterized in the literature using multiple models [93–95], with the Leeson model [93] being one of the earliest and well known models. Leeson phase model is based on a linear time-invariant (LTI) approach, and is given by

$$L(f_m) = 10 \log \left[\frac{FkT}{2P_{sig}} \left(1 + \left(\frac{f_0}{2Q_L f_m} \right)^2 \right) \left(1 + \frac{f_k}{f_m} \right) \right], \quad (3.5)$$

where F is the effective noise figure of the oscillator, k is the Boltzmann constant 1.38×10^{-23} J/K, T is the temperature in K, f_0 is the nominal frequency of the oscillator, Q_L is the loaded quality factor of the resonator, and f_k is the flicker noise corner frequency in the phase noise, which is not always equal to the flicker noise corner frequency of the active devices. Thus, minimizing the phase noise of an oscillator can be done by tackling the three parameters F , P_{sig} , and Q_L , where F needs to be minimized while P_{sig} and Q_L need to be maximized.

3.1.2 General Architecture of PLLs

A PLL is a control system that locks the phase and subsequently the frequency of a voltage-controlled oscillator (VCO) to the phase of a reference input signal. The output signals of a PLL are based on the phase difference between the output of the VCO and the reference signal [90,96–98].

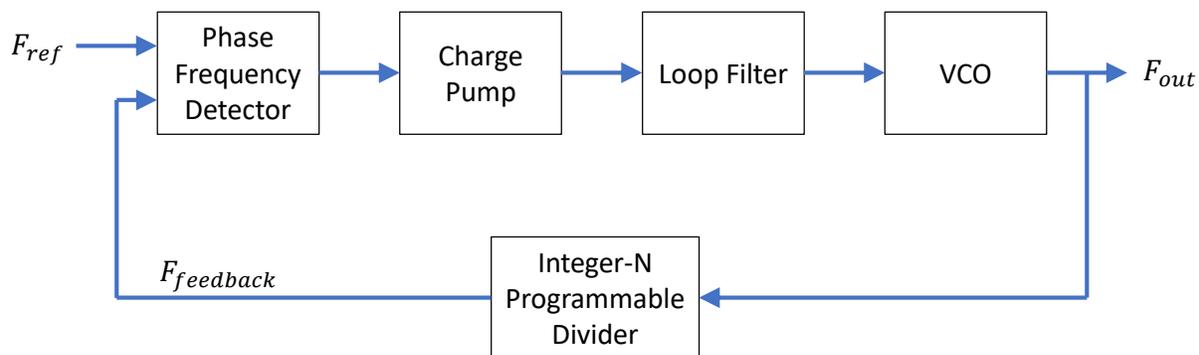


Figure 3.3: Typical PLL architecture with frequency synthesizer. F_{ref} represents the input frequency to the PLL, this frequency can be lower version than the input reference frequency, which can be generated using frequency dividers. F_{out} is the output frequency of the VCO, while the $F_{feedback}$ is a divided version of the F_{out} . From [6] © 2021 IEEE.

Fig. 3.3 shows a simplified block diagram of a PLL with frequency synthesizer (or integer-N programmable divider). A frequency synthesizer is an electronic circuit that generates multiple frequencies out of a single reference frequency, and it is mainly used to ensure that the feedback frequency is equal to the input reference frequency. The main blocks in Fig. 3.3 are

- Integer-N programmable divider: using a controller, the integer-N programmable divider functions as a frequency divider. The frequency divider is used to ensure that the frequency of the feedback signal is equal to the frequency of the input reference. In many cases the input reference signal is also fed through another programmable divider (integer-M programmable divider for example) before being fed to the PLL in order to match the output frequency of the integer-N programmable divider [90].
- Phase frequency detector (PFD): Initially phase detectors (PDs) were used to compare the phase of the incoming reference signal with the phase of the VCO. A PD could be represented by a mixer for analog signals and by an XOR circuit for digital signals, among other configurations. A PD puts out a signal that is proportional to the phase difference between two input signals of the same frequency. The PD produces a series of output pulses whose width is proportional to the phase difference, which are afterwards smoothed by the loop

filter. The problem of PDs is that they only lock to phase and not frequency. PFD is an enhanced version of PD that locks the VCO phase and frequency to that of the incoming signal, which leads to higher sensitivity and faster frequency locking. PFDs can be built using D flip-flops or NAND gates [99].

- Charge pump (CP): the CP acts like a DC to DC converter that takes as an input the output of the PFD and gives either positive or negative voltage with high efficiency allowing the VCO to quickly lock onto the phase of the incoming signal.
- Loop filter: a low-pass filter (LPF) is used in a PLL to minimize the noise. The LPF is used to filter out all the in-band noise contributed by the PFD and CP leaving only the much lower VCO noise outside the loop bandwidth. When the VCO is locked to the reference input, the desirable configuration of a PLL is called a clean-up PLL where the LPF is designed to be narrow, typically below 1 kHz. On the other hand, when the PLL is not locked to a reference signal, the LPF can have a wider bandwidth to allow the PLL to lock to signals with larger frequency offsets than the cut-off frequency of the clean-up PLL.
- VCO: the VCO inside of a PLL tends to have a higher phase noise compared to the reference signal, at least if the reference signal is fed through a cable. Once locked, the frequency and phase of the VCO are locked to that of the primary node or reference system. Since the phase and frequency of local oscillators drift over time, regardless of how stable they are, the locked VCO and the reference oscillator both drift in frequency and phase over time, but without having any relative phase and frequency drift. Nevertheless, in the case of wireless frequency synchronization, there will always be a varying relative phase offset between the reference oscillator and the VCO, as will be shown in Chapter 4.

An important feature to monitor when frequency synchronization is performed using pulsed signals is the locking time. It is necessary to know the required pulse width that enables the PLL to lock. It is possible to estimate the locking time by first obtaining the transfer function of the PLL, which can be represented as a simple negative feedback system as shown in Fig. 3.4. This system

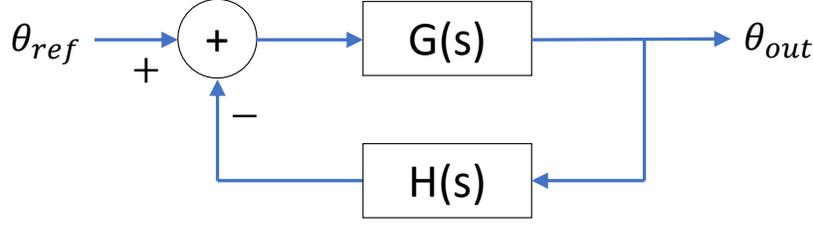


Figure 3.4: PLL represented using a negative feedback model.

gives a transfer function of the form [90].

$$\frac{\theta_{out}(s)}{\theta_{ref}(s)} = T(s) = \frac{G(s)}{1 + G(s)H(s)}, \quad (3.6)$$

where θ_{out} is the output phase of the PLL loop, θ_{ref} is the phase of the reference signal, $H(s) = 1/N$, and $G(s)$ is the forward transfer function. $G(s)$ of the presented PLL architecture is expressed as

$$G(s) = \frac{I_{CP}}{2\pi} \cdot F(s) \cdot \frac{2\pi K_{VCO}}{s}, \quad (3.7)$$

where $\frac{I_{CP}}{2\pi}$ is the gain of the PFD and CP, I_{CP} is the CP current, $\frac{2\pi K_{VCO}}{s}$ is the gain of the VCO, K_{VCO} is the sensitivity of the VCO, and $F(s)$ is the transfer function of the low-pass filter. This study is based on the model in [90], where the filter was chosen as shown in Fig. 3.5 and it had the following transfer function

$$F(s) = \frac{V_{cnt}}{I_{CP}} = \frac{sR_zC_z + 1}{s(sR_zC_z + C_z + C_p)}, \quad (3.8)$$

where the resistance R_z , and the capacitance C_z and C_p represent the filter passive components that are shown in Fig. 3.5.

The PLL transfer function can be expressed as a second order transfer function as follows [97]

$$T(s) = \frac{2\zeta w_n s + w_n^2}{s^2 + 2\zeta w_n s + w_n^2}, \quad (3.9)$$

where

$$\zeta = \frac{R_z C_z}{2} w_n, \quad (3.10)$$

and

$$w_n = \frac{I_{CP} K_{VCO}}{N(C_p + C_z)}, \quad (3.11)$$

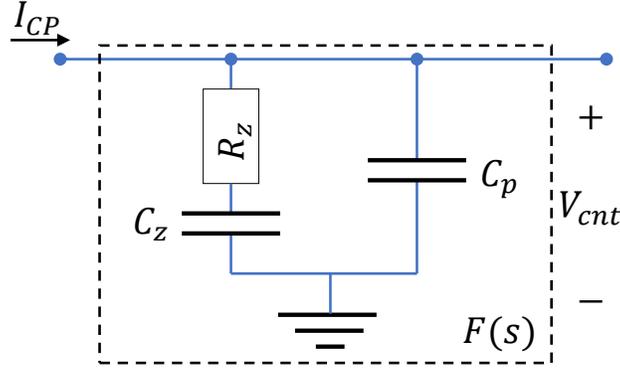


Figure 3.5: A three element LPF with resistive and capacitive components.

where N represents the frequency divider value. By taking the inverse Laplace of (3.9), the settling or lock time can be approximated as

$$t_{lock} = -\frac{\ln(\text{tolerance fraction} \times \sqrt{1 - \zeta^2})}{\zeta \omega_n}. \quad (3.12)$$

For PLL applications, *tolerance fraction* represents $f_{accuracy}/f_{jump}$, where $f_{accuracy}$ represents the acceptable error margin, and f_{jump} represents the frequency deviation from f_0 [100]. In this case, the locking time can be expressed as

$$t_{lock} = -\frac{\ln\left(\frac{f_{accuracy}}{f_{jump}} \times \sqrt{1 - \zeta^2}\right)}{\zeta \omega_n}. \quad (3.13)$$

While PLLs have a finite locking time to achieve a phase-locked state, especially when they are locked to pulsed signals, this time is on the order of microseconds [90, 101], which is significantly short than a typical frequency synchronization pulse and can be ensured easily. It is important to note that the phase transfer function that is derived in (3.6) is only valid when the VCO is locked to the frequency of the reference signal.

3.1.3 PLL Phase Noise

Phase noise of a VCO placed inside a PLL is shaped in accordance to the PLL specifications, the reference signal, and its phase noise characteristics. In a free running configuration, when the PLL is not locking to a reference frequency, the VCO phase noise is the same as its original phase noise

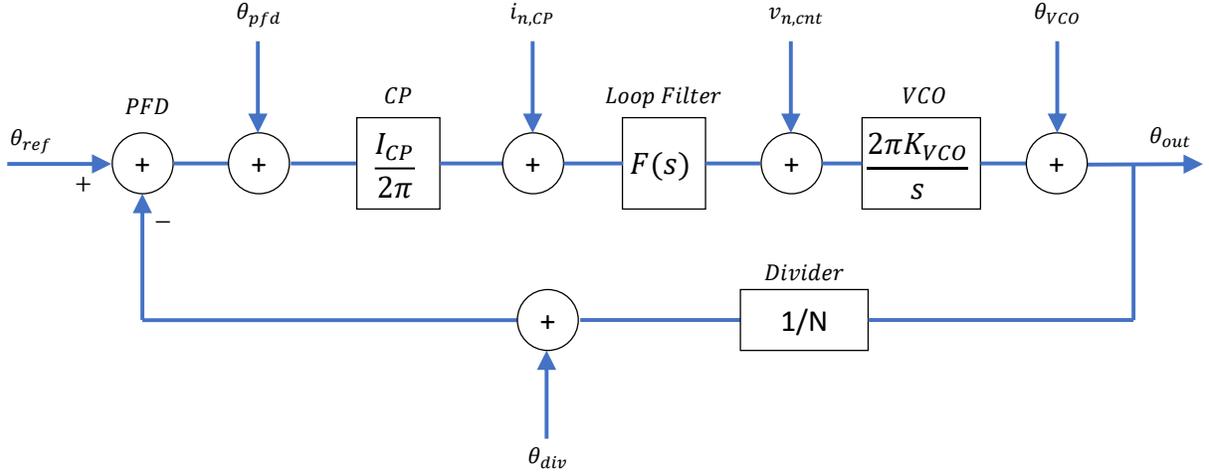


Figure 3.6: PLL noise model.

as specified in the datasheet. On the other hand, when the VCO is operating inside a locked PLL, the phase noise of the VCO is referred to as PLL phase noise and is characterized by the transfer function of the PLL, which is obtained by evaluating the noise contribution from every block in the PLL.

The phase noise in PLLs has been characterized using multiple models, in this section the model that was derived in [90] is of interest. The model is shown in Fig. 3.6. The term θ_{ref} is represented in $\text{rad}/\sqrt{\text{Hz}}$ and it includes the phase noise of the reference input, which combines the noise of the reference crystal oscillator, crystal buffer, and reference frequency divider. The noise caused by the integer-N programmable divider is represented by θ_{div} and is expressed in $\text{rad}/\sqrt{\text{Hz}}$. θ_{VCO} is the phase noise of the VCO represented in $\text{rad}/\sqrt{\text{Hz}}$. The term θ_{PFD} represents the PFD noise in $\text{rad}/\sqrt{\text{Hz}}$. The noise at the VCO input due to the loop filter and other coupled noise sources to the control line is represented by $v_{n,cnt}$ in $V/\sqrt{\text{Hz}}$. the term $i_{n,CP}$ represents the noise of the CP current expressed in $A/\sqrt{\text{Hz}}$. The total noise is calculated by adding the contribution from all the blocks in an RMS sum

$$\theta_{out}^2 = \left(\frac{G(s)}{1 + G(s)H(s)} \right)^2 \cdot \left[\theta_{ref}^2 + \theta_{div}^2 + \theta_{pfd}^2 + \left(i_{n,CP} \cdot \frac{2\pi}{I_{CP}} \right)^2 \right] + \left(\frac{1}{1 + G(s)H(s)} \right)^2 \cdot \left[\theta_{VCO}^2 + \left(v_{n,cnt} \cdot \frac{2\pi K_{VCO}}{s} \right)^2 \right]. \quad (3.14)$$

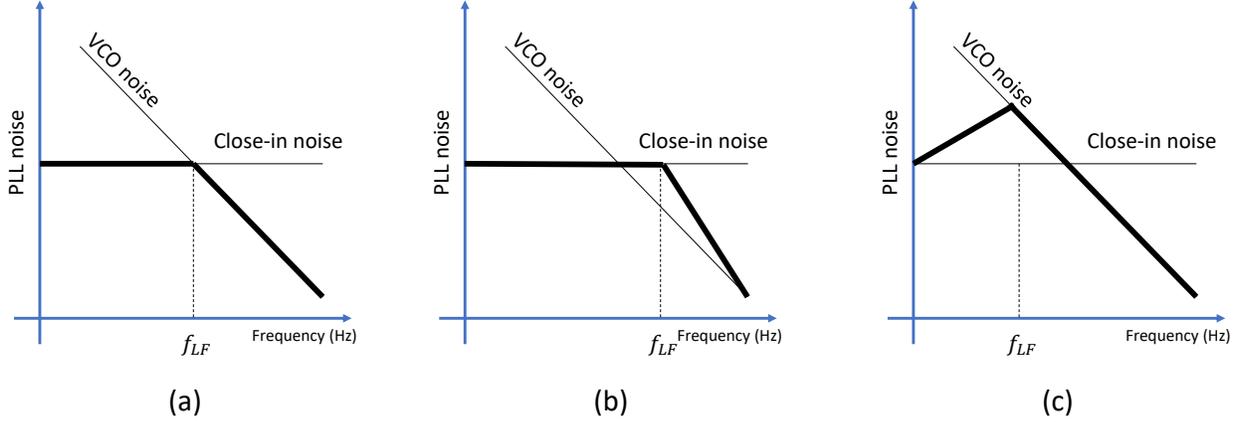


Figure 3.7: PLL phase noise shape based on the selection of the loop bandwidth, where f_{LF} represents the frequency of the loop filter of the PLL. The bandwidth can be: (a) optimum, (b) large, (c) narrow.

The complete derivation of the PLL phase noise can be found in [90]. As can be seen, the factor $\frac{G(s)}{1+G(s)H(s)}$ acts as a low pass filter for the phase noise contribution from the reference, PFD, divider, and CP. On the other hand, the factor $\frac{1}{1+G(s)H(s)}$ acts as a high pass filter for the VCO phase noise, and the phase noise generated by the control line. Thus within the loop bandwidth (up to the frequency f_{LF}), the phase noise of the VCO and control line is suppressed. Knowing that, the bandwidth of the PLL must be chosen at the point where the VCO phase noise intersects the close-in phase noise as shown in Fig. 3.7a, otherwise the behavior in Figs. 3.7b and 3.7c is observed. The close-in phase noise in dBc is represented as

$$L_0 = L_{PLL,nf} + 20 \log(N) + 10 \log(f_{ref}), \quad (3.15)$$

where $L_{PLL,nf}$ represents the PLL noise floor due to its circuitry. A typical PLL output phase noise is represented in Fig. 3.8; the first region represents the reference phase noise, the second is dominated by L_0 , and the third region is dominated by the phase noise of the VCO.

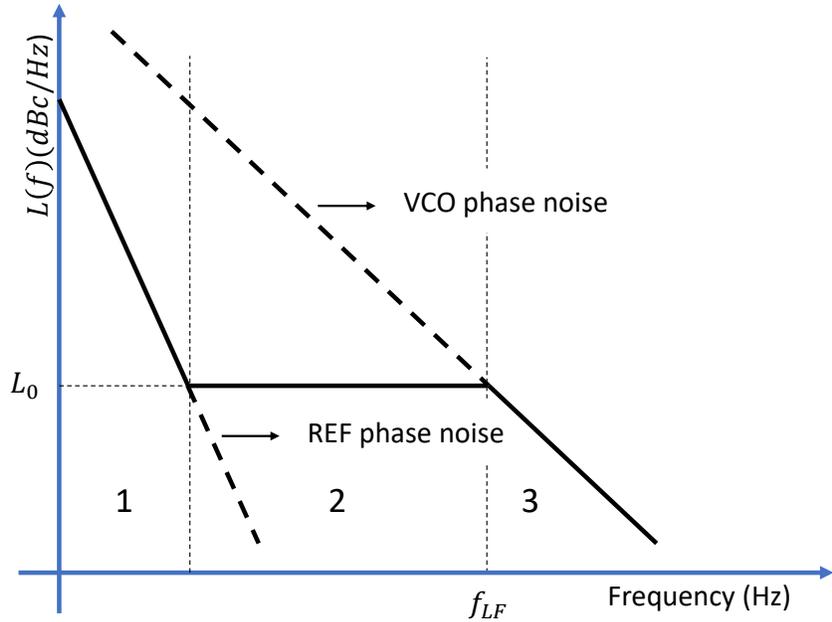


Figure 3.8: Typical PLL phase noise profile. In the first region, the phase noise of the reference signal dominates the PLL phase noise; on the other hand, the close-in noise dominates region 2 leading to an increase in the PLL noise floor; finally, region 3 is dominated mainly by the phase noise of the locked VCO. f_{LF} , which represents the frequency of the loop filter of the PLL, needs to be selected appropriately as shown in Fig. 3.7 to minimize the phase noise of the PLL. From [6] © 2021 IEEE.



Figure 3.9: Example of a clock jitter for the real case compared to the ideal time domain performance of a clock where no jitter is observed.

3.2 Phase Noise Interpretation of Locked and Unlocked Oscillators in Frequency and Time

3.2.1 Time Jitter

The phase noise is a representation of the spectral impurity of an oscillator. In the time domain, this noise is represented as time jitter in seconds or phase jitter radians as shown in Fig. 3.9. The

RMS phase jitter in radians and the RMS jitter in seconds can be extracted from the phase noise profile of a VCO or PLL. At frequencies lower than 10 Hz (around the nominal frequency), the phase noise is dominated by the frequency drift and frequency random walk, while at frequencies higher than 10 Hz, the phase noise is shaped by the jitter. Since the clock of a system is obtained from the VCO, they both experience similar drift and jitter effects. The phase noise of a VCO can be represented as shown in Fig. 3.2, while the phase noise of a PLL is shaped similarly to Fig. 3.8. Regardless of the phase noise profile, the integrated phase noise power A is obtained by evaluating the following equation

$$A = 10 \log (A_1 + \dots + A_n), \quad (3.16)$$

where the total area A from Figs. 3.8 or 3.2 is broken into the separate areas A_1 through A_n . As an example, the area A_i can be calculated using the following equation

$$A_i = 10^{\{[L(f_i)+L(f_{i+1})]/2+10\log(f_{i+1}-f_i)\}/10}. \quad (3.17)$$

The phase noise can be integrated until any desired frequency depending on the application and the hardware; the value $f_0/2$ was chosen in this dissertation, which is the half value of the nominal frequency of a given oscillator. Integrating the phase noise to higher values has negligible effects with low quality oscillators since they have high noise powers far before the threshold $f_0/2$, however, for very stable oscillators, this extra integration bandwidth can have a significant effect on the calculated jitter, and in most of the cases it might not represent the actual observed jitter on the oscillator. The RMS phase jitter is obtained from

$$\text{Phase Jitter} = \sqrt{2 \times 10^{A/10}}. \quad (3.18)$$

Similarly, the RMS time jitter is obtained from

$$\text{Time Jitter} = \frac{\text{Phase Jitter}}{2\pi f_0}. \quad (3.19)$$

3.2.2 Allan Deviation and Frequency Drift

The Allan deviation (ADEV) or Allan variance (AVAR), which is the square of ADEV, are used to evaluate the stability of the systems affected by noise processes, and are represented by $\sigma_y(\tau)$ and $\sigma_y^2(\tau)$. Two-sample or M-sample AVAR and ADEV are commonly used for sensor and oscillator characterization to evaluate how much they vary over a given time. M-sample AVAR and ADEV are represented by $\sigma_y^2(M, T, \tau)$ and $\sigma_y(M, T, \tau)$, where T represents the time between measurements and τ the observation time for a single measurement. Evaluation of oscillators mainly relies on the two-sample AVAR $\sigma_y^2(\tau) = \langle \sigma_y^2(2, \tau, \tau) \rangle$, where $T = \tau$ and $S = 2$ [102]. Nevertheless, the M-sample AVAR is obtained from

$$\sigma_y^2(M, T, \tau) = \frac{1}{M-1} \left\{ \sum_{i=1}^{M-1} \bar{y}_i^2 - \frac{1}{M} \left[\sum_{i=1}^{M-1} \bar{y}_i \right]^2 \right\}, \quad (3.20)$$

where \bar{y} is the average fractional frequency obtained from

$$\bar{y}(t, \tau) = \frac{1}{\tau} \int_0^\tau y(t + t_v) dt_v, \quad (3.21)$$

here the average is taken over observation time τ , and $y(t)$ is the fractional-frequency error which can be obtained from

$$y(t) = \frac{f(t) - f_0}{f_0}, \quad (3.22)$$

where $f(t)$ is the oscillator frequency and f_0 is the nominal frequency. Note that $\bar{y}(t, \tau)$ may also be obtained from

$$\bar{y}(t, \tau) = \frac{x(t + \tau) - x(t)}{\tau}, \quad (3.23)$$

where $x(t)$ is the clock value. As for our case we are interested in the two-sample AVAR, $\sigma_y^2(\tau)$ is derived from (3.20) and is expressed as

$$\sigma_y^2(\tau) = \langle \sigma_y^2(2, \tau, \tau) \rangle = \frac{1}{2} \langle (\bar{y}_{n+1} - \bar{y}_n)^2 \rangle. \quad (3.24)$$

AVAR or ADEV can be deduced through transformation from a typical phase noise profile such as the one shown in Fig. 3.2, as explained in [103, 104]. The line segments representing $\sigma_y^2(\tau)$ are shown in Fig. 3.10 and are analyzed in [105].

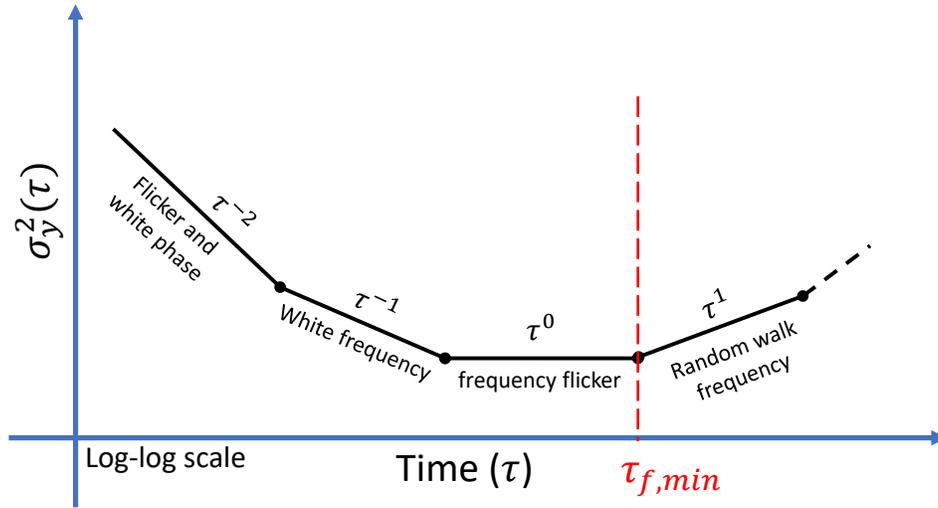


Figure 3.10: AVAR profile for an oscillator. $\tau_{f,min}$ represents the instance where the AVAR is mainly dominated by the random walk FM noise. Later on, when analyzing the RMS frequency drift, it is important to select a value $\tau_f \geq \tau_{f,min}$. From [6] © 2021 IEEE.

The main interest in $\sigma_y^2(\tau)$ or $\sigma_y(\tau)$ is due to their ability to characterize the random frequency walk and frequency drift (for the remainder of this document, both the random frequency walk and frequency drift are going to be referred to as frequency drift). As mentioned in Section 3.2.1, the frequency drift is evaluated using the frequencies within 10 Hz from the nominal frequency in the phase noise profile. Nevertheless, characterizing the first 10 Hz of the phase noise is almost always impossible, especially for the frequencies within few Hz from the nominal frequency (if possible, extremely expensive hardware is needed to measure the phase noise at very low frequencies). AVAR offers an alternative way to study the frequency drift such as follows: an ADEV of 10^{-11} for $\tau = 1$ s represents an instability in frequency between two observations 1 s apart with a relative RMS value of 10^{-10} . Thus, 0.001 Hz RMS drift is observed after 1 s for a 10 MHz oscillator. This metric allows us to tune the synchronization time to allow a 0.0001 Hz RMS drift with 1 Hz update rate or lower RMS drift values with higher update rates. Even though this frequency drift value seems small, when generating a 5 GHz carrier from a similar VCO, an RMS drift of $\frac{5}{10} \frac{\text{G}}{\text{M}} \times 0.001 \text{ Hz} = 0.5 \text{ Hz}$ is observed after 1 s, which could lead to a dramatic phase shift over time between the nodes in the array. Since it is possible to design the update time based on the

tolerable frequency drift among the nodes in the distributed array, the RMS frequency drift values can be interpolated and obtained for the update times of interest using

$$\text{RMS Frequency Drift} = \sigma_y(\tau_f) f_0 \sqrt{\frac{T}{\tau_f}}, \quad (3.25)$$

where T is the update time, and τ_f is chosen once the random frequency walk asymptote τ becomes visible in the AVAR, ideally at $\tau_{f,min}$ in Fig. 3.10. At $\tau_{f,min}$, the variance caused by the phase noise at frequencies higher than 10 Hz softens, and the AVAR becomes dominated by the random frequency walk and frequency drift, thus any value at or beyond $\tau_{f,min}$ is usable. AVAR or ADEV can be evaluated for a free-running VCO or locked VCO using pulsed synchronization signals, since between the synchronization pulses the frequency of the VCO drifts in comparison to the reference source. On the other hand, when a VCO is frequency locked using CW, there is no relative frequency drift between the VCO and the reference, for the case of coherent distributed arrays, this translates to no relative frequency drift between the secondary nodes and the primary node. For the case of frequency locked VCO, the frequency drift reaches its maximum closer to the beginning of the next update interval.

3.2.3 Time Drift

Since the time of a clock is extracted from the cycles of the VCO, The RMS time drift is linked to the RMS frequency drift. However, the time deviation (TDEV), which represents the standard deviation of the timing errors in a clock/oscillator, is used instead of the ADEV measurements that are only used to assess the spectral purity of a VCO. Similar to the processing of AVAR, the TDEV value of interest resides at an observation time τ_t equal or higher than $\tau_{t,min}$, at which the effect of the time drift dominates the effect of the time jitter. $\tau_{t,min}$ represents the time at which the slope of TDEV starts to have a positive slope. The RMS time drift at any time τ or any update time T is interpolated using

$$\text{RMS Time Drift} = \text{TDEV}(\tau_t) \sqrt{\frac{T}{\tau_t}}. \quad (3.26)$$

Similar to the frequency drift, time drift is only present if the frequency synchronization is not achieved using CW signals. In contrast to the RMS frequency drift, the RMS time drift has lower impact on the coherence of the distributed arrays, since the time drift only affects the modulation at a certain carrier f_c and the tolerated timing errors are inversely proportional to the bandwidth of the modulated signal, compared to the bandwidth of the signal for the case of frequency errors.

3.3 Distributed Beamforming Analysis

This section sets a framework to evaluate the effects of phase and time jitter, frequency drift, and time drift on the coherent gain G_c from (2.1). The summation of the received signals from N nodes in the far field $s_r(t) = \sum_0^{N-1} s_{r,n}(t)$ is modified to include the effects of instantaneous phase, frequency, and time errors that are observed in a real case scenario where frequency locking using PLLs is implemented with pulsed synchronization signals. Also, both $s_r(t)$ and the ideal summation of the received signals from N nodes in the far field $s_i(t) = \sum_0^{N-1} s_{i,n}(t)$ are modified to include any desired signal modulation when evaluating the effect of timing errors. Performance degradation due to errors in the calibration of propagation channel, frequency, time, and phase can be studied separately and evaluated as compounded errors such as in Chapter 2, where the localization errors were investigated separately.

Two types of transmissions are considered to evaluate the coherent gain in terms of the noise parameters in Section 3.2: first case is represented by the transmission of a CW (from the coherent distributed array) where the effects of phase jitter, phase shift due to frequency shift, and frequency shift are observed. This scenario is evaluated with both CW synchronization signals and pulsed synchronization signals. The second case is represented by the transmission of phase modulated signals where the effects of time jitter and time drift are added to the received signals that are observed in the first case. The ideal and non-ideal received signals from N nodes in the far field are denoted with $s_{i,k}(t)$ and $s_{r,k}(t)$, where k is set to 1 for the case of CW transmission by the array and it is set to 2 for the case of modulated-signal transmission. For the first case we have

$$s_{i,1,n}(t) = e^{j\left(\frac{f_{c,n}}{f_0}(2\pi t f_0)\right)}. \quad (3.27)$$

The non-ideal received signals are represented by

$$s_{r,1,n}(t) = e^{j\left(\frac{f_{c,n}}{f_0}(2\pi t[f_0 + \delta f_n] + [\phi_{0,n} + \delta\phi_n])\right)}, \quad (3.28)$$

where $f_{c,n}$ represents the carrier frequency at a node n , the frequency drift δf_n for the n^{th} oscillator is taken from a normal distribution with a standard deviation equal to the RMS value from (3.25). Similarly, $\delta\phi$ represents the phase jitter, taken from a normal distribution with a standard deviation equal to the RMS value in (3.18). After every update time, ϕ_0 is added to compensate for the extra phase shift obtained from considering frequency drift equal throughout the transmission. The frequency drift at a time t leads to a frequency offset δf_n that varies over time to reach a certain final value before the next frequency/phase update as shown in (3.25). It is not possible to track the exact frequency shift over time, since for every synchronization cycle, the frequency drifts differently. Nevertheless, the average change of frequency of a certain interval can be approximated with a linear change with a slope t/T , this averaging of the frequency change over an interval T represents the average case. The phase at a time T with a linear frequency/phase shift is given by

$$\phi_{actual} = 2\pi \int_0^T f_0 + \frac{t}{T}\delta f_n dt. \quad (3.29)$$

The total phase error at time T is then

$$\phi_0 = \phi_{actual} - 2\pi \int_0^T f_0 + \delta f_n dt. \quad (3.30)$$

As for the second scenario, the transmitted signal is represented by a BPSK signal [106] where a modulation/phase mismatch results in nulling the beamformed signals. The bandwidth of the signal was selected depending on the carrier frequency, and in general a high bandwidth was assigned to the modulation in order to account for the worst case scenario while evaluating the coherent gain. the ideal signal is given by

$$s_{i,2}(t, n) = \cos\left(\frac{f_c(m)}{f_0}(2\pi t f_0) + \pi(1 - b(t))\right), \quad (3.31)$$

where $b(t) = 0$ or 1 , representing a phase shift of 180° or 0° . The non-ideal received signals $s_{r,2,n}$ are represented by

$$s_{r,2}(t, n) = \cos\left(\frac{f_c(m)}{f_0}(2\pi t (f_0 + \delta f_n) + \phi_{0,n} + \delta\phi_n) + \pi(1 - b(t + \delta t_n))\right), \quad (3.32)$$

where $b(t + \delta t_n)$ is the desired bit for the time $t + \delta t_n$ (the bit cannot fluctuate between 0 and 1 if the time jitters back and forth, once it is assigned to the next bit, it cannot be reassigned to the previous one). δt_n represents both the time drift and jitter and is taken from a normal distribution with a standard deviation equal to the RMS value in (3.19). An offset equal to the random time drift is added to the normal distribution; this offset was obtained from a normal distribution with standard deviation equal to RMS time drift obtained from (3.26).

3.4 Practical Examples

In this section I use the above framework to evaluate the impact on coherent beamforming gain in a distributed phased array in the presence of oscillator phase noise in multiple examples. I consider an array of multiple scattered nodes and use a threshold performance metric of $G_c \geq 0.9$, indicating that the distributed beamforming operation achieves 90% of the ideal beamforming gain (a degradation of less than 0.5 dB). Array sizes $N = 2, 10, 20$, and 100 are considered. Performance bounds on the mainbeam gain at frequencies extending from the microwave to the millimeter-wave bands, using two nominal 100 MHz oscillator phase noise profiles are considered. First, the coherent gain is analyzed in the presence of the reference signal being transmitted continuously without interruptions. Next, the coherent gain is evaluated for the case where the reference signal is transmitted using pulsed waveforms with a varying update time T . The reference signal is assumed to be broadcast wirelessly from an arbitrary location within the array. Each node receives the reference signal and inputs it to its PLL. As noted above, the locking time for a typical PLL is on the order of microseconds; it is reasonable to assume that the pulsed reference signal is of a sufficiently longer duration to support locking. The two oscillator phase noise profiles that are shown in Table 3.1 are considered; VCO A represents a typical VCO, while VCO B represents a low phase noise VCO. The phase noise data, ADEV and TDEV used in this work are similar to what is available in the market or in the literature such as in [107–110].

© 2021 IEEE. Section 3.4 is reprinted with minor modifications, with permission, from Section V of "S. R. Mghabghab, and J. A. Nanzer "Impact of VCO and PLL Phase Noise on Distributed Beamforming Arrays with Periodic Synchronization," IEEE Access, vol. 9, pp. 56578-56588, 2021".

Table 3.1: SSB phase noise of the selected 100 MHz oscillators.

| | VCO A | VCO B |
|------------------|-------------|-------------|
| $f_1 = 10$ Hz | -60 dBc/Hz | -110 dBc/Hz |
| $f_2 = 100$ Hz | -90 dBc/Hz | -140 dBc/Hz |
| $f_3 = 1$ kHz | -120 dBc/Hz | -165 dBc/Hz |
| $f_4 = 10$ kHz | -145 dBc/Hz | -185 dBc/Hz |
| $f_0/2 = 50$ MHz | -155 dBc/Hz | -190 dBc/Hz |

The phase noise of the VCO inside of a PLL has two possible phase noise behaviors depending on the locking status of the PLL. When the PLL is not receiving the frequency locking signals, the VCO is in a free-run mode, and the PLL output phase noise will correspond solely to that of the VCO. On the other hand, when the PLL is frequency locked, its phase noise will have a shape similar to that in Fig. 3.8. In this section the coherent gain is evaluated in a coherent distributed array for three architectures: 1) the reference signal generated using VCO A (the high-phase noise oscillator) and the secondary nodes are equipped with the same VCO A; 2) the reference signal generated using VCO B (the low-phase noise oscillator) and the secondary nodes are equipped with the same VCO B; and 3) the reference signal generated using VCO B from an external transmitter and the secondary nodes are equipped with VCO A. The case where the reference signal is generated using VCO A from an external transmitter and the secondary nodes are equipped with VCO B was not considered since in this case the phase noise of the reference VCO will be much higher than the phase noise of the VCOs used by the secondary nodes, which leads to an overall system performance similar to the first architecture.

Two assumptions are considered in this section; first, it is assumed that the phase noise profile of the input reference signal matches that of the reference oscillator. In practice, the phase noise is usually higher for the input reference signal since a large path loss might be present between the primary and secondary nodes which would lead to a decrease in the received signal SNR on the secondary node. This decrease in signal power then may lead to an increase in white phase noise. In addition to that, interference might be present which would increase the phase noise. In

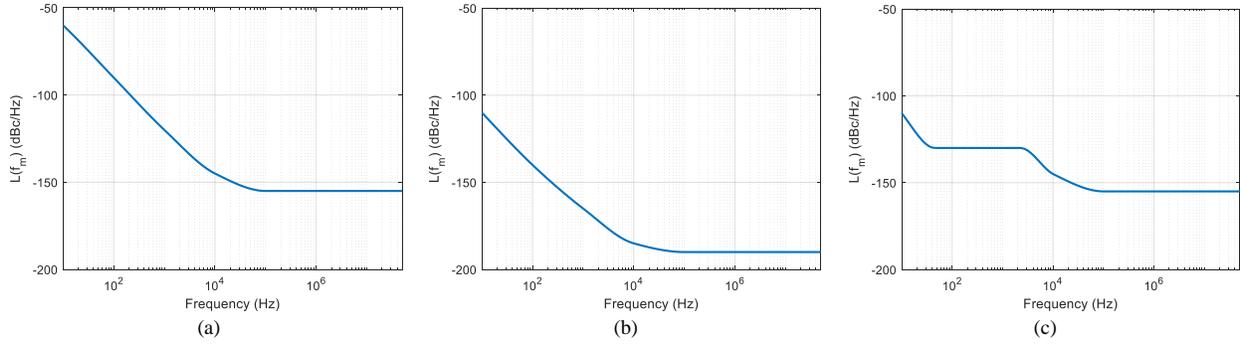


Figure 3.11: Phase noise of the locked PLLs for architectures: (a) 1, (b) 2, and (c) 3. In architecture 1, VCO A was being locked by the PLLs of the secondary nodes and it was used to generate the reference signals. In architecture 2, VCO A was replaced by VCO B. In architecture 3, VCO B was used to generate the reference signals, while VCO A was being locked by the PLLs of the secondary nodes. In all the architectures, the loop filter was assumed to be designed appropriately to minimize the phase noise of the PLLs.

order to deal with this assumption in practice, it is possible to estimate the phase noise of the input reference signal for the worst-case scenario to estimate a bound on beamforming performance. Second, I consider the case where no frequency multiplication or division is needed (which is the case when the frequency of the input signal is equal to the nominal frequency of the oscillator). In this case the phase noise of the input reference signal is equal to the phase noise of the input signal to the PLL. If the reference frequency requires multiplication/division by a factor M , the phase noise of input reference increases/decreases by a factor of $20 \log(M)$.

In the case where the same VCO is used as the reference and PLL VCO, the phase noise of the continuously locked PLL will resemble the phase noise of the VCO, since region 1 in Fig. 3.8 will overlap region 3. This means that the phase noise characteristics of the VCOs can be considered as the phase noise characteristics at the output of the PLL for architectures 1 and 2. For the third architecture, the phase noise at the output of the locked PLL will resemble that of Fig. 3.8. The close-in phase noise of the PLL was taken to be -130 dB/Hz until the frequency of the loop filter (matching typical PLL values), with a rapid decrease in magnitude beyond that point. In all the architectures, the loop filter was assumed to be appropriately selected to match the profile in Fig. 3.8. Based on the values in Table 3.1 and the approximated close-in phase noise of the PLL, the phase noise of the PLL for the three selected architectures is shown in Fig. 3.11.

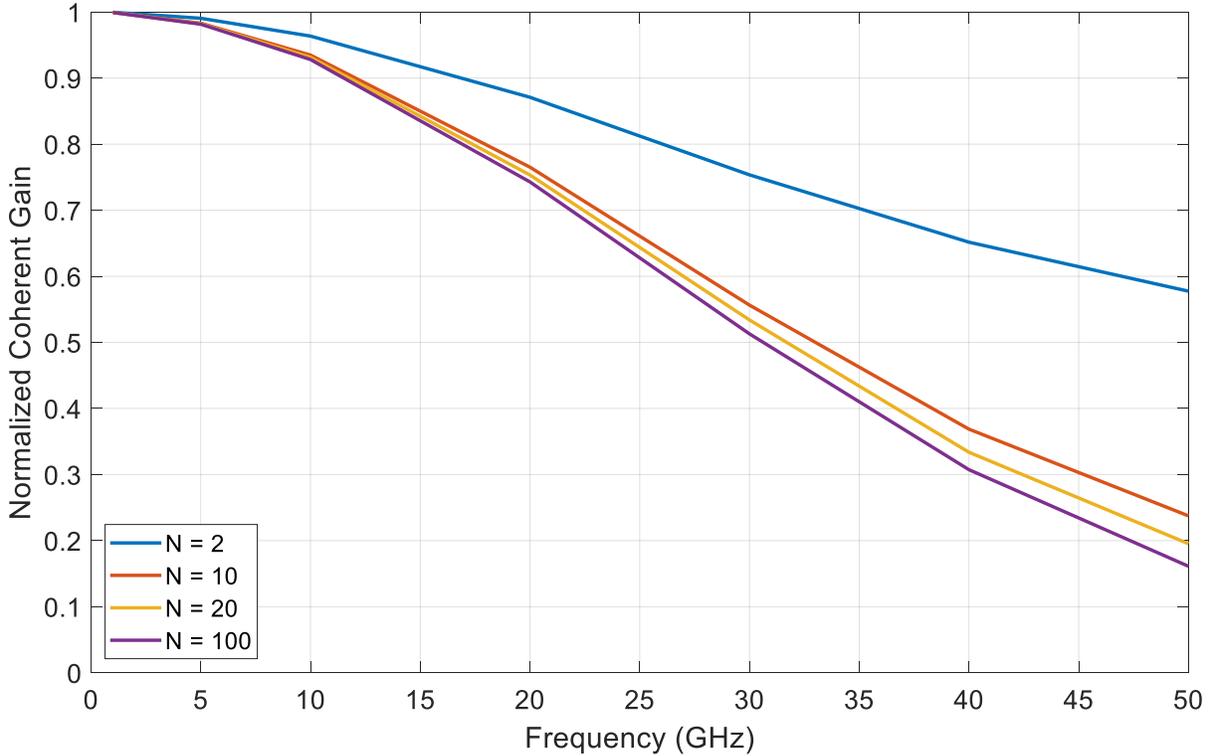


Figure 3.12: Coherent gain for various carrier frequencies for arrays of sizes $N = 2, 10, 20,$ and 100 with continuous locking for architecture 1 (using the high phase noise oscillator as the reference and PLL VCO) with the phase noise profile from Fig. 3.11a.

For VCO A, the RMS of the phase jitter is equal to 2.7×10^{-3} rad and the RMS of the time jitter is equal to 4.37×10^{-12} s, calculated from (3.18) and (3.19), respectively. For VCO B, the RMS of the phase jitter is equal to 9.95×10^{-6} rad and the RMS of the time jitter is equal to 1.58×10^{-14} s. For the third architecture, when the nodes are continuously locked, the RMS of the phase jitter is equal to 1.85×10^{-4} rad and the RMS of the time jitter is equal to 2.95×10^{-13} s; in the intervals where the VCOs are not frequency locked, their RMS phase jitter and RMS time jitter are equal to that of their internal VCOs (VCO A in this case). The ADEV for VCO A at $\tau_f = 1$ s is $\sigma_y(1) = 10^{-10}$, while for VCO B at $\tau_f = 1$ s it is $\sigma_y(1) = 2.3 \times 10^{-12}$. The selected TDEV at $\tau_t = 1$ s for VCO A is $\text{TDEV}(1) = 8 \times 10^{-11}$ s, while for VCO B at $\tau_t = 1$ s it is $\text{TDEV}(1) = 2 \times 10^{-12}$ s. τ_f and τ_t values were inspired by datasheets of available VCOs in the market such as in [107]; a common practice is to show in the datasheets either the entire ADEV data of the VCO or to report τ_f at 1 s since this value is usually close to $\tau_{f,min}$; similar practice is

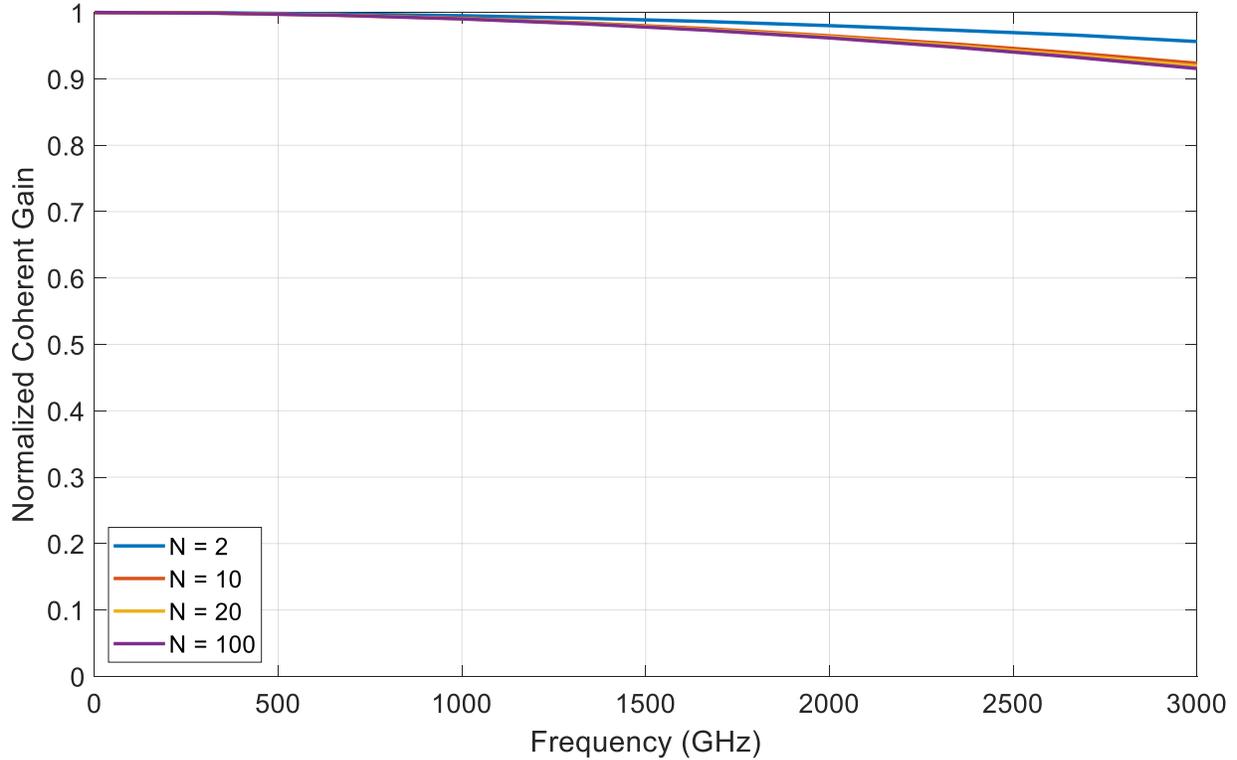


Figure 3.13: Coherent gain for various carrier frequencies for arrays of sizes $N = 2, 10, 20,$ and 100 with continuous locking for architecture 2 (using the low phase noise oscillator as the reference and PLL VCO) with the phase noise profile from Fig. 3.11b.

done in regards to TDEV. Whenever the entire data for ADEV and TDEV is present, it is preferable to select the values of τ_f and τ_t at $\tau_{f,min}$ and $\tau_{t,min}$ since these values tend to capture the worst case scenario.

3.4.1 Beamforming Performance with Continuous Synchronization

When the PLLs of the secondary nodes receive the reference signal continuously without interruption, there are no relative frequency or time drifts, and furthermore the effects of timing errors are negligible (many orders of magnitude less than frequency errors). Thus, the signals can be modeled as in (3.28) with $\delta f_n = 0$. The principal performance parameter is thus the carrier frequencies f_c that are supported by the chosen VCOs for distributed beamforming. Generally, with continuous locking, even oscillators with moderate to poor phase noise can support good distributed beamforming performance up to millimeter-wave frequencies. The coherent gain was plotted for

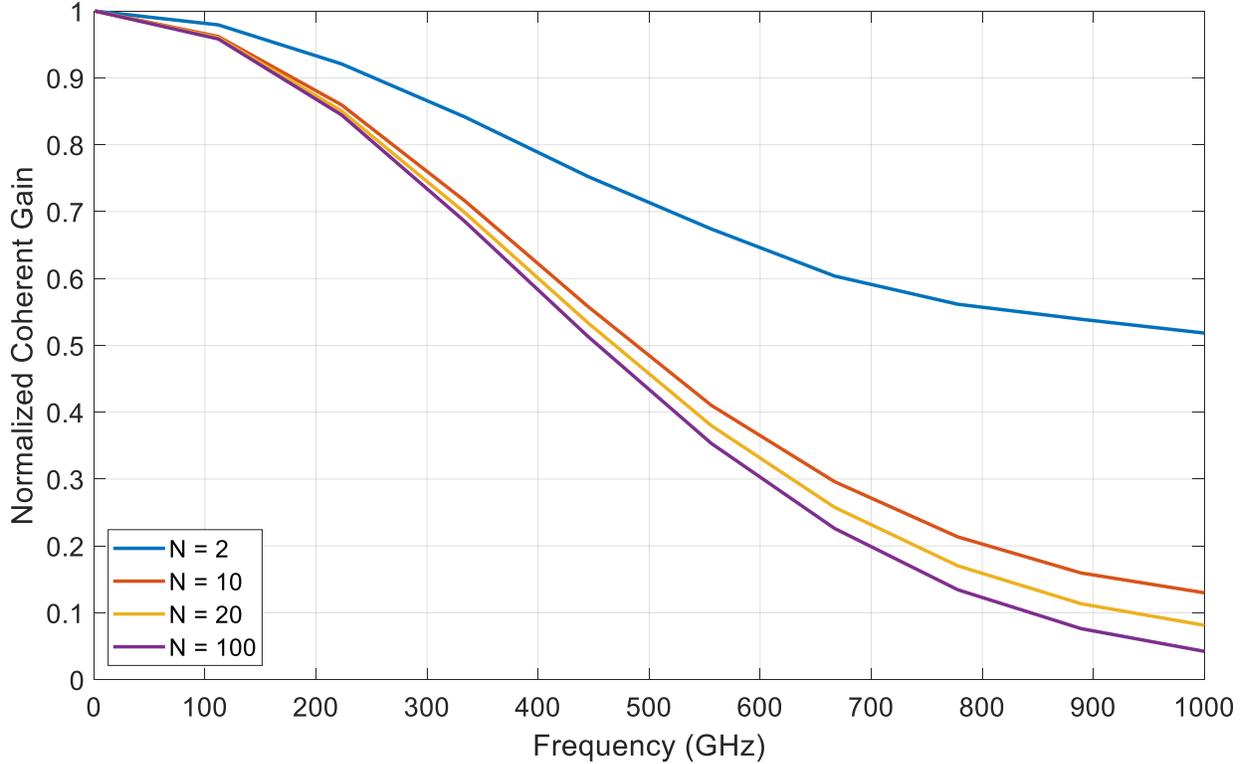


Figure 3.14: Coherent gain for various carrier frequencies for arrays of sizes $N = 2, 10, 20,$ and 100 with continuous locking for architecture 3 (using the low phase noise oscillator as the reference and the high phase noise oscillator as the PLL VCO) with the phase noise profile from Fig. 3.11c.

the three architectures, multiple array sizes, and multiple f_c values; one thousand Monte Carlo simulations were run for carrier frequencies up to 50 GHz for architecture 1, as for the second architecture, frequencies up to 3 THz were considered, and for the third architecture, frequencies up to 1 THz were evaluated. The results for architecture 1 are shown in Fig. 3.12. As it can be seen, when two nodes are used, at least a coherent gain of 0.9 can be achieved for frequencies below 17 GHz. As for the other array sizes, at least 0.9 of coherent gain is achievable for frequencies below 12 GHz. The results for architecture 2 are shown in Fig. 3.13. For all the array sizes that were used, much higher than 0.9 of coherent gain can be achieved for frequencies higher than 3 THz. This frequency is very high for coherent distributed arrays, and in practice it is not feasible since there will be many other sources leading to phase errors. Nevertheless, this shows that the phase noise in the selected oscillator will not be a limiting factor for coherent transmission even at very high carrier frequencies. Regarding the third architecture, the results are shown in Fig. 3.14.

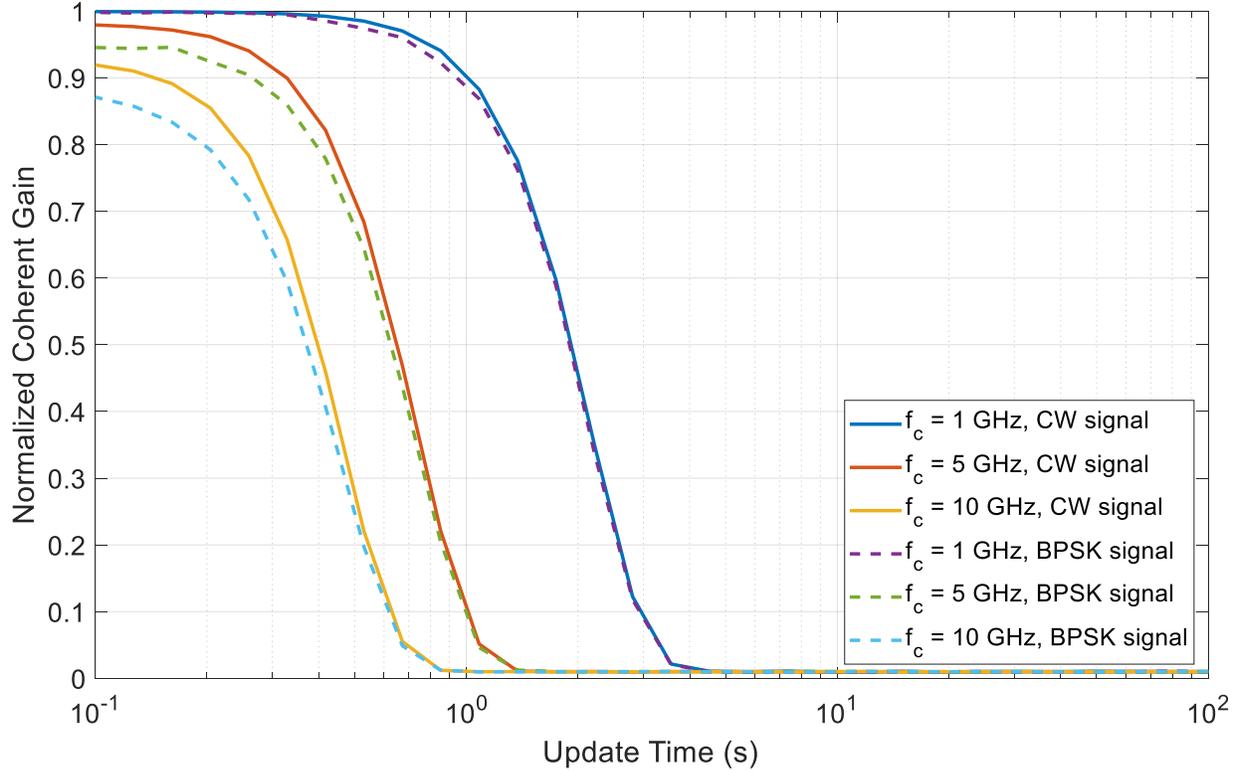


Figure 3.15: Coherent gain for CW and BPSK signals with periodic synchronization, where the jitter and frequency drifts (CW and BPSK), and timing errors (BPSK) were considered for nodes equipped with VCO A. Array size $N = 100$ was selected.

When two nodes are used, at least 0.9 of coherent gain can be achieved for frequencies below 252 GHz. As for the other array sizes, at least 0.9 of coherent gain is achievable for frequencies below 160 GHz. Clearly, continuous synchronization provides good distributed beamforming performance; if a low phase noise oscillator is used as the reference, good performance well beyond 100 GHz is attainable.

3.4.2 Beamforming Performance with Periodic Synchronization with Phase and Frequency Errors

Practically, continuous synchronization is challenging to obtain in distributed beamforming. Here we analyze the impacts of periodic synchronization on beamforming performance. Due to the inherent drift of the oscillators, periodic synchronization is necessary, even with low phase noise oscillators. This section analyzes the phase and frequency errors without considering the time

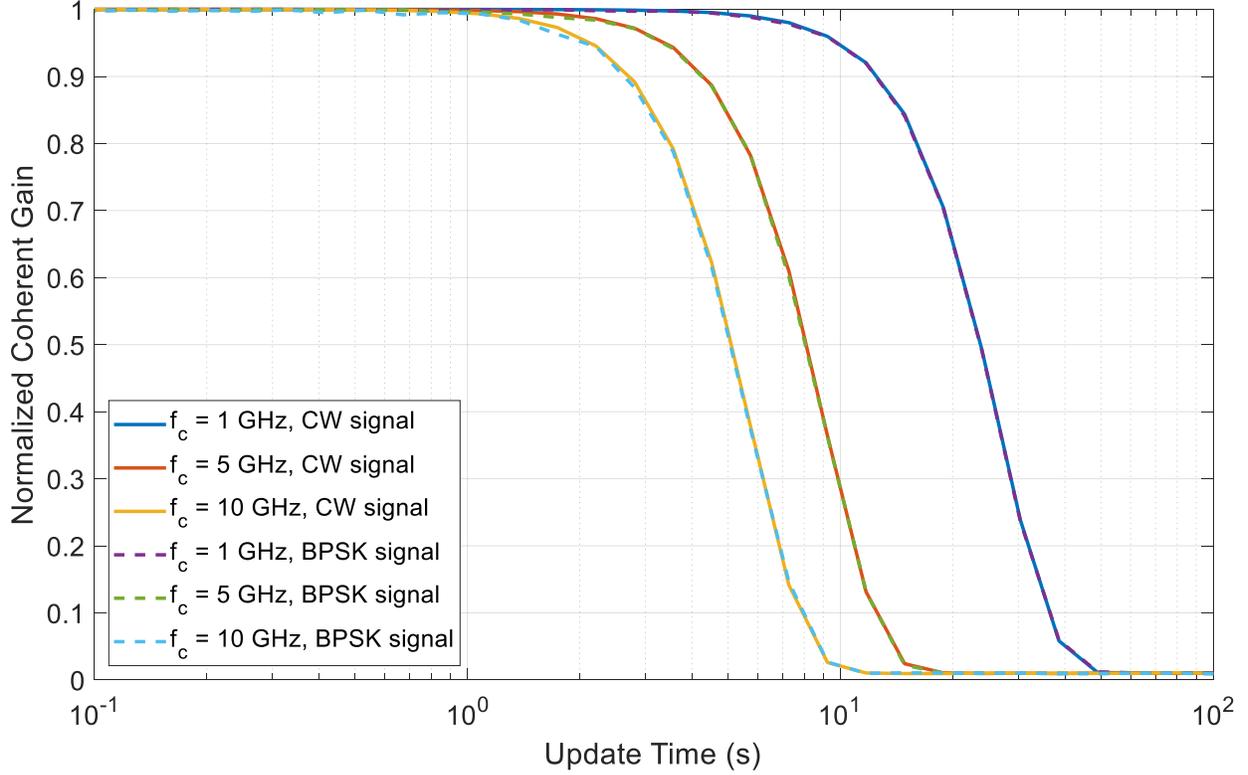


Figure 3.16: Coherent gain for CW and BPSK signals with periodic synchronization, where the jitter and frequency drifts (CW and BPSK), and timing errors (BPSK) were considered for nodes equipped with VCO B. Array size $N = 100$ was selected.

jitter and time drift, an approach supporting CW beamforming without modulation; the transmitted signals are thus modeled as in (3.28). The coherent gain was analyzed for update times T between 100 ms and 100 s. 1000 Monte Carlo simulations were generated for the carrier frequencies of 1, 5, and 10 GHz, and the coherent gain was calculated for arrays with 100 elements (larger arrays generally yield more stringent requirements, as seen in Section 3.4.1). The coherent gain was calculated for the last 100 wavelengths before the next update time to evaluate the worst case scenario, since the last few wavelengths represent the signal with the maximum drift before the next update. The phase noise profile for architectures 1 and 3 was obtained from the phase noise of VCO A, since the VCOs are free running, and the phase noise in architecture 2 was obtained from the phase noise of VCO B. The results are shown in Fig. 3.15 for architectures 1 and 3, and in Fig. 3.16 for architecture 2. When the high phase noise oscillator is used in a free-run mode or in the locked PLL, the jitter causes appreciable coherent gain degradation at higher frequencies,

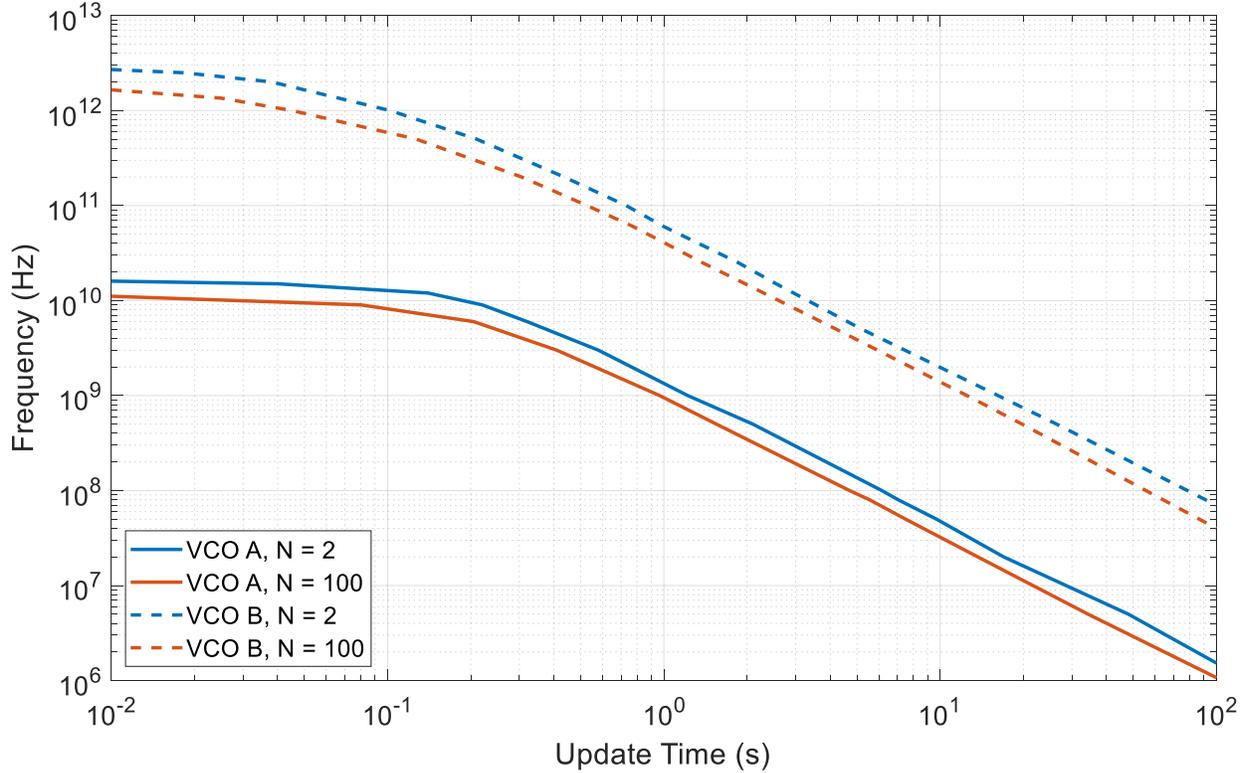


Figure 3.17: Carrier frequencies supporting a coherent gain of 0.9 for periodic synchronization with array sizes $N = 2$ and 100, where the jitter, frequency drifts and timing errors were considered for nodes equipped with either VCO A or B.

indicated by the fact that the curves never reach the ideal value of $G_c = 1$. Furthermore, update frequencies above 1 Hz are necessary to achieve reasonable coherent gain values. When the low phase noise oscillator is used as both reference and PLL VCO, the gain achieves the ideal value for relatively low update frequencies extending below 1 Hz.

3.4.3 Beamforming Performance with Periodic Synchronization with Phase, Frequency, and Timing Errors

In this section we consider modulated waveforms where timing is relevant, and perform the same analysis as in Section 3.4.2 with the addition of the timing errors, thus using (3.32) for the signal model. The BPSK signal was transmitted at bit rates relative to the transmitted carrier frequencies; for carrier frequencies of 1, 5, and 10 GHz, the bit rates were 100, 500, and 1000 Mbit/s. High bit rates were chosen to capture the effect of the timing errors in the worst case scenarios. Results for

$N = 100$ are shown in Figs. 3.15 and 3.16 for the three architectures as explained in Section 3.4.2. It can be seen that the timing errors have a small impact on the coherent gain, even for very high modulation rates. This result is expected since the timing requirements for a coherent operation are relative to the symbol rate, whereas the requirements for frequency and phase synchronization are relative to the carrier frequency. The degradation in coherent gain is more visible for VCO A since it had high TDEV in comparison to VCO B. It can be seen that in Fig. 3.15 the initial coherent gain for the 5 and 10 GHz has dropped for the BPSK signals in comparison to the CW signals. Also, the decrease in coherent gain started at a slightly reduced update time for the BPSK transmission. Timing errors had a smaller effect in Fig. 3.16 due to the better stability of the oscillator.

The maximum carrier frequencies that can be achieved with a 0.9 Coherent gain where the jitter, frequency drifts, and timing errors are considered for VCOs A (representing architectures 1 and 3) and B (representing architecture 2) with $N = 2$ and 100 in Fig. 3.17, demonstrating that the low phase noise oscillator generally supports higher operating frequencies with longer update intervals.

CHAPTER 4

WIRELESS FREQUENCY SYNCHRONIZATION FOR CENTRALIZED SYSTEMS

The importance of wireless frequency synchronization for coherent distributed array was discussed in Chapter 1. Chapter 3 investigated the performance of distributed phased arrays that rely on PLLs for wireless frequency synchronization. It was shown that wireless frequency synchronization using CW signals produces the best outcome, as it prevents the frequency of the oscillators on every node from drifting. Throughout this dissertation, wireless frequency synchronization was always achieved using CW synchronization signals, and the secondary nodes were frequency locked using the internal PLLs on the SDRs. In addition, the implemented frequency synchronization circuit can be used to lock the frequencies of multiple SDRs since every node can be formed by more than one SDR.

4.1 Frequency Synchronization Circuit

I focus in this work on centralized systems where the secondary nodes lock their electrical states to the electrical states of the primary node. In regards to wireless frequency synchronization, all the secondary nodes are equipped with an adjunct self-mixing circuit similar to the circuit in Fig. 4.1. Figure 4.1 represents the wireless frequency synchronization circuit that was used for the outdoor cooperative ranging experiment in Chapter 5; in the indoor experiments, a modified version of this circuit was used, where one of the LNAs was removed to accommodate for the increased power at the reception. The block diagram of the self-mixing circuit is shown in Fig. 4.2. The self-mixing circuit receives a two-tone CW synchronization signal with 10 MHz tone separation from the primary node, and then demodulates the tone separation to obtain a 10 MHz signal, which is the required frequency to lock the internal PLLs of the Ettus USRP X310 SDRs.

For indoor application, the first two LNAs in Fig. 4.2 were chosen as Mini-Circuits ZX60-83LN12+. In the outdoor experiment in Chapter 5, the second LNA was replaced with Mini-Circuits ZRL-1150LN+. These LNAs are used to boost the input power, and improve the SNR

of the received signals. ZX60-83LN12+ has an operational frequency from 0.5 GHz to 8 GHz, a gain ranging from 18 dB to 22.1 dB, and a noise figure (NF) between 1.6 dB to 2.3 dB. ZRL-1150LN+ has an operational frequency from 0.65 GHz to 1.4 GHz, a gain ranging from 32 dB to 34 dB, and a noise figure (NF) ranging from 0.8 dB to 1.1 dB. Afterwards, a bandpass cavity filter is used to suppress all the unwanted frequencies. Two bandpass filters were used, depending on the selected bands for wireless frequency synchronization: Fairview Microwave FMFL000 which operates from 902 MHz to 928 MHz, and Mini-Circuits ZVBP-4300+ which operates from 4.25 GHz to 4.35 GHz. The power splitter Mini-Circuits ZX10-2-852-S+, which has an operational frequency from 0.5 GHz to 8.5 GHz, was used to split the frequency synchronization signals and feed the two outputs to the mixer inputs. The RF input of the mixer is passed through a 10 dB attenuator to prevent the mixer from saturating, and to suppress the non-linearities at the output of the mixer. Either one of ZX05-73L+ or ZX05-43-S+ (both from Mini-Circuits) was used as mixer with 6.2 dB and 6.1 dB conversion loss respectively. The mixer was used to mix the input frequencies together so that the 10 MHz signal can be extracted. The lowpass filter Mini-Circuits BLP-10.7+, which has an 11 MHz cutoff frequency and 0.65 dB insertion loss at 10 MHz, was used to filter the harmonics and unwanted frequencies obtained from the output of the mixer. At the end, two Mini-Circuits ZFL-500HLN+ LNAs were used to boost the output power of the reference inputs up to 15 dBm and a minimum of 0 dBm, which is the desired power range for the REF IN input on the SDRs. The last two LNAs have an operational frequency from 10 MHz to 500 MHz, a 19 dB gain, and 3.8 dB NF.

A comparison between the frequencies of both, the self-mixing circuit and the clock output of the primary SDR was obtained using the setup in Fig. 4.3. In this test, the circuit was simplified and it did not include any attenuator or a bandpass filter, in addition, the last two LNAs were replaced with a generic 30 dB LNA. In this experiment, the carrier frequency was set to 2.5 GHz with two CW sidebands at ± 5 MHz. The results are shown in Fig. 4.4.

The additional test that is depicted in Fig. 4.5 was conducted. The aim of this experiment is to compare the 10 MHz frequency of the primary SDR to that of the internal oscillator frequencies for

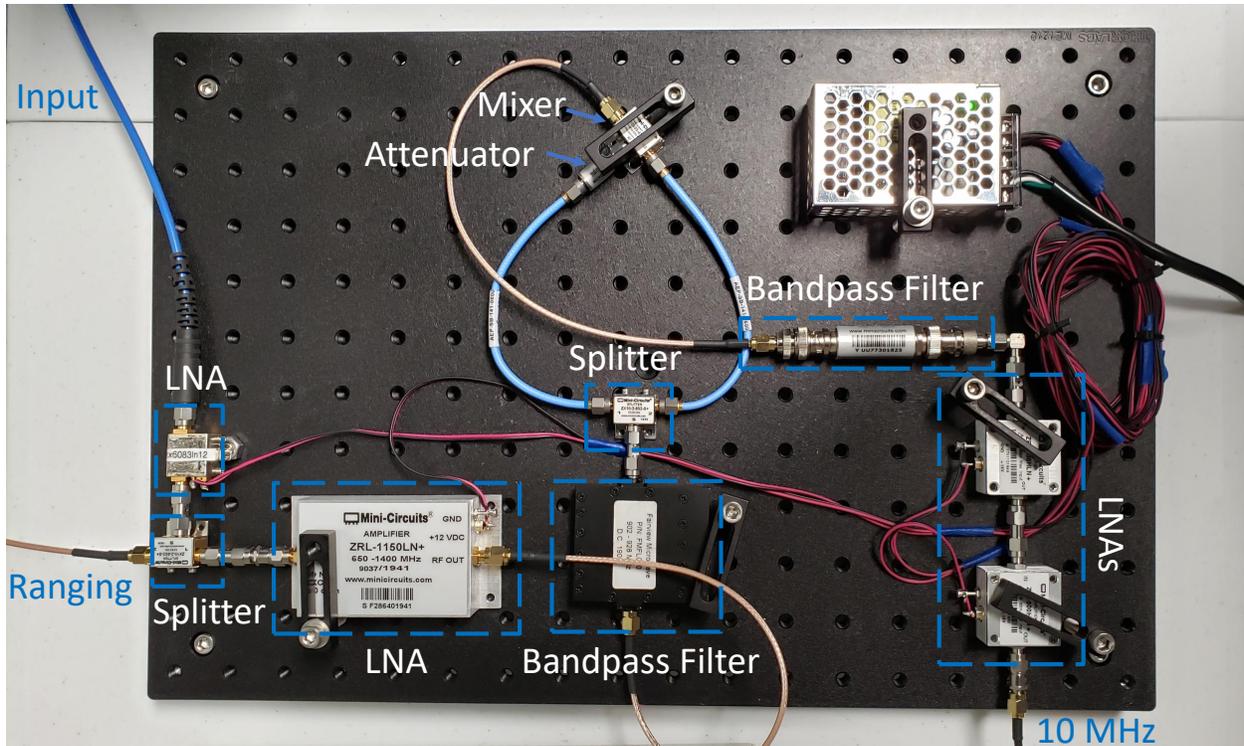


Figure 4.1: The adjunct self-mixing circuit that was used by the secondary nodes to lock their frequencies to that of the primary node. This circuit receives two-tone signals with 10 MHz tone separation and outputs a 10 MHz signal that can be fed to the input of the PLL (REF IN). A splitter was used at the input to split the received signals into ranging and frequency synchronization signals. From [7] © 2021 IEEE.

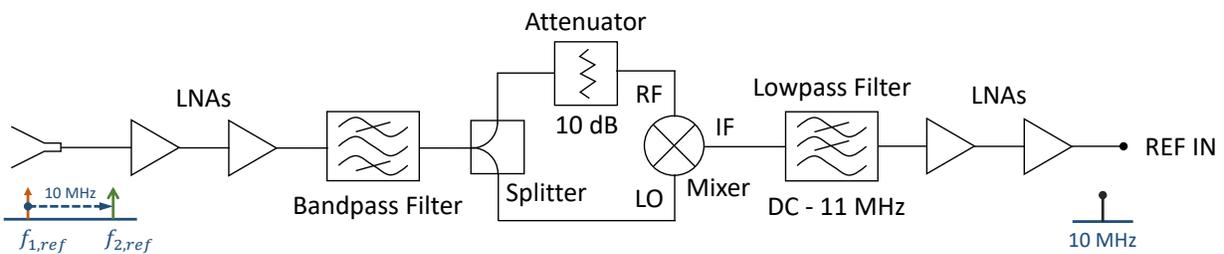


Figure 4.2: Block diagram of the adjunct self-mixing circuit for wireless frequency synchronization (adapted from [8]).

the case of both frequency locked and frequency unlocked SDRs. The reference output frequencies were captured and compared, and the results are shown in Fig. 4.6. As can be seen, the 10 MHz reference output of the primary SDR (containing the reference oscillator) matches exactly the 10 MHz signal from the frequency locked SDR. As for the frequency unlocked SDR, it can be seen that the actual frequency of the 10 MHz signal is slightly shifted. Two extra lobes are present

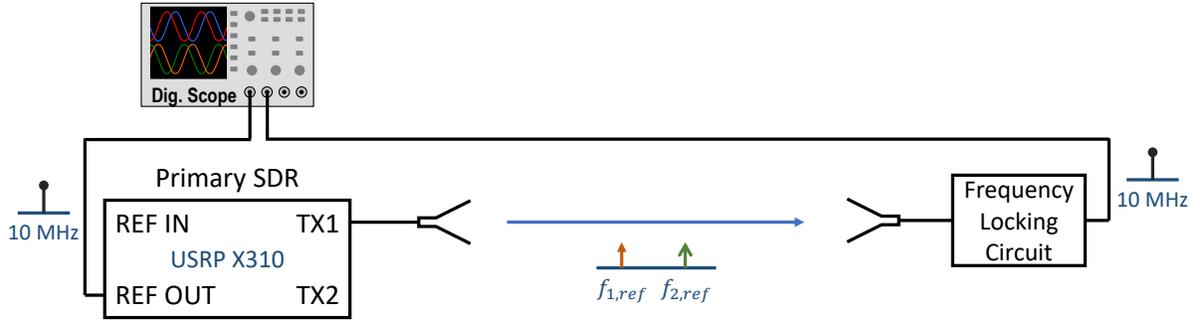


Figure 4.3: Block diagram of the experimental setup used for testing the spectrum at the output of the frequency locking circuit.

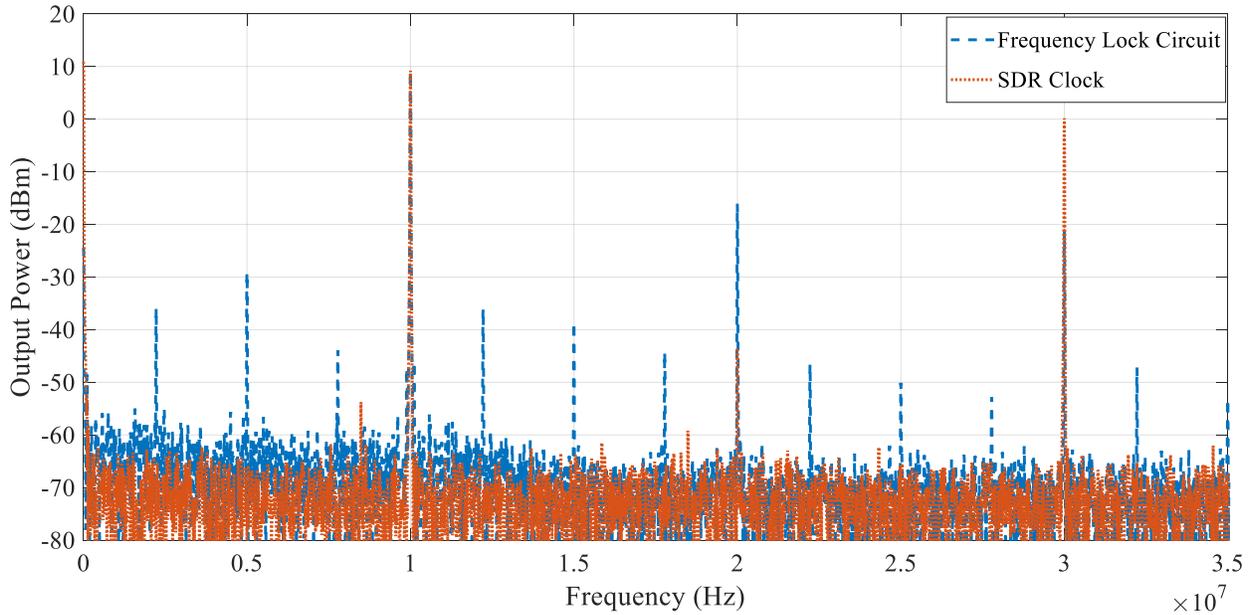


Figure 4.4: Power spectral density of both the reference clock, and the output of the proposed frequency locking circuit. From [9] © 2021 IEEE.

in the spectrum for the frequency unlocked case, this is caused by the behavior of the SDR, which was trying to lock its frequency to any available 10 MHz reference input.

4.2 Output of the Frequency Synchronization Circuit

As shown in [8], the transmitted two-tone CW frequency synchronization signals from the primary node can be represented by

$$A [\sin (2\pi f_{1,ref} t) + \sin (2\pi f_{2,ref} t)], \quad (4.1)$$

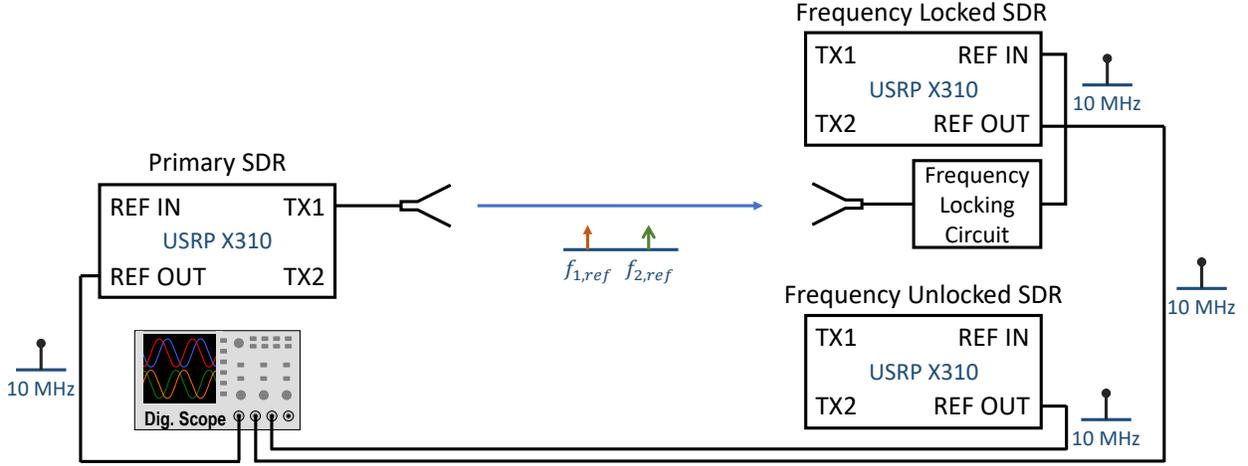


Figure 4.5: Block diagram of the experimental setup used for testing the CW 10 MHz output signals from both frequency locked and frequency unlocked SDRs.

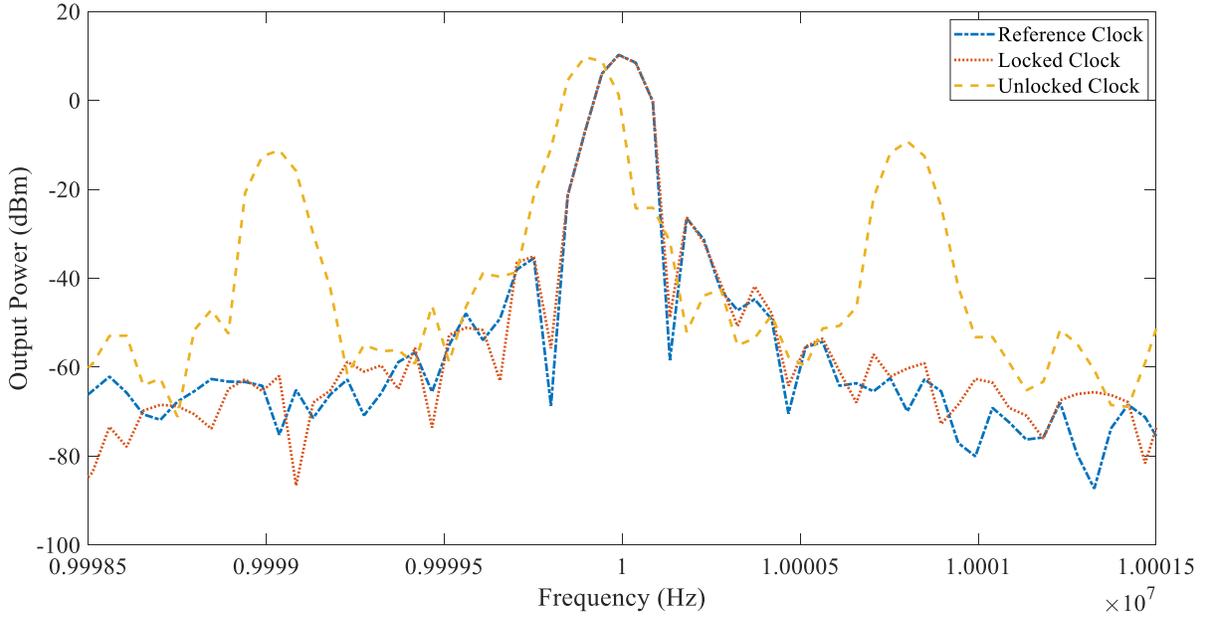


Figure 4.6: Power spectral density of the reference clock, the locked clock, and the unlocked one. From [9] © 2021 IEEE.

where A is the signal amplitude and the frequencies $f_{1,ref}$ and $f_{2,ref}$ are given by

$$f_{1,ref} = f_{c,ref} - \frac{f_{ref}}{2}, \quad (4.2)$$

$$f_{2,ref} = f_{c,ref} + \frac{f_{ref}}{2}, \quad (4.3)$$

where $f_{c,ref}$ the center frequency of the dual-tone signal, and f_{ref} is the frequency reference, which for the Ettus USRP X310 SDRs is equal to 10 MHz. The normalized LO and RF inputs to the mixer

can be given by

$$V_{RF} = \sin(2\pi f_{1,ref}t + \phi_1) + \sin(2\pi f_{2,ref}t + \phi_2), \quad (4.4)$$

$$V_{LO} = \sin(2\pi f_{1,ref}t + \phi_3) + \sin(2\pi f_{2,ref}t + \phi_4). \quad (4.5)$$

The phases ϕ_1 and ϕ_2 are obtained based on the distance separating the primary and secondary nodes. Since the signal input to the LO is obtained by splitting the received signal, the phases ϕ_3 and ϕ_4 are derived from $\phi_3 = \phi_1 + c_1$ and $\phi_4 = \phi_2 + c_2$, where c_1 and c_2 are obtained from

$$c_1 = \frac{2\pi f_{1,ref} (d_{LO} - d_{RF})}{c}, \quad (4.6)$$

$$c_2 = \frac{2\pi f_{2,ref} (d_{LO} - d_{RF})}{c}, \quad (4.7)$$

where the lengths of the cables connecting the splitter to the LO and RF inputs are represented by d_{LO} and d_{RF} respectively.

The mixer output V_{IF} contains the two input frequencies $f_{1,ref}$ and $f_{2,ref}$, along with the sum and difference of the frequencies of the signals present at the LO and RF inputs [111], and other harmonics. Knowing that, the two tones of the input signal are selected with 10 MHz tone separation, so that a 10 MHz frequency can be observed at the output of the mixer, among other frequencies. Once the output of the mixer is passed through the lowpass filter, the only remaining or dominant frequency should be the 10 MHz frequency. The output signal for the self-mixing circuit is represented by

$$V_{IF} = \frac{1}{2} [\cos(2\pi (f_{2,ref} - f_{1,ref})t + \phi_5) + \cos(2\pi (f_{2,ref} + f_{1,ref})t + \phi_6)], \quad (4.8)$$

where

$$\phi_5 = \phi_2 - \phi_1 - c_1, \quad (4.9)$$

$$\phi_6 = \phi_2 - \phi_1 + c_2. \quad (4.10)$$

The output consists of two sinusoids, the frequency of each being the difference between the two input frequencies, where $f_{2,ref} - f_{1,ref} = f_{ref}$. Because there are two sinusoids, it is desired that the relative phases be adjusted so that the signals coherently sum to maximize the output power and

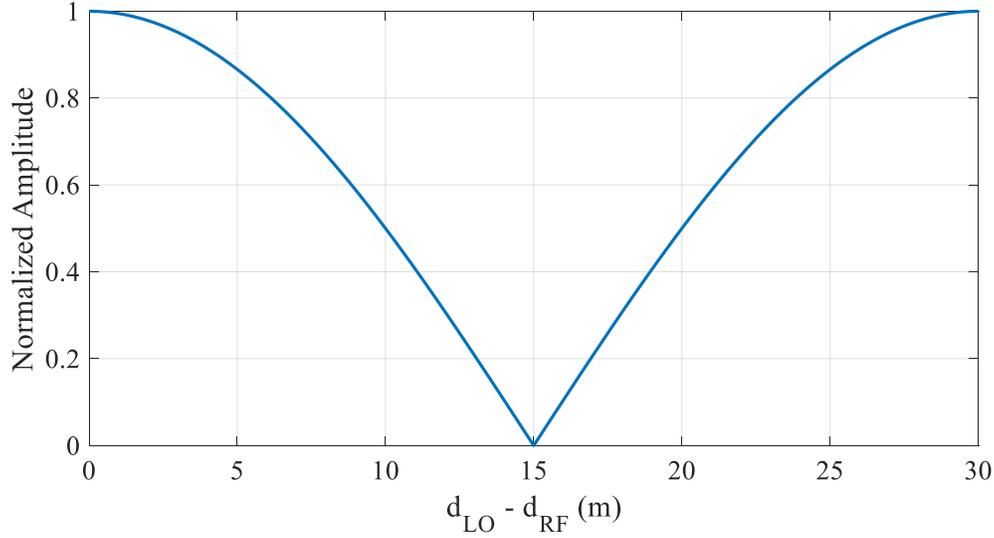


Figure 4.7: The effect of mismatch between d_{LO} and d_{RF} on the 10 MHz output from the mixer.

sufficiently drive the PLL. This can be easily ensured by choosing $d_{LO} = d_{RF}$, and as a result the phase constants ϕ_5 and ϕ_6 become equal. However, the mismatch between d_{LO} and d_{RF} needs to be very large in order to have a considerable effect on the output power of the mixer. The amplitude of the 10 MHz output versus the mismatch between d_{LO} and d_{RF} is shown in Fig. 4.7.

4.3 Impact of Doppler Frequency

When the secondary nodes are locking their frequencies to that of the primary node, any relative motion between a primary and a secondary node will result in a frequency shift at the output of the adjunct self-mixing circuit due to Doppler effects. When the relative velocity between the primary and a secondary node is $\Delta v = v_t - v_r$, where v_t is the velocity of the transmitter and v_r is the velocity of the receiver, any frequency received at the receiver of the secondary node is scaled as follows

$$f = f_{c,ref} \left(\frac{c}{c - \Delta v} \right) \approx f_{c,ref} \left(1 + \frac{\Delta v}{c} \right). \quad (4.11)$$

This approximation holds as long as $\Delta v \ll c$. The Doppler frequency shift can be computed using

$$\Delta f = \frac{f_{c,ref} \Delta v}{c}. \quad (4.12)$$

For the presented frequency synchronization approach, a dual-tone signal is being transmitted, thus each of the two transmitted frequencies $f_{1,ref}$ and $f_{2,ref}$ will have a different Doppler shift. Using (4.12),

$$\Delta f_{1,ref} = \frac{(f_{c,ref} - \frac{f_{ref}}{2})\Delta v}{c} \quad (4.13)$$

$$\Delta f_{2,ref} = \frac{(f_{c,ref} + \frac{f_{ref}}{2})\Delta v}{c}. \quad (4.14)$$

From (4.8), (4.13), and (4.14), the output of the frequency locking circuit is then

$$V_{IF} = \frac{1}{2} \cos\{2\pi[(f_{2,ref} + \Delta f_{2,ref}) - (f_{1,ref} + \Delta f_{1,ref})]t + \phi_5\} \\ + \frac{1}{2} \cos\{2\pi[(f_{2,ref} + \Delta f_{2,ref}) - (f_{1,ref} + \Delta f_{1,ref})]t + \phi_6\}. \quad (4.15)$$

The two sinusoids in the mixer output signal reside at a frequency $f_{ref} + f_d$, where the relative Doppler shift f_d is given by

$$f_d = \Delta f_{2,ref} - \Delta f_{1,ref} = \frac{f_{ref}\Delta v}{c}. \quad (4.16)$$

This result shows that the frequency shift is only affected by the Doppler shift relative to the modulation frequency, which in our case is 10 MHz. Thus, the carrier used for wireless frequency synchronization does not have any effect on the observed Doppler Shift. This result is intuitive since the Doppler shift will affect both tones with a slight difference depending on the tone separation. However, when the array is beamforming in the far field using a carrier f_c , the Doppler shift that is observed at the 10 MHz reference input, is scaled by the multipliers used to transform the 10 MHz to f_c . The Doppler shift on the transmitted carrier can be computed from (4.12) by replacing $f_{c,ref}$ with f_c . The Doppler shift for 10 MHz, 1 GHz, and 10 GHz signals is evaluated for multiple relative velocities between a primary and a secondary node as shown in Fig. 4.8. As can be seen, the Doppler shift is larger at high frequencies. However, if the frequency synchronization is achieved between relatively static nodes, such as in a drone swarm, the relative velocities between the nodes is expected to be low, leading to low Doppler shifts at a given instance.

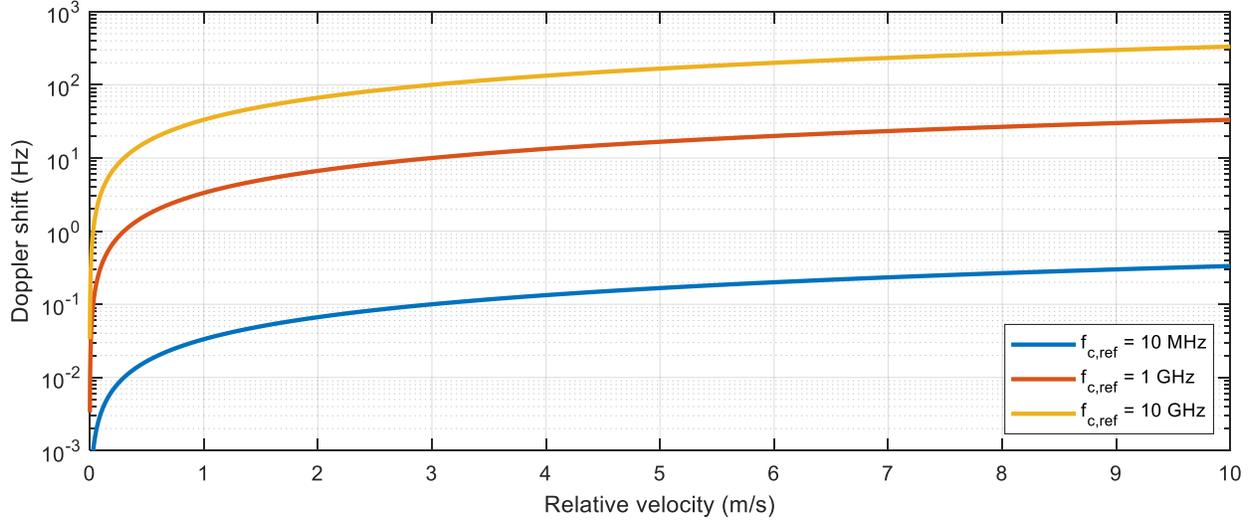


Figure 4.8: The observed Doppler shift for 10 MHz, 1 GHz, and 10 GHz signals in the case where the relative velocity between a primary and a secondary node is varied from 0 to 10 m/s.

4.4 Phase Change Due To Frequency Synchronization Signal Propagation

Coherent transmission of signals from a distributed array is only possible if time, frequency, and phase are synchronized. Thus it is important to analyze the phase shift caused by wireless frequency synchronization on every node. Since internal PLLs lock the frequency of the SDR to the reference frequency by comparing the oscillator phase to the phase of the incoming signal, the change in relative distance between the primary node and any secondary node need to be monitored so that the phase shift caused by the wireless frequency synchronization process can be tracked. Changes in distance between the antennas performing the frequency synchronization operation lead to a phase shift that can be estimated using

$$\Delta\phi = \frac{2\pi\Delta d_{IN}}{\lambda_c} \quad (4.17)$$

where Δd_{IN} is the change in inter-node distance between the primary node and the secondary node of interest. In the case where the two-tone CW frequency synchronization signal is being transmitted, the phase shift at the output of the self-mixing circuit is obtained from

$$\Delta\phi_{\text{ref}} = -2\pi \left(\frac{f_{2,ref} \cdot \Delta d_{IN}}{c} - \frac{f_{1,ref} \cdot \Delta d_{IN}}{c} \right) = -\frac{2\pi f_{ref} \cdot \Delta d_{IN}}{c}. \quad (4.18)$$

To observe the resulting phase shift at the transmitted carrier of the secondary node, one must observe how the carrier is generated from the local oscillator. If we assume that the frequency of the local oscillator is 10 MHz, a carrier frequency of 100 MHz is generated by multiplying the 10 MHz by 10. A similar multiplicative effect is observed for the phase shift at the carrier, and it can be computed from

$$\Delta\phi_{c_1} = \frac{f_c}{f_{ref}} \Delta\phi_{ref}. \quad (4.19)$$

This result shows that the two frequencies $f_{1,ref}$ and $f_{2,ref}$ have no effect on the observed phase shift at f_c , where the phase shift is only observed at the modulated frequency f_{ref} . Thus, knowing f_c and d_{IN} is enough to estimate the phase shift at the carrier.

To verify that the phase shift observed at a carrier follows (4.19) the experiment shown in Fig. 4.9 was conducted. Since the Ettus USRP X310 SDRs were used, f_{ref} was set to 10 MHz ($\lambda_{ref} = 30$ m). The two tones were transmitted at $f_{1,ref} = 3.995$ GHz and $f_{2,ref} = 4.005$ GHz, where $f_{c,ref} = 4$ GHz. The phase shift was observed at $f_c = 100$ MHz ($\lambda_c = 3$ m). Using (4.19), the anticipated phase shift was calculated as follows, $\Delta\phi_{c_1} = -10\Delta\phi_{ref}$. Throughout the experiment, the receiver of the secondary node was displaced by 3 m in total, and the phase shift was recorded with 2.5 cm increments using an oscilloscope (initial phase calibration was done). The results are shown in Fig. 4.10, where the change in phase shift with a given displacement tracked well with the expected phase shift.

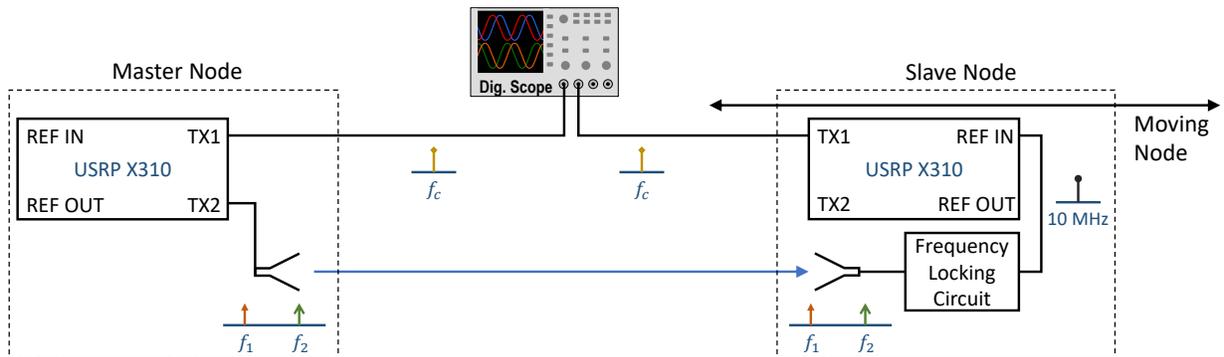


Figure 4.9: Block diagram of the setup used to measure the phase shift produced by the frequency synchronization circuit when the antenna receiving the two-tone signal is displaced. From [8] © 2021 IEEE.

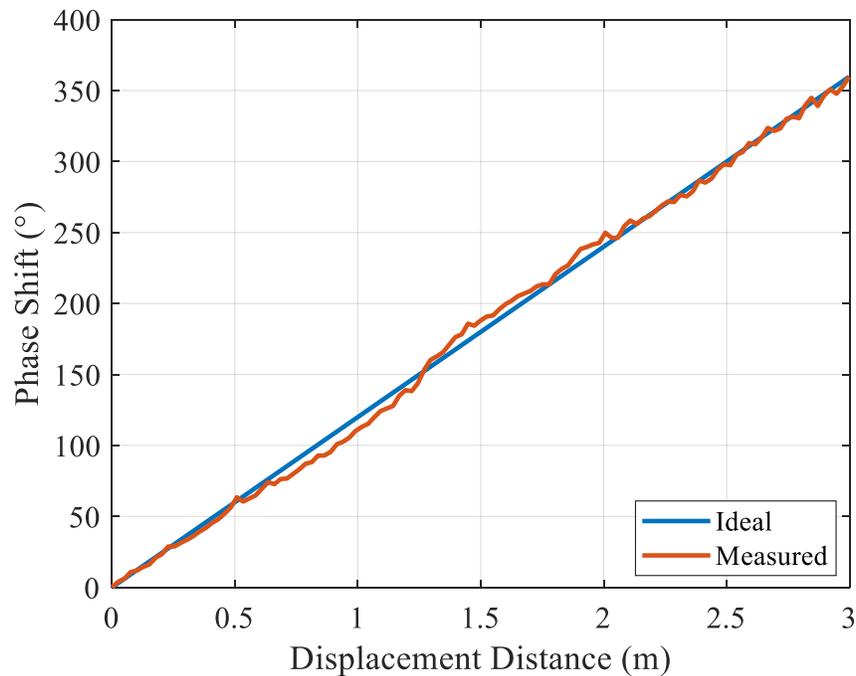


Figure 4.10: Experimental results showing the phase shifts between the primary and secondary nodes signals for multiple displacement distances. From [8] © 2021 IEEE.

CHAPTER 5

ADAPTIVE RANGING THAT SUPPORTS COHERENT BEAMFORMING OPERATIONS

I focus in this chapter on ranging using two-tone waveforms, which offer near-optimal ranging accuracies. The CRLB is evaluated for two-tone and other multi-tone waveforms that build on the high accuracy of two-tone waveforms and offer added functionalities. An adaptive ranging framework that is based on the proportional–integral–derivative (PID) controller and the high accuracy two-tone ranging waveform is demonstrated in this chapter. Adaptive ranging algorithms can offer high ranging accuracies with low spectral footprint, in particular they can maintain desired ranging accuracies that support accurate phase alignment and coherent beamforming as explained in Chapter 2 while employing ranging waveforms with the minimal desired bandwidth. The adaptive ranging approach was tested in indoor and outdoor settings.

5.1 CRLB of Range Estimates

Ranging radars operate by transmitting either CW or pulsed signals. For CW ranging signals, the phase or frequency difference between the transmitted and received pulses is used to determine the time delay of the received pulses. Whereas for pulsed ranging signals, which are the focus of this work, generally the time of arrival of the received pulse is used to estimate the time delay. The target range is determined from the time-delay estimates using the formula

$$R = \frac{c\tau}{2}, \quad (5.1)$$

where $c = 3 \times 10^8$ m/s refers to the speed of light, and τ is the time-delay estimate.

In practice, time-delay estimates are never exact, and the accuracy of the time delay estimator can be evaluated using RMSE. In the case where the target has an unknown distance, which is the case in practice, the standard deviation is evaluated instead of RMSE. Whether RMSE, standard deviation, or variance are calculated, they can be all compared to the CRLB, which represents the best achievable accuracy for an unbiased estimator. As the CRLB states, the minimum achievable

variance for an unbiased estimator is inversely proportional to the Fisher Information [112]. The Fisher Information indicates the amount of information that can be extracted from a signal. However, the CRLB only defines the minimum achievable variance for a certain ranging waveform with a given SNR, without being able to provide a method that can achieve this theoretical bound.

Radar signals can be expressed as [75]

$$s_R(x) = \alpha s(x; \mathbf{u}) + w(x), \quad (5.2)$$

where x in this section represents the time at which the signal was received, α is a complex parameter that represents the amplitude and phase fluctuations due to environment, beam pattern, multi-path, etc., $s(x; \mathbf{u})$ is a function of the unobservable parameters \mathbf{u} that need to be estimated, which include the time delay, velocity of the target, etc., and $w(x)$ is additive Gaussian white noise (AWGN). Since this chapter focuses on time-delay estimation (ranging), (5.2) can be expressed as

$$s_R(t) = \alpha g(t - \tau) + w(t). \quad (5.3)$$

In [113], it was shown that for a deterministic α and unknown \mathbf{u} , the CRLB can be determined from

$$\sigma^2(\hat{u} - u) \geq \frac{N_0}{2|\alpha|^2} \left(\int \left| \frac{\partial s(x; \mathbf{u})}{\partial u} \right|^2 dx - \frac{1}{E} \left| \int \frac{\partial s(x; \mathbf{u})^*}{\partial u} s(x; \mathbf{u}) dx \right|^2 \right)^{-1}, \quad (5.4)$$

where the signal energy $E = \int |s(x; \mathbf{u})|^2 dx$, N_0 is the noise per unit bandwidth. The first integral in (5.4) is often referred to as the mean-square bandwidth or the second moment of the energy spectrum, and it is represented by ζ_f^2 . ζ_f^2 is evaluated using Plancherel's theorem as follows

$$\begin{aligned} \zeta_f^2 &= \int \left| \frac{\partial s(x; \mathbf{u})}{\partial u} \right|^2 dx \\ &= \int \left| \frac{\partial}{\partial \tau} g(t - \tau) \right|^2 dt \\ &= \int \left| \frac{\partial}{\partial \tau} G(f) e^{-j2\pi f \tau} \right|^2 df \\ &= \int (2\pi f)^2 |G(f)|^2 df, \end{aligned} \quad (5.5)$$

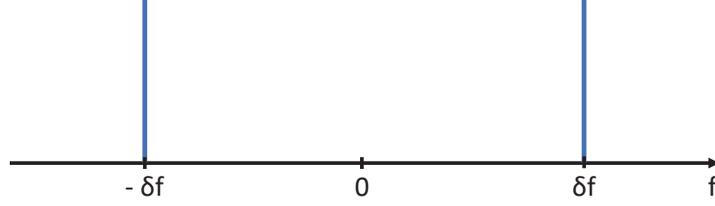


Figure 5.1: The ideal spectrum of the two-tone waveforms expressed in (5.10).

where $G(f)$ is the Fourier transform of $g(t)$. Similarly, the second integral, which represents the first moment of the energy spectrum or the mean frequency, is evaluated as follows

$$\begin{aligned}
 \mu_f^2 &= \left| \int \frac{\partial s(x; \mathbf{u})^*}{\partial u} s(x; \mathbf{u}) dx \right|^2 \\
 &= \left| \int \left(\frac{\partial}{\partial \tau} g(t - \tau) \right)^* g(t - \tau) dt \right|^2 \\
 &= \left(\int 2\pi f |G(f)|^2 df \right)^2.
 \end{aligned} \tag{5.6}$$

The term μ_f^2 can be ignored in the case where the transmitted/received signal shows a symmetry in the spectrum domain i.e. $|G(f)| = |G(-f)|$. The lower bound for the variance of time-delay estimates is obtained from (5.4), which can be expressed in terms of ζ_f^2 and μ_f^2 ,

$$\text{var}(\hat{\tau} - \tau) \geq \frac{N_0}{2|\alpha|^2 \left(\zeta_f^2 - \frac{1}{E} \mu_f^2 \right)}. \tag{5.7}$$

This lower bound for the time-delay estimates is translated to a lower bound for range estimates using (5.3), where

$$\text{var}(\hat{R} - R) = \frac{c^2}{4} \text{var}(\hat{\tau} - \tau). \tag{5.8}$$

Furthermore, the post-processing SNR (**SNR**) can be represented in (5.7) by $\mathbf{SNR} = \frac{2|\alpha|^2 E}{N_0}$; note that in some formulations in the literature, α is omitted leading to $\mathbf{SNR} = \frac{2E}{N_0}$. Thus, (5.7) can be expressed as follows

$$\text{var}(\hat{\tau} - \tau) \geq \frac{1}{\mathbf{SNR} \left(\frac{\zeta_f^2}{E} - \frac{\mu_f^2}{E^2} \right)}. \tag{5.9}$$

5.1.1 CRLB of Typical Two-Tone Waveforms

The CRLB of a ranging waveform is the lowest when the energy of the transmitted/received signal is concentrated around the edge of the spectrum. The baseband signal that minimizes the variance of the time-delay estimates can be expressed using

$$s_1(t) = e^{-j2\pi\delta ft} + e^{j2\pi\delta ft}, \quad (5.10)$$

where δf represents the one-half separation of the two tones. The ideal spectrum of $s_1(t)$ is shown in Fig. 5.1. When $2\delta f$ is equal to the maximum available bandwidth, (5.10) minimizes the CRLB, which consequently maximizes the ranging accuracy. The CRLB of $s_1(t)$ is obtained from the following

$$\begin{aligned} S_1(f) &= \int_{-\infty}^{\infty} s_1(t) e^{-j2\pi ft} dt \\ &= \int_{-\infty}^{\infty} (e^{-j2\pi\delta ft} + e^{j2\pi\delta ft}) e^{-j2\pi ft} dt \\ &= \delta(f + \delta f) + \delta(f - \delta f). \end{aligned} \quad (5.11)$$

$$\begin{aligned} |S_1(f)|^2 &= |\delta(f + \delta f) + \delta(f - \delta f)|^2 \\ &= \delta^2(f + \delta f) + \delta^2(f - \delta f) + 2\delta(f + \delta f)\delta(f - \delta f) \\ &= \delta(f + \delta f) + \delta(f - \delta f). \end{aligned} \quad (5.12)$$

$$\begin{aligned} \int_{-\infty}^{\infty} (2\pi f)^2 |S_1(f)|^2 df &= 4\pi^2 \int_{-\infty}^{\infty} f^2 [\delta(f + \delta f) + \delta(f - \delta f)] df \\ &= 4\pi^2 (-\delta f)^2 + 4\pi^2 (\delta f)^2 = 8\pi^2 (\delta f)^2. \end{aligned} \quad (5.13)$$

$$\int_{-\infty}^{\infty} |S_1(f)|^2 df = \int_{-\infty}^{\infty} \delta(f + \delta f) + \delta(f - \delta f) df = 1 + 1 = 2. \quad (5.14)$$

$$\frac{\zeta_f^2}{E} = \frac{\int_{-\infty}^{\infty} (2\pi f)^2 |S_1(f)|^2 df}{\int_{-\infty}^{\infty} |S_1(f)|^2 df} = \frac{8\pi^2 (\delta f)^2}{2} = (2\pi\delta f)^2. \quad (5.15)$$

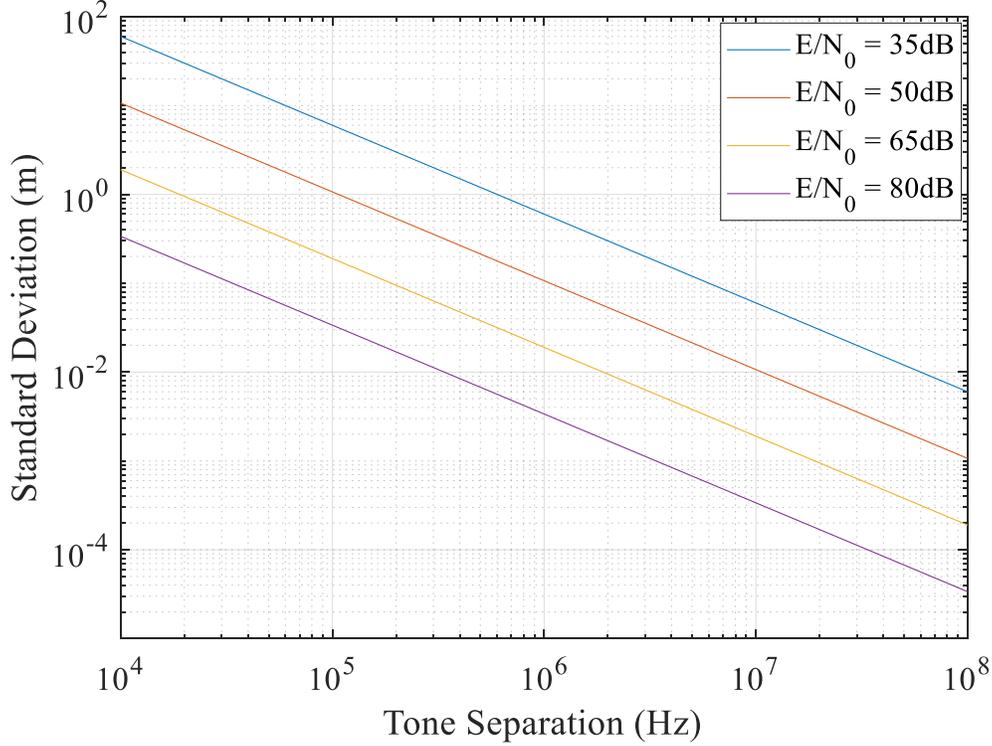


Figure 5.2: The minimum achievable standard deviation/RMSE for two-tone CW ranging signals for various values of E/N_0 . From [10] © 2021 IEEE.

$$\sigma_{t,1}^2 = \text{var}(\hat{\tau} - \tau) = \frac{1}{\frac{\zeta_f^2}{E} \text{SNR}} = \frac{1}{(2\pi\delta f)^2 \text{SNR}}. \quad (5.16)$$

In (5.16), $\mu_f^2 = 0$ since the waveform is symmetric in the spectrum domain. Further, (5.16) shows the CRLB for two-tone CW signals, however, in practice, the ranging waveforms are evaluated for pulsed signals with limited time duration, especially for the case of multi-tone signals. Thus, the spectrum of the two-tone signal is spread around the two tones, instead of having two perfect impulse functions at $-\delta f$ and δf . For pulses that are long relative to the carrier period, this effect is minimal. This change in spectrum shifts the CRLB to slightly higher values, which can generally be obtained using numerical methods. Nevertheless, since the CRLB serves as an indicator for the maximum achievable accuracy, and since the form representing CW signals achieves the lowest CRLB, (5.16) can be used to represent any two-tone waveform where the two tones are equidistant around the center of the spectrum. The minimum achievable standard deviation/RMSE of two-tone waveforms is depicted in Fig. 5.2 for multiple E/N_0 values.

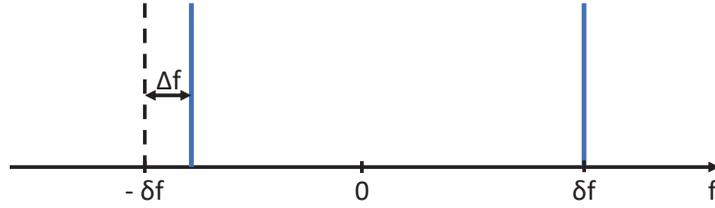


Figure 5.3: The ideal spectrum of the two-tone waveforms expressed in (5.17).

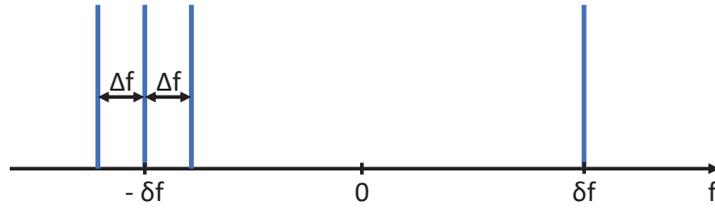


Figure 5.4: The ideal spectrum of the four-tone waveforms expressed in (5.18).

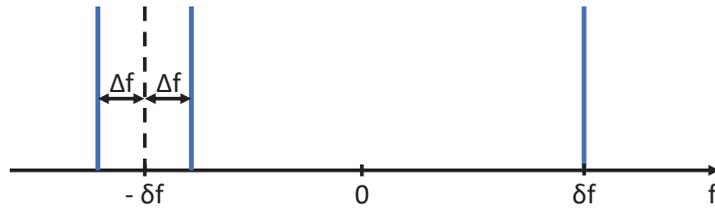


Figure 5.5: The ideal spectrum of the three-tone waveforms expressed in (5.18).

Although two-tone signals have the lowest CRLB among all other waveforms, they are rarely used in the literature for radar applications due to their ambiguous nature; in many cases, it is not possible to detect multiple targets while using two-tone ranging waveforms. Nevertheless, two-tone waveforms provide an opportunity to achieve the desired ranging accuracies for this work, especially that open-loop distributed phased arrays require very accurate range estimates. The ambitiousness of two-tone ranging waveforms is resolved by equipping the targeted nodes with active repeaters. The active repeaters amplify the received signals and retransmit them back to their source, allowing the radar to observe the target as a point source. Such an approach improves the ranging accuracy due to the applied amplification in this process and suppresses the reflections from the neighboring targets.

5.1.2 CRLB of Multi-Tone Waveforms

In addition to accurate range estimation, in many cases it is beneficial to use waveforms that provided added functionality. Given that wireless phase and frequency synchronization are of interest for coherent distributed arrays, one can use waveforms that can perform phase and frequency synchronization jointly. As it will be seen in the next chapter, wireless frequency synchronization is achieved in this work using signals with 10 MHz tone separation, and these tones can be modulated for instance on the ranging waveforms. To study the implication of modifying or adding extra modulations to a preexisting ranging waveform, the CRLB of the three ranging waveforms

$$s_2(t) = e^{j2\pi(\Delta f - \delta f)t} + e^{j2\pi\delta ft}, \quad (5.17)$$

$$s_3(t) = e^{j2\pi(-\Delta f - \delta f)t} + e^{-j2\pi\delta ft} + e^{j2\pi(\Delta f - \delta f)t} + e^{j2\pi\delta ft}, \quad (5.18)$$

and

$$s_4(t) = e^{j2\pi(-\Delta f - \delta f)t} + e^{j2\pi(\Delta f - \delta f)t} + e^{j2\pi\delta ft}, \quad (5.19)$$

is evaluated below. Although the spectrum of these three waveforms is not perfectly symmetric, μ_f^2 is going to be approximated as 0, which is a valid case for widely separated tones [114]. The spectra of $s_2(t)$, $s_3(t)$, and $s_4(t)$ are shown in Figs. 5.3, 5.4, and 5.5. $\sigma_{t,2}^2$ is computed as follows

$$\begin{aligned} S_2(f) &= \int_{-\infty}^{\infty} s_2(t) e^{-j2\pi ft} dt \\ &= \int_{-\infty}^{\infty} (e^{j2\pi(-\delta f + \Delta f)t} + e^{j2\pi\delta ft}) e^{-j2\pi ft} dt \\ &= \delta(f - (\Delta f - \delta f)) + \delta(f - \delta f). \end{aligned} \quad (5.20)$$

$$\begin{aligned} |S_2(f)|^2 &= |\delta(f - (\Delta f - \delta f)) + \delta(f - \delta f)|^2 \\ &= \delta^2(f - (\Delta f - \delta f)) + \delta^2(f - \delta f) + 2\delta(f - (\Delta f - \delta f))\delta(f - \delta f) \\ &= \delta(f - (\Delta f - \delta f)) + \delta(f - \delta f). \end{aligned} \quad (5.21)$$

$$\begin{aligned}
\int_{-\infty}^{\infty} (2\pi f)^2 |S_2(f)|^2 df &= 4\pi^2 \int_{-\infty}^{\infty} f^2 [\delta(f - (\Delta f - \delta f)) + \delta(f - \delta f)] df \\
&= 4\pi^2 (\Delta f - \delta f)^2 + 4\pi^2 (\delta f)^2 \\
&= 4\pi^2 (\Delta f^2 + \delta f^2 - 2\Delta f \delta f) + 4\pi^2 \delta f^2 \\
&= 4\pi^2 (\Delta f^2 + 2\delta f^2 - 2\Delta f \delta f).
\end{aligned} \tag{5.22}$$

$$\int_{-\infty}^{\infty} |S_2(f)|^2 df = \int_{-\infty}^{\infty} \delta(f - (\Delta f - \delta f)) + \delta(f - \delta f) df = 1 + 1 = 2. \tag{5.23}$$

$$\frac{\zeta_f^2}{E} = \frac{\int_{-\infty}^{\infty} (2\pi f)^2 |S_2(f)|^2 df}{\int_{-\infty}^{\infty} |S_2(f)|^2 df} = \frac{4\pi^2 (\Delta f^2 + 2\delta f^2 - 2\Delta f \delta f)}{2} = 2\pi^2 (\Delta f^2 + 2\delta f^2 - 2\Delta f \delta f). \tag{5.24}$$

$$\sigma_{t,2}^2 = \frac{1}{\frac{\zeta_f^2}{E} \text{SNR}} = \frac{1}{2\pi^2 (\Delta f^2 + 2\delta f^2 - 2\Delta f \delta f) \text{SNR}}. \tag{5.25}$$

Similarly, $\sigma_{t,3}^2$ is computed as follows

$$\begin{aligned}
S_3(f) &= \int_{-\infty}^{\infty} s_3(t) e^{-j2\pi ft} dt \\
&= \int_{-\infty}^{\infty} (e^{j2\pi(-\delta f - \Delta f)t} + e^{-j2\pi\delta ft} + e^{j2\pi(-\delta f + \Delta f)t} + e^{j2\pi\delta ft}) e^{-j2\pi ft} dt \\
&= \delta(f - (-\delta f - \Delta f)) + \delta(f + \delta f) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f).
\end{aligned} \tag{5.26}$$

$$\begin{aligned}
|S_3(f)|^2 &= |\delta(f - (-\delta f - \Delta f)) + \delta(f + \delta f) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f)|^2 \\
&= \delta^2(f - (-\delta f - \Delta f)) + \delta^2(f + \delta f) + \delta^2(f - (-\delta f + \Delta f)) + \delta^2(f - \delta f) \\
&\quad + 2\delta(f - (-\delta f - \Delta f))\delta(f + \delta f) + 2\delta(f - (-\delta f - \Delta f))\delta(f - (-\delta f + \Delta f)) \\
&\quad + 2\delta(f - (-\delta f - \Delta f))\delta(f - \delta f) + 2\delta(f + \delta f)\delta(f - (-\delta f + \Delta f)) \\
&\quad + 2\delta(f + \delta f)\delta(f - \delta f) + 2\delta(f - (-\delta f + \Delta f))\delta(f - \delta f) \\
&= \delta(f - (-\delta f - \Delta f)) + \delta(f + \delta f) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f).
\end{aligned} \tag{5.27}$$

$$\begin{aligned}
&\int_{-\infty}^{\infty} (2\pi f)^2 |S_3(f)|^2 df \\
&= 4\pi^2 \int_{-\infty}^{\infty} f^2 [\delta(f - (-\delta f - \Delta f)) + \delta(f + \delta f) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f)] df \\
&= 4\pi^2 [(-\delta f - \Delta f)^2 + (-\delta f)^2 + (-\delta f + \Delta f)^2 + (\delta f)^2] \\
&= 4\pi^2 (4\delta f^2 + 2\Delta f^2).
\end{aligned} \tag{5.28}$$

$$\begin{aligned}
\int_{-\infty}^{\infty} |S_3(f)|^2 df &= \int_{-\infty}^{\infty} \delta(f - (-\delta f - \Delta f)) + \delta(f + \delta f) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f) df \\
&= 1 + 1 + 1 + 1 = 4.
\end{aligned} \tag{5.29}$$

$$\frac{\zeta_f^2}{E} = \frac{\int_{-\infty}^{\infty} (2\pi f)^2 |S_3(f)|^2 df}{\int_{-\infty}^{\infty} |S_3(f)|^2 df} = \frac{4\pi^2 (4\delta f^2 + 2\Delta f^2)}{4} = 4\pi^2 \delta f^2 + 2\pi^2 \Delta f^2. \tag{5.30}$$

$$\sigma_{t,3}^2 = \frac{1}{\frac{\zeta_f^2}{E} \mathbf{SNR}} = \frac{1}{(4\pi^2 \delta f^2 + 2\pi^2 \Delta f^2) \mathbf{SNR}}. \tag{5.31}$$

Finally, $\sigma_{t,4}^2$ is obtained from the following

$$\begin{aligned}
S_4(f) &= \int_{-\infty}^{\infty} s_4(t) e^{-j2\pi ft} dt \\
&= \int_{-\infty}^{\infty} (e^{j2\pi(-\delta f - \Delta f)t} + e^{j2\pi(-\delta f + \Delta f)t} + e^{j2\pi\delta f t}) e^{-j2\pi ft} dt \\
&= \delta(f - (-\delta f - \Delta f)) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f).
\end{aligned} \tag{5.32}$$

$$\begin{aligned}
|S_4(f)|^2 &= |\delta(f - (-\delta f - \Delta f)) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f)|^2 \\
&= \delta^2(f - (-\delta f - \Delta f)) + \delta^2(f - (-\delta f + \Delta f)) + \delta^2(f - \delta f) \\
&\quad + 2\delta(f - (-\delta f - \Delta f))\delta(f - (-\delta f + \Delta f)) + 2\delta(f - (-\delta f - \Delta f))\delta(f - \delta f) \\
&\quad + 2\delta(f - (-\delta f + \Delta f))\delta(f - \delta f) \\
&= \delta(f - (-\delta f - \Delta f)) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f).
\end{aligned} \tag{5.33}$$

$$\begin{aligned}
&\int_{-\infty}^{\infty} (2\pi f)^2 |S_4(f)|^2 df \\
&= 4\pi^2 \int_{-\infty}^{\infty} f^2 [\delta(f - (-\delta f - \Delta f)) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f)] df \\
&= 4\pi^2 [(-\delta f - \Delta f)^2 + (-\delta f + \Delta f)^2 + (\delta f)^2] = 4\pi^2 (3\delta f^2 + 2\Delta f^2).
\end{aligned} \tag{5.34}$$

$$\begin{aligned}
\int_{-\infty}^{\infty} |S_4(f)|^2 df &= \int_{-\infty}^{\infty} \delta(f - (-\delta f - \Delta f)) + \delta(f - (-\delta f + \Delta f)) + \delta(f - \delta f) df \\
&= 1 + 1 + 1 = 3.
\end{aligned} \tag{5.35}$$

$$\frac{\zeta_f^2}{E} = \frac{\int_{-\infty}^{\infty} (2\pi f)^2 |S_4(f)|^2 df}{\int_{-\infty}^{\infty} |S_4(f)|^2 df} = \frac{4\pi^2 (3\delta f^2 + 2\Delta f^2)}{3} = 4\pi^2 \delta f^2 + \frac{8}{3}\pi^2 \Delta f^2. \tag{5.36}$$

$$\sigma_{t,4}^2 = \frac{1}{\frac{\zeta_f^2}{E} \text{SNR}} = \frac{1}{(4\pi^2 \delta f^2 + \frac{8}{3}\pi^2 \Delta f^2) \text{SNR}}. \quad (5.37)$$

A comparison between $\sigma_{t,1}^2$, $\sigma_{t,2}^2$, $\sigma_{t,3}^2$, and $\sigma_{t,4}^2$ is shown using a numerical example. In this example, the total signal bandwidth is set to 200 MHz (thus δf was varying depending on the waveform) and $\Delta f = 5$ MHz. The obtained CRLB values are as follows: $\sigma_{t,1}^2 = 2.53 \times 10^{-18}/\text{SNR } s^2$, $\sigma_{t,2}^2 = 2.53 \times 10^{-18}/\text{SNR } s^2$, $\sigma_{t,3}^2 = 2.66 \times 10^{-18}/\text{SNR } s^2$, and $\sigma_{t,4}^2 = 2.66 \times 10^{-18}/\text{SNR } s^2$. $\sigma_{t,1}^2$ and $\sigma_{t,2}^2$ are equal because we considered the same bandwidth for both signals, however, if a fixed value of 100 MHz for δf is used, $\sigma_{t,2}^2$ would be equal to $2.66 \times 10^{-18}/\text{SNR } s^2$. These results show that the accurate two-tone waveform can be amended to contain one or multiple additional tones with minor degradation to the ranging accuracy, as long as δf is in the range of a magnitude larger than Δf . The additional tones can be used in future work for wireless frequency synchronization or time synchronization. Nevertheless, let us consider that a tone separation of 10 MHz is needed for wireless frequency synchronization, if this tone separation achieves desirable ranging accuracies, a two-tone waveform can be used for simultaneous ranging and wireless frequency synchronization without any added modulation.

5.2 High Accuracy Ranging

Accurate time-delay estimation (or ranging) is one of the main pillars to achieve coherent transmission in distributed arrays operating without any feedback from the target or other external systems. As discussed in Chapter 2, the coherent gain is directly affected by the accuracy of the time-delay estimates, where the coherent gain can be increased by improving the localization estimates. Accurate estimates can be achieved using two-tone waveforms, since they offer optimal ranging accuracies. Two-tone CW signals were introduced in (5.10) in the continuous-time domain, however, range estimation is done using the digital form of the received signals. In the case of two-tone signals, the transmitted pulsed signal can be represented by

$$s[n] = (u[n] - u[n - n_T + 1]) \left(e^{j2\pi f_1 \frac{n}{f_s}} + e^{j2\pi f_2 \frac{n}{f_s}} \right), \quad (5.38)$$

and the received pulsed signal by

$$r[n] = (u[n - \lceil \tau f_s \rceil] - u[n - \lfloor \tau f_s \rfloor - n_T + 1]) \times \left(e^{j2\pi f_1 (\frac{n}{f_s} - \tau)} + e^{j2\pi f_2 (\frac{n}{f_s} - \tau)} \right) + \text{AWGN}[n], \quad (5.39)$$

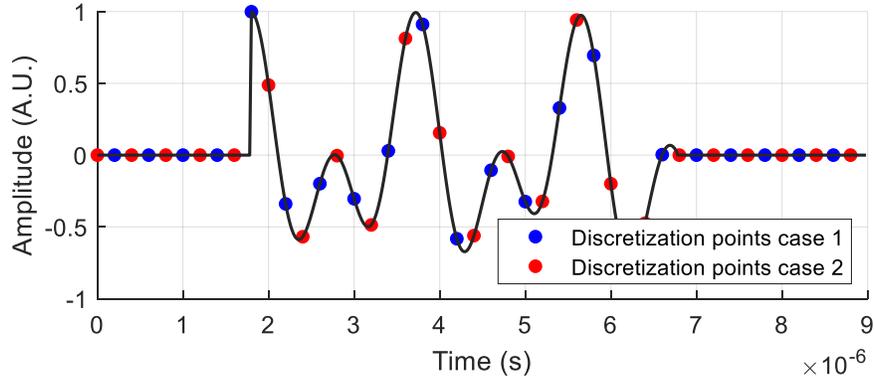
where $u[\cdot]$ is the unit step function, n_T is the sample representing the end of the pulse with a pulse width $T = n_T/f_s$, f_1 and f_2 are the two tones of the transmitted and received signals, which can be defined as $f_1 = -\delta f$ and $f_2 = +\delta f$ for the baseband representation or as $f_1 = f_0 - \delta f$ and $f_2 = f_0 + \delta f$ if the center frequency f_0 is being considered. The sampling rate is represented using f_s , τ is the time delay, $\text{AWGN}[n]$ represents the additive white Gaussian noise, $\lceil \cdot \rceil$ maps its argument to the least integer greater than or equal to its value, and $\lfloor \cdot \rfloor$ maps its argument to the greatest integer less than or equal to its value.

The accuracy of the range estimates can be further improved by utilizing matched filtering or pulse compression. The matched filter is the optimal linear filter that maximizes the SNR of the received pulses by correlating the received signals with the transmitted signals. This approach maximizes the SNR by providing an additional processing gain that is represented by the time-bandwidth products, which is defined by TBW_r , where BW_r represents the receiver bandwidth that can be determined from the sampling frequency. The matched filter output is obtained from

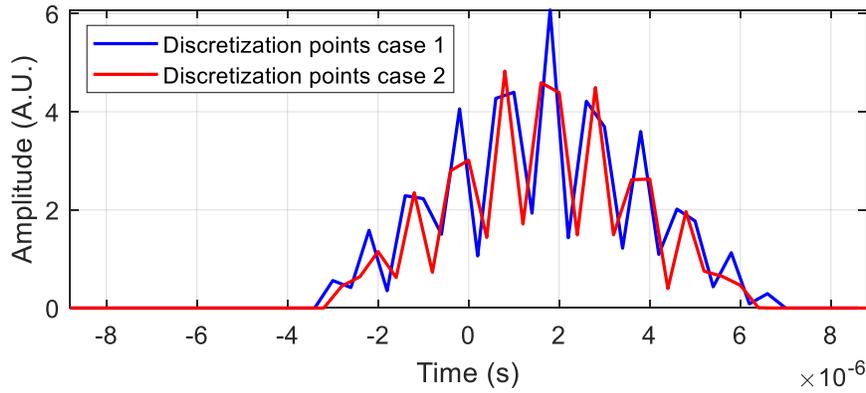
$$y[n] = r[n] * s^*[-n] = \text{IFT} \{R[k]S^*[k]\}, \quad (5.40)$$

where $R[k]$ and $S[k]$ are the discrete Fourier transforms of $r[n]$ and $s[n]$, $*$ represents convolution, $(^*)$ represents complex conjugate, and $\text{IFT} \{ \cdot \}$ is the inverse Fourier transform.

Time-delay estimation is done by determining the peak of the matched filter output. However, determining the peak is a challenging task, given that the peak needs to be estimated from a discrete matched filter output. To visualize this challenging task, Fig. 5.6 shows the same received two-tone pulse with two different discretization instances (the noise/interference was set to zero in this example), along with the matched filter output for the two scenarios. The matched filter obtained from the second set of discretization points (which represents the usual case) illustrates the complications faced while estimating the peak of the matched filter output. Depending on the delay of



(a)



(b)

Figure 5.6: (a) The in-phase part of a two-tone signal with $f_1 = 53$ kHz, $f_2 = 1.03$ MHz, and a sampling frequency of $f_s = 2.5$ MHz showing discretizations equivalent to two different delays. (b) Output of the matched filter for both scenarios, showing the error due to discretization.

the received signal, which produces a different version of the discretized received signal/matched filter output, the main lobe of the discretized filter output is truncated differently.

Also, it can be seen from Fig. 5.6b, that the matched filter output of two-tone signals is highly ambiguous. This could lead for the cases similar to the second set of discretization points to misidentification of the peak, given that the sidelobes can have higher peaks due to discretization. This problem was resolved in [115], where a disambiguation pulse was transmitted between every two ranging pulses in order to detect the main peak of the matched filter output. For cases where ranging is done between nodes with small relative motion, such as drones flying in formation, the disambiguation pulses can be transmitted in between multiple ranging pulses to speed up the

ranging process. The discrete form of the transmitted disambiguation pulse can be represented by

$$s_{dis}[n] = (u[n] - u[n - \lfloor 4f_s^2/f_2 \rfloor + 1]) e^{j2\pi \frac{f_2}{4f_s} n}. \quad (5.41)$$

As can be seen, the pulse width is equal to one cycle of the transmitted frequency $f_2/4$. Since only one cycle of this frequency is transmitted, the matched filter output of the disambiguation pulse consists of only one lobe. Thus, the output of the matched filter is unambiguous and by estimating its peak, it is possible to determine the main lobe of the matched filter output of the ranging pulse. However, given that the pulse width of the disambiguation pulse is considerably smaller than that of the ranging pulse, the time-delay estimation of the disambiguation pulse is very coarse, hence the ranging pulse is needed to determine the target range with high accuracy. The frequency $f_2/4$ was selected for the disambiguation pulse so that the transmitted pulse is long enough to produce a desirable time-bandwidth product (even lower frequencies can be selected, depending on the frequency f_2). Consequently, the peak of the matched filter output of the disambiguation pulse will have multiple discretization points on the output lobe; at least 8 points in the case where f_2 is too close to f_s . This is needed since the output lobe from the disambiguation pulse is much wider than the lobes of the two-tone waveform. Further analysis on the disambiguation pulse is available in [115]. An example of the disambiguation of a two-tone pulse is shown in Fig. 5.7, where a two-tone waveform with frequencies $f_1 = 20$ kHz and $f_2 = 7.52$ MHz was used along with a disambiguation pulse with a 1.875 MHz frequency. Since one cycle of the disambiguation pulse was transmitted, the pulse width of the disambiguation pulse was 533 ns.

5.2.1 Minimization of Ranging Biases

Generally the time-delay estimation variance cannot reach the lower bound due to signal discretization, and hardware-induced errors, among other factors. Furthermore, estimation algorithms impact the ranging performance and can lead to increased variance and bias. Two-/multi-tone waveforms are prone to bias errors especially when the tones are close to zero or the Nyquist rate. As mentioned earlier, time-delay estimation or ranging is done by estimating the peak of the main

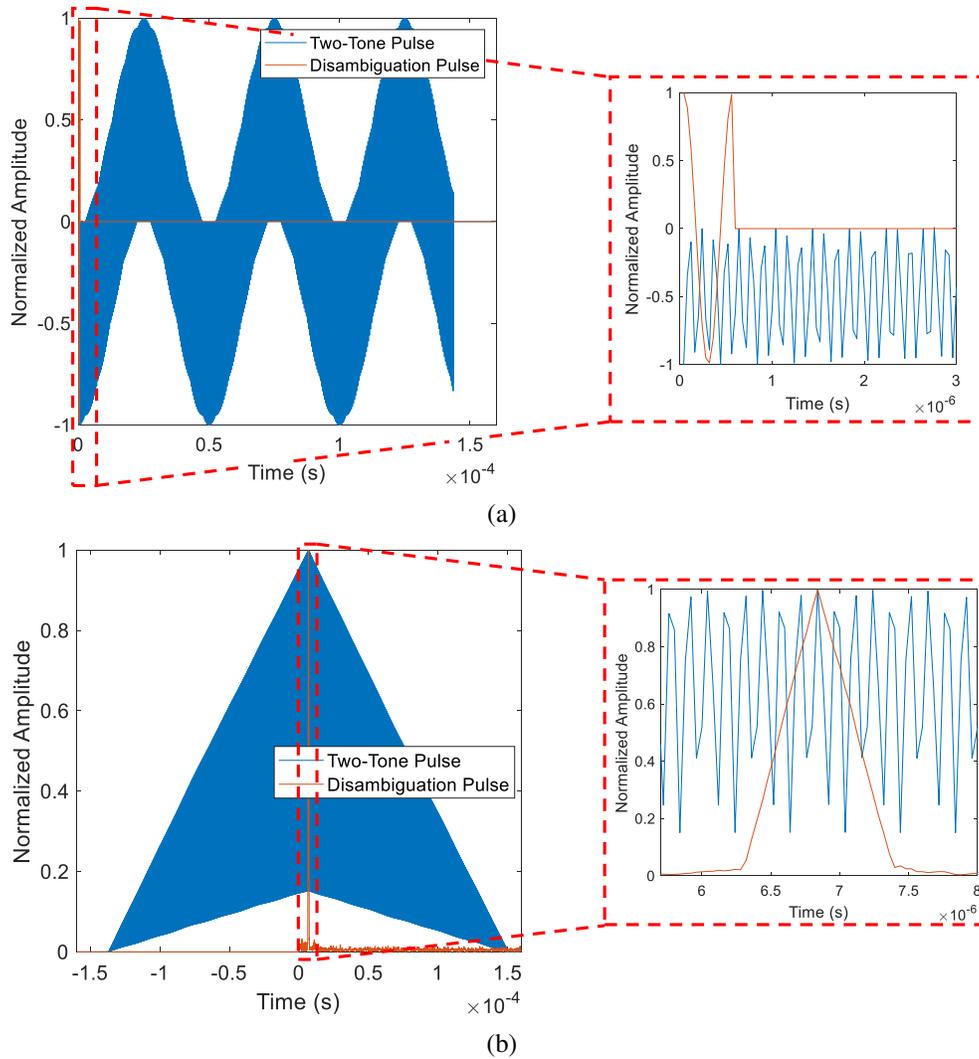


Figure 5.7: (a) Two-tone ranging pulse with $f_1 = 20$ kHz and $f_2 = 7.52$ MHz, along with a disambiguation pulse with $f_d = 1.875$ MHz. These signals were sampled at 25 MSps. (b) Output of the matched filter for both the ranging pulse and the disambiguation pulse. From [10].

lobe of the matched filter output, and given that estimation process is deals with discretized signals, almost always, the peak resides between two discretization points. Estimating the actual peak is using multiple methods, where interpolation is one of the most common ones. Although generally interpolation provides accurate estimates, it tends to be slow, hence in [116] the nonlinear least squares sinc fitting (NL-LS) was used for ranging. NL-LS is a common peak estimator, and it has been used in the literature for hyperbolic and Gaussian functions [117–119]. Although these common peak estimator perform very well for most of the ranging waveforms, it is shown in

this dissertation that in many cases, they produce biased estimates. Thus, the new peak estimator matched filter least-squares (MF-LS) was developed to minimize the bias.

5.2.1.1 Peak Estimators

Estimation Using Interpolation As the received signal is being captured, it is discretized with time intervals $T_s = 1/f_s$. Thus, if no other peak estimator is used to detect the actual peak, a coarse grid with T_s time spacing is formed, and the time-delay estimation will have an extremely low accuracy; if the sampling rate is set to 20 MSps for example, the time-delay of the received pulse will have a coarse grid of 50 ns, yielding a range grid of 7.5 m. Obviously, if no extra processing is done, it is not possible to use these estimates to synchronize an open-loop coherent distributed array. Refinement can be done by interpolating L samples preceding and following the initial time-delay estimate. The initial time-delay estimate is obtained from the coarse estimate of the peak n_{pk} of the matched filter by evaluating $\tau_c = n_{pk}T_s$. Since the matched filter output has a sinusoidal behavior, a spline interpolation with K points between each two points is desirable. Thus, the output of this spline function will contain $2LK + 1$ samples. Once interpolated, a refined time shift to the initial time-delay estimate is calculated from $\tau_r = k_0T_s/K$, where k_0 is the peak of the interpolated waveform with $2LK + 1$ samples. By adding τ_r to τ_c , the range is obtained from $R = c(\tau_c + \tau_r)/2$. The interpolated waveform will have a large number of samples, especially with large L values. Nevertheless, if no disambiguation techniques are used to detect the main lobe, L has to be large in order to have a higher chances to detect the main lobe. However, if a disambiguation pulse is used, L can be set as low as 1, but in certain cases, a higher value of L can be desirable to minimize the bias.

Estimation Using Sinc Nonlinear Least Squares (NL-LS) As demonstrated in [116], NL-LS can be implemented efficiently for LFM and pseudo-noise coded ranging waveforms. The main advantage of this approach is that it is a fast algorithm with much lower complexity than interpolation, mainly because only three points are required to estimate the refined peak of a given

Input: $f_1, f_2, f_s, \mathbf{y}$.

Output: τ_e , which is the estimated time delay.

$\lambda_0 = y_1; \lambda_1 \approx 0; \lambda_2 = 2(f_2 - f_1)/f_s;$

iter = 5 (preferably iter \geq 4);

for $m = 1$ **to** iter **do**

$$\left| \begin{array}{l} J = \left[\frac{f(\mathbf{x}; \boldsymbol{\lambda}_m)}{\lambda_0}, \frac{f(\mathbf{x}; \boldsymbol{\lambda}_m)}{\lambda_1}, \frac{f(\mathbf{x}; \boldsymbol{\lambda}_m)}{\lambda_2} \right]; \\ \Delta \mathbf{y} = \mathbf{y} - f(\mathbf{x}, \boldsymbol{\lambda}_m); \\ \Delta \boldsymbol{\lambda} = (J^T J)^{-1} J^T \Delta \mathbf{y}; \\ \boldsymbol{\lambda}_{m+1} = \boldsymbol{\lambda}_m \Delta \boldsymbol{\lambda}; \end{array} \right.$$

end

$\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}_{m+1};$

$\tilde{n}_{pk} = n_{pk} + \tilde{\lambda}_1;$

$\tau_e = \tilde{n}_{pk}/f_s;$

return τ_e

Algorithm 5.1: NL-LS

waveform. The main objective of NL-LS is to estimate the peak by minimizing the residual error using

$$L = \sum_{i=0}^2 (y_i - f(x_i, \boldsymbol{\lambda}))^2, \quad (5.42)$$

where

$$f(\mathbf{x}; \boldsymbol{\lambda}) = \lambda_0 \text{sinc}((\mathbf{x} - \lambda_1)\lambda_2), \quad (5.43)$$

$$\mathbf{y} : y_i = |d[n_{pk-1+i}]| \quad i \in \{0, 2\}, \quad (5.44)$$

$$\mathbf{x} = [-1, 0, 1]^T, \quad (5.45)$$

$$\boldsymbol{\lambda} = [\lambda_0, \lambda_1, \lambda_2]^T, \quad (5.46)$$

where y_i are the samples from $y[n]$ around the estimated peak, $f(x_i, \boldsymbol{\lambda})$ is the predicted matched filter output main lobe for the samples x_i around the peak, \mathbf{x} contains the elements x_i representing the relative location of the matched filter output samples to the peak of the matched filter output, $d[n_{pk-1+i}]$ represents the amplitude of the matched filter output around the peak n_{pk} , and $\boldsymbol{\lambda}$ is the vector of the coefficients. Gauss–Newton optimization, which uses the following gradients with

Input: $n_T, T, f_1, f_2, f_s, s[n], y[n], H$.

Output: τ_e .

Obtain the initial peak estimate n_{pk} ;

for $h = 1$ to H **do**

$$dt_h = n_{pk}T_s - 0.5/(f_2 - f_1) + (h/H)/(f_2 - f_1);$$

$$t_h[n] = -dt_h : 1/f_s : 2(n_T - 1)/f_s - dt_h;$$

$$\tilde{r}[n] = (u[t_h] - u[t_h - T]) (e^{j2\pi f_1 t_h} + e^{j2\pi f_2 t_h});$$

$$\tilde{y}[n] = \tilde{r}[n] \otimes s^*[-n];$$

$$RSS[h] = \sum_{n=0}^N (y[n]/\max(y) - \tilde{y}[n]/\max(\tilde{y}))^2;$$

end

Using the interpolation algorithm, estimate h_{min} , the refined value that minimizes RSS ;

$$\text{Local time delay: } dt_{loc} = -0.5/(f_2 - f_1) + (h_{min}/H)/(f_2 - f_1);$$

$$\text{Estimate the time of arrival using: } \tau_e = n_{pk}T_s + dt_{loc};$$

return τ_e

Algorithm 5.2: MF-LS

respect to the model parameters, is used to solve (5.42)

$$\frac{f(\mathbf{x}; \boldsymbol{\lambda})}{\lambda_0} = \text{sinc}(\lambda_2(\mathbf{x} - \lambda_1)), \quad (5.47)$$

$$\frac{f(\mathbf{x}; \boldsymbol{\lambda})}{\lambda_1} = \frac{\lambda_0 [\text{sinc}(\lambda_2(\mathbf{x} - \lambda_1)) - \cos(\pi\lambda_2(\mathbf{x} - \lambda_1))]}{\mathbf{x} - \lambda_1}, \quad (5.48)$$

$$\frac{f(\mathbf{x}; \boldsymbol{\lambda})}{\lambda_2} = \frac{\lambda_0 [\cos(\pi\lambda_2(\mathbf{x} - \lambda_1)) - \text{sinc}(\lambda_2(\mathbf{x} - \lambda_1))]}{\lambda_2}. \quad (5.49)$$

NL-LS peak estimation steps are shown in Algorithm 5.1. More details are shown in [116]. In general, this approach is fast and gives reliable estimates in terms of accuracy and bias for many waveforms, but as shown later in this section, this approach suffers from significant bias for two-tone waveforms with bandwidths that are small or near the Nyquist frequency.

Estimation Using Matched Filter Least Squares (MF-LS) MF-LS was designed to fit any ranging waveform, where the algorithm works by comparing the computed matched filtered output to a set of H potential responses, where each of the considered responses corresponds to a certain shift in the discretization points. The logic behind this method can be visualized in Fig. 5.6; depending on when was the pulse received relatively to the discretization points, the matched filter output will have a varying response. By comparing the obtained response to a set of candidate responses, it is possible to determine the time delay. This method should always work as long as

the transfer function that transforms the transmitted signal to the received signal can be determined. Once the obtained matched filter output is compared to the H potential candidates, the normalized residual sum of squares (RSS) is used to assess the every potential response using

$$RSS[h] = \sum_{n=0}^N (y[n]/\max(y) - \tilde{y}[n]/\max(\tilde{y}))^2, \quad (5.50)$$

where $N + 1$ is the total number of samples of a given waveform. $\tilde{y}[n] = \tilde{r}[n] * s^*[-n]$ represents the predicted value of $y[n]$, and $\tilde{r}[n]$ represents the predicted discretized version of the received signal for every predicted time shift dt_h (as shown in Fig. 5.6a). The values of dt_h are selected such that a linear vector (equally sampled) of candidate time delays around the coarse estimate of the matched filter output n_{pk} is considered.

By using (5.50) for all H potential responses, a curve similar to the ones shown in Fig. 5.8 is obtained and used to determine the time delay by employing the method of least squares. For example, with $H = 50$ potential matched filter outputs, such as in Fig. 5.8, the refined estimate of the local time delay dt_{loc} is obtained using interpolation. Afterwards, the estimated time delay of the pulse τ_e is obtained using $\tau_e = n_{pk}T_s + dt_{loc}$. The steps for the MF-LS time delay estimator are shown in Algorithm 5.2. Larger H leads to better results with lower bias generally, since more potential outcomes are evaluated. This algorithm, as will be shown below, reduces the biases for two-tone ranging, as it evaluates multiple outcomes with varying relative times for the discretization points compared to the received ranging pulses. However, MF-LS has a much higher computational complexity compared to both interpolation and NL-LS. Nevertheless, if biases are observed in the estimates for a given ranging waveforms, using a peak estimator with a higher computational time to obtain much more accurate results is a desirable option.

5.2.1.2 Comparison Between the Peak Estimators

The three presented peak estimators were compared using simulation and experiment for ranging using two-tone waveforms. The experimental setup is shown in Fig. 5.9. Simulations were carried out using MATLAB, and the measurements were conducted using an Ettus Universal Software

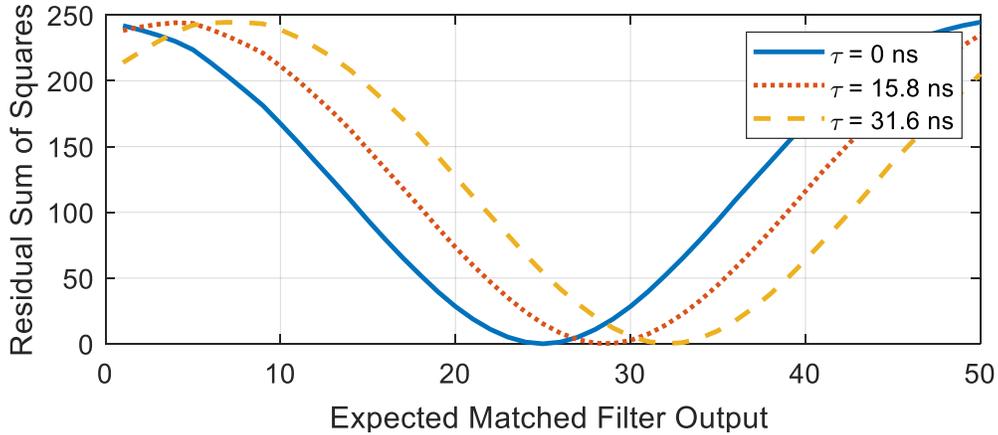


Figure 5.8: RSS of the normalized matched filter outputs for three time delay cases for a ranging waveform with $f_1 = 20$ kHz, $f_2 = 4.5$ MHz, $f_s = 10$ MHz, and $T = 99.8 \mu\text{s}$. The three selected time delay cases reside within the same two discretization points. Every expected matched filter output h corresponds to a predicted time shift dt_h .

Radio Peripheral (USRP) X310 SDR with a UBX-160 daughterboard, which was controlled using LabVIEW 2018. Instead of using antennas for ranging, cables were connected between the transmitter and receiver in order to have a controlled environment where the ground truth can be extracted accurately. The time delay was controlled by adding or removing 3.048 m coaxial cables. Simulation and experimental results are shown in Figs. 5.10, 5.11, and 5.12. In the simulations and experiments we used $f_1 = 20$ kHz, $f_s = 10$ MHz and f_2 was varied from 60 kHz to 4.94 MHz. The pulse width was set to $99.8 \mu\text{s}$ and the carrier frequency was 2.8 GHz. The results were obtained by averaging 100 estimates. The errors that are observed for each algorithm in Figs. 5.10, 5.11, and 5.12 are clearly shown in Fig. 5.13. As can be seen, the highest bias was obtained from the interpolation approach, regardless of the bandwidth of the received waveform. As for NL-LS, lower biases were observed, however, once the bandwidth of the pulse started increasing, the bias increased considerably. As for the MF-LS approach, much lower bias values were observed aside from a small number of special cases where only bandwidths that very small or very close to the Nyquist frequency (5 MHz) showed considerable biases.

A clear comparison between the RMSE of the estimates of each method is shown in Table 5.1. Two bandwidth intervals were compared; the first interval contains all the considered bandwidths,

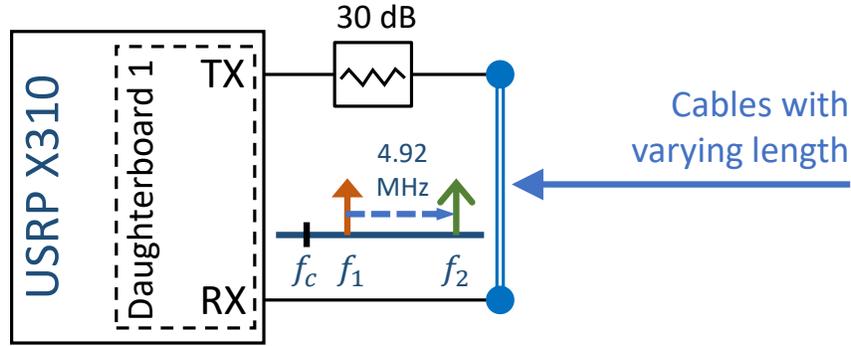


Figure 5.9: Experimental SDR setup. Extra cables were added to modify the received signal time delay.

which are from 40 kHz to 4.92 MHz, whereas the second interval contains the bandwidths from 2 MHz to 4.5 MHz, representing a more relevant bandwidth for ranging signals. The values in Table 5.1 demonstrate that the MF-LS algorithm reduces errors by nearly an order of magnitude for 2 MHz to 4.5 MHz bandwidths compared to the other considered algorithms. Although interpolation and NL-LS showed higher biases, they can still be used for two-tone ranging, especially if the total considered displacement distance is small compared to the rate of change for the biases (few centimeters for example), in this case the initial bias can be calibrated out.

The standard deviation of every waveform and algorithm along with the CRLB are shown in Fig. 5.14. The CRLB curve common for all the peak estimators as it is only dependent on the ranging waveform and the SNR . Every 100 samples, the standard deviation was calculated for the same cable length, but for varying bandwidths. The CLRb increases for every 100 samples, as for longer cable lengths, the SNR drops. It can be seen that the standard deviations of all the estimators were similar. The increased standard deviation of MF-LS compared to the other algorithms around the regions of high bandwidths was due to the biases that were observed by interpolation and NL-LS which caused the standard deviation to drop below the CRLB, which is a metric used to evaluate unbiased estimators.

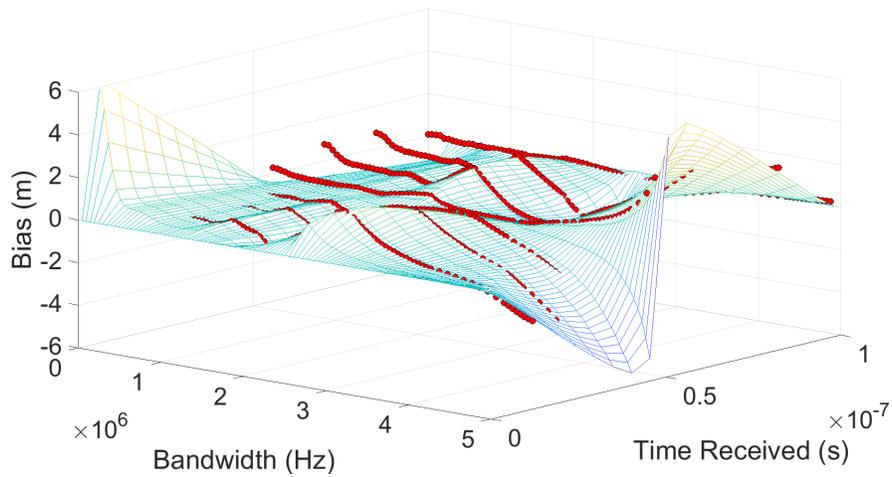


Figure 5.10: Simulated and experimental results showing a comparison between the actual distances traveled by the signals and their estimates. The surface shows the simulated results, and the red dots show the experimental results of interpolation. From [11] © 2021 IEEE.

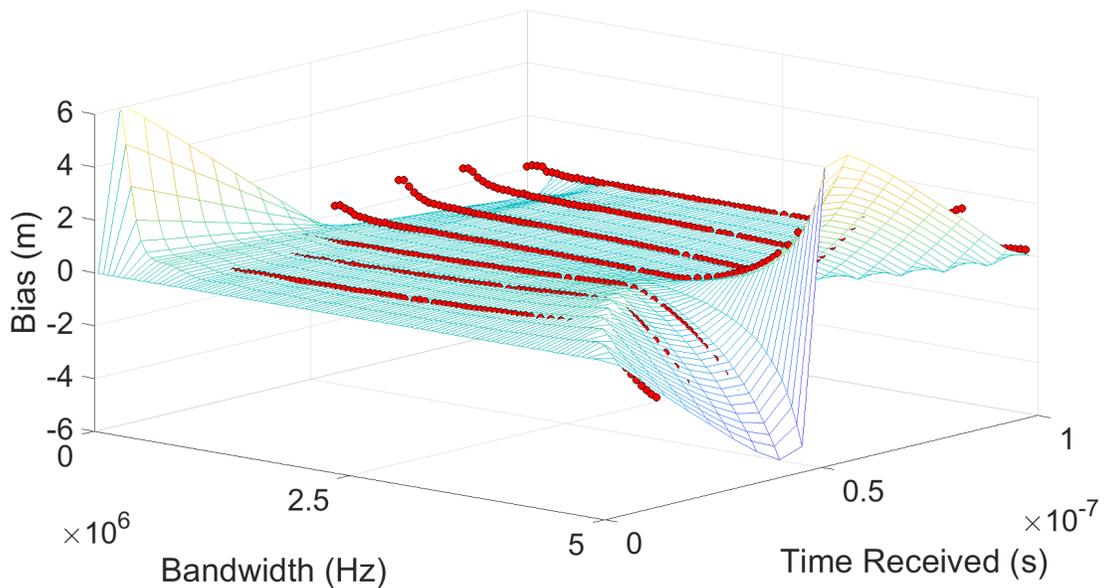


Figure 5.11: Simulated and experimental results showing a comparison between the actual distances traveled by the signals and their estimates. The surface shows the simulated results, and the red dots show the experimental results of NL-LS. From [11] © 2021 IEEE.

5.2.2 Scalability of Two-Tone Waveforms

Two-tone ranging waveforms offer optimal performance, however they are not scalable in nature.

In [120], two-tone stepped-frequency waveform (TTSFW), a modified version of the two-tone

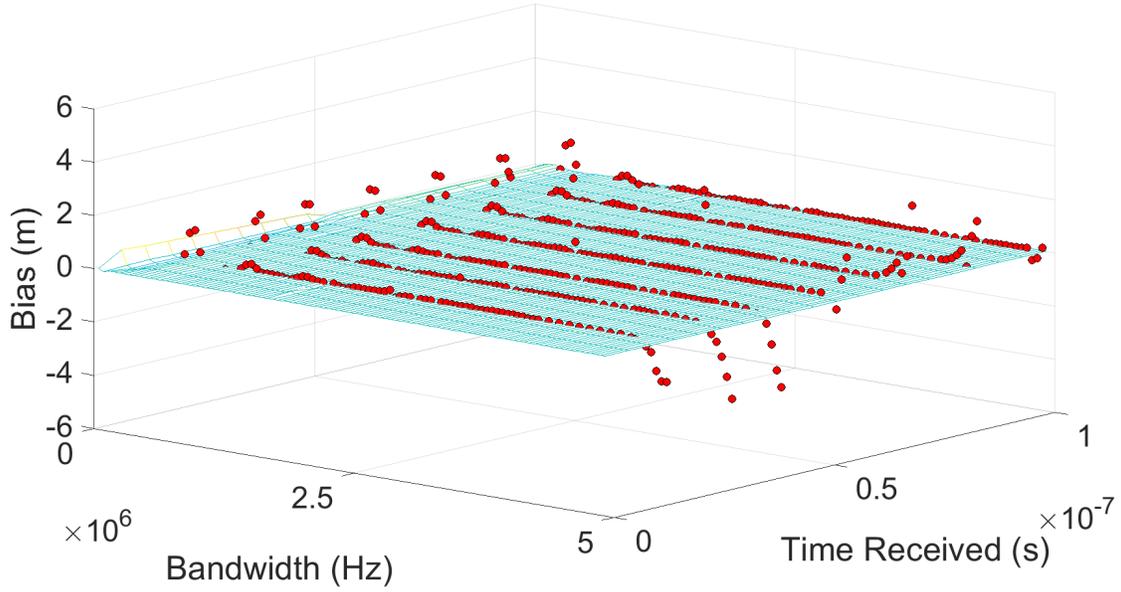


Figure 5.12: Simulated and experimental results showing a comparison between the actual distances traveled by the signals and their estimates. The surface shows the simulated results, and the red dots show the experimental results of MF-LS. From [11] © 2021 IEEE.

Table 5.1: Ranging RMSE for the three ranging algorithms.

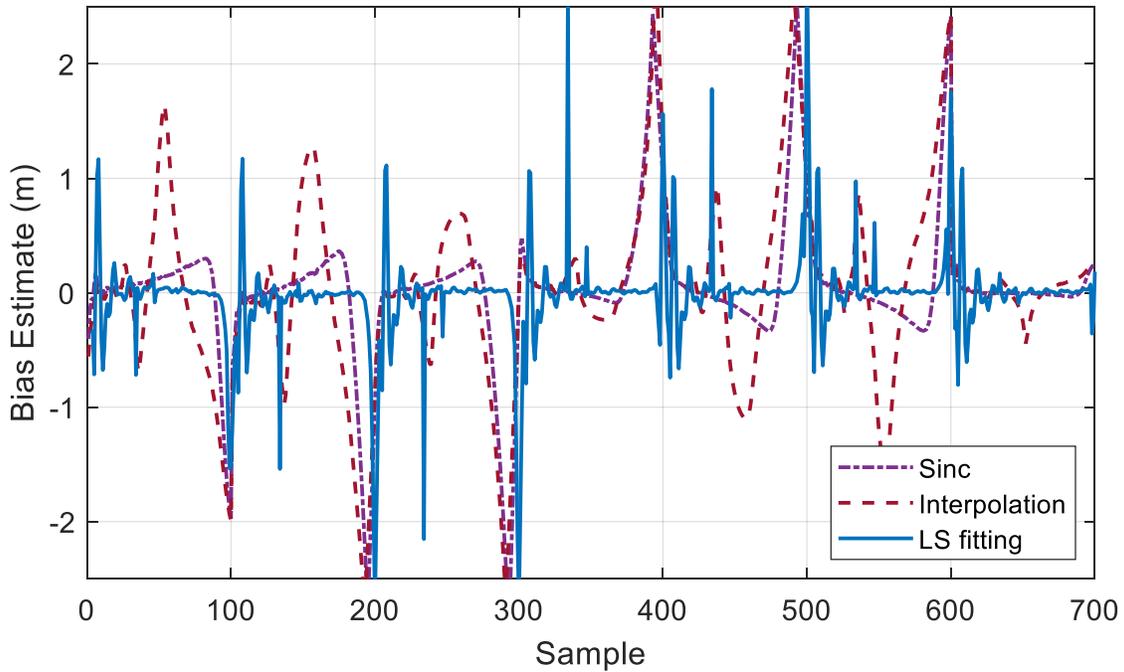
| Bandwidth | RMSE for interpolation | RMSE for NL-LS | RMSE for MF-LS |
|--------------------|------------------------|----------------|----------------|
| 40 kHz to 4.92 MHz | 0.688 m | 0.482 m | 0.407 m |
| 2 MHz to 4.5 MHz | 0.78 m | 0.421 m | 0.056 m |

waveform that combines accuracy with scalability, was introduced. TTFSW is formed by multiple two-tone encoding pulses where each pulse has a predetermined frequency step. By varying the frequencies of the two-tones and including multiple pulses, TTFSW offers the desired scalability. TTFSW is expressed as

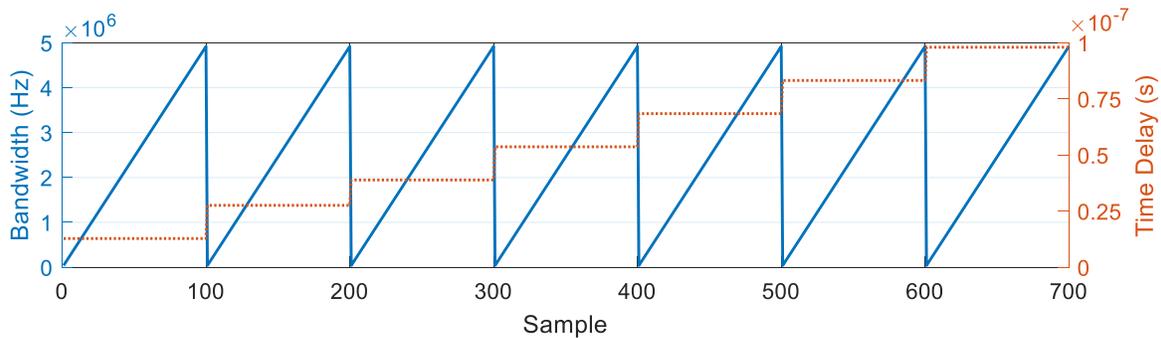
$$s_{TTFSW}(t) = \frac{1}{N_p} \sum_{n=0}^{N_p-1} \text{rect}\left(\frac{t - nT_r}{T}\right) (e^{j2\pi f_1 t} + e^{j2\pi f_2 t}) e^{j2\pi n \delta f_p t} \quad (5.51)$$

where N_p is the number of encoding pulses, T_r is the nonzero portion of the pulse duty cycle, and δf_p is the step frequency expressed as

$$\psi f = \frac{BW}{2N_p - 1} \quad (5.52)$$



(a)



(b)

Figure 5.13: (a) The errors from Figs. 5.10, 5.11, and 5.12 for the various peak estimation algorithms for different cable lengths and bandwidths (average of 100 estimates). (b) Time delay based on the cable length (dotted orange line) and signal bandwidth (solid blue line) per sample.

where BW is the total waveform bandwidth, not to be confused with the bandwidth of every separate encoding pulse. The first tone is usually set to a low value, and the second tone is obtained from $f_2 = f_1 + \Delta f_p$, where $\Delta f_p = N_p \delta f_p$. A total of N_p encoding pulses allow TTFSW to service $N_p!$ connections. A time domain and time-frequency spectrum representation of the TTFSW is shown in Figs. 5.15 and 5.16 respectively.

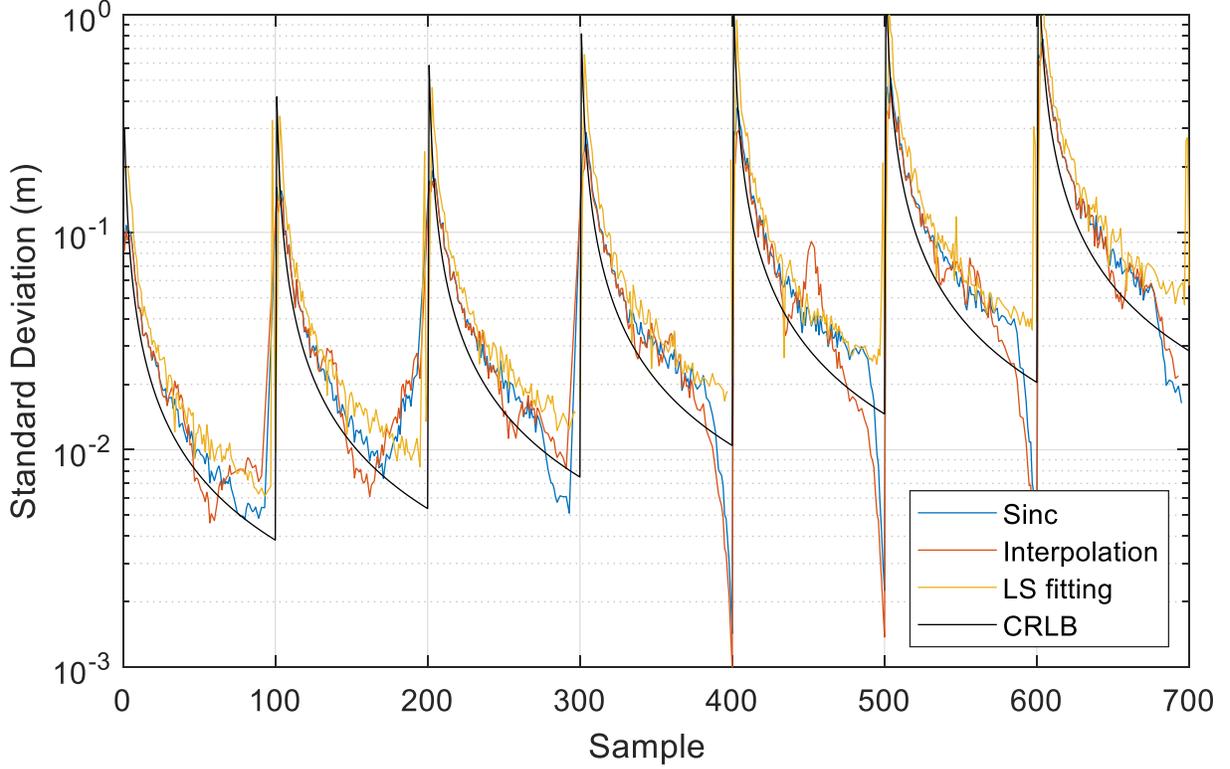


Figure 5.14: The standard deviation for each ranging method along with the CRLB. The SNR for the first 100 samples was 32 dB and then it decreased by around 2.9 dB for every added cable (every 100 samples), reaching an SNR of 14.6 dB. For some samples, the standard deviation of both the interpolation and NL-LS is smaller than the CRLB, this is due to the bias in the estimates. From [11] © 2021 IEEE.

The CRLB of TTFSW is computed from

$$\text{var}(\hat{\tau} - \tau) \geq \frac{1}{\frac{\zeta_f^2}{E} \text{SNR}}, \quad (5.53)$$

Assuming that the waveform is symmetric in the spectrum domain. $\frac{\zeta_f^2}{E}$ is computed from

$$\frac{\zeta_f^2}{E} = \pi^2 \left(\frac{BW}{2 - \frac{1}{N}} \right)^2 + \frac{(2\pi BW)^2}{N(4N^2 + 4N + 1)} \sum_{n=0}^{N-1} n^2. \quad (5.54)$$

In addition to the scalability of TTFSW, this waveform reduces the ambiguities as observed in Fig. 5.17. Further details about TTFSW are available in [120].

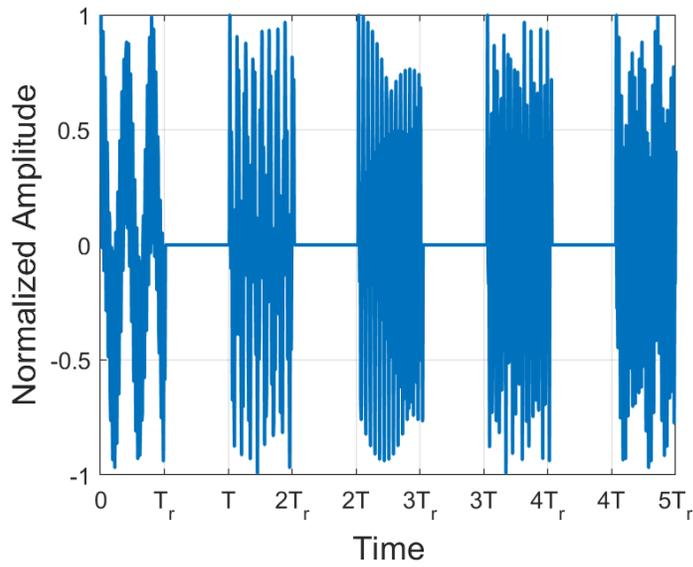


Figure 5.15: Time domain of a TTSFW with $N_p = 5$ and 50% duty cycle. From [7] © 2021 IEEE.

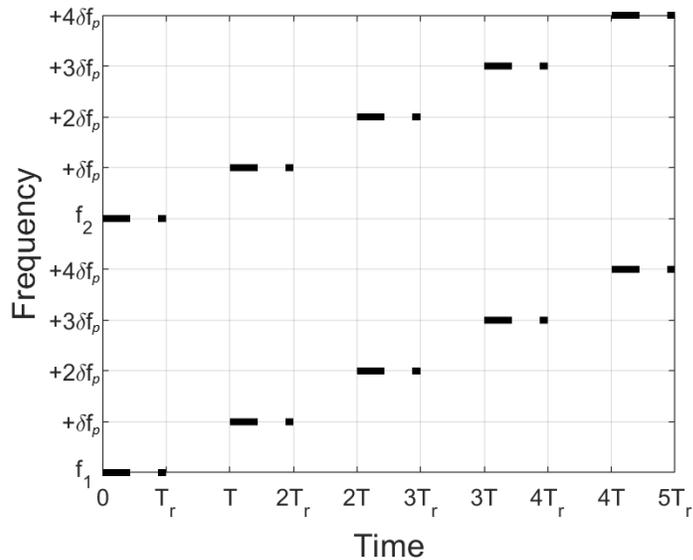


Figure 5.16: Time-frequency spectrum of a TTSFW with $N_p = 5$ and 50% duty cycle. From [7] © 2021 IEEE.

5.3 Adaptive Ranging

Accurate time delay-estimation is needed for open-loop coherent arrays, nevertheless, the requirements differ depending on the application. If the nodes are moving in one dimension and the frequency synchronization is not achieved wirelessly, it was shown that an average standard devia-

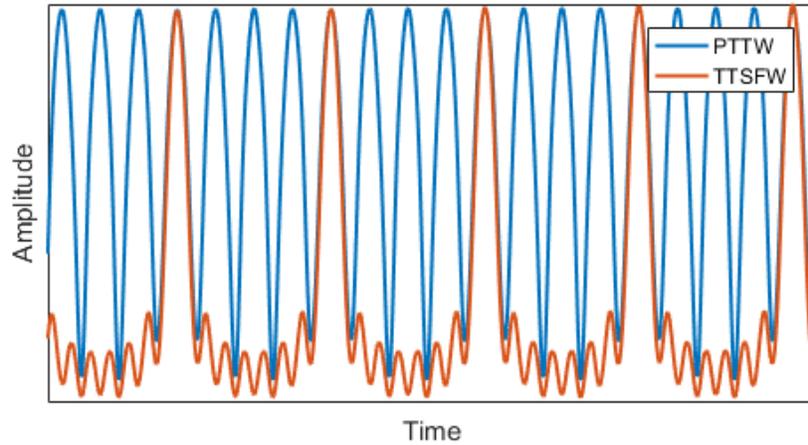


Figure 5.17: Matched filter output for TTSFW with $N_p = 4$ compared to the matched filter output of pulsed two-tone waveforms (PTTW). From [7] © 2021 IEEE.

tion of $\lambda_c/15$ for range estimates is sufficient to achieve 0.9 coherent gain with a high probability. However, different metrics are needed for the cases where frequency synchronization is achieved wirelessly, which is the usual case. In such architectures, a ranging standard deviation of at least $\lambda_c/27$ is desirable when a coherent gain of 0.9 is targeted. Further, much more accurate estimates are needed when the nodes are moving in two dimensions, and the metrics are dependent on the employed localization approach and on the spread of the nodes.

Thus, the desired time-delay estimation or ranging accuracy constantly changing and the parameters of the ranging waveform need to also be changing to overcome the variations of the SNR. For a given pulse width, one can select the highest possible bandwidth to maximize the time-delay estimation accuracy, ensuring that the desired requirements for accurate phase alignment are met. Nevertheless, using very high bandwidth introduces interference for other communications or sensing systems operating on neighboring bands and leads to an overly-crowded radio spectrum.

Generally, resource allocation is always a concern in any system, especially for RF systems; resources can be in the form of power [121, 122], time [123], and bandwidth [10, 12, 124, 125], among others. For instance power is an issue when there is a constraint on the power budget, limiting the localization antennas and the transmitted power [121]. There are also time constraints on the ranging pulse width, where a ranging rate of at least 200 Hz is desired for nodes with strong

vibration profiles, such nodes can represent transmitters on drones or helicopters [123]. Spectral footprint or radio spectrum crowdedness was addressed differently depending on the scenario, for instance, cognitive radio networks transmitting OFDMA signals maintained the same performance in [124, 125] while decreasing the used bandwidth at the expense of higher transmitted power. On the other hand, some systems focus on maximizing the bandwidth of the transmitted signals while minimizing the interference [126, 127]. Other cognitive systems, once greatly affected by interference, relocate their signals to other desirable bands [128, 129].

This section focuses on minimizing the bandwidth of the ranging signals to reduce the interference that can be observed on neighboring bands. The bandwidth is controlled such that the coherent gain, if impacted, is gracefully degraded. A framework ensuring that a desired ranging accuracy is obtained while minimizing the spectral footprint is developed. Such approach allows the adaptive ranging system to adapt to changing weather conditions by achieving the desired ranging accuracy while maintaining a minimal bandwidth.

5.3.1 Using the Standard Deviation of Multiple Range Estimates as Feedback

5.3.1.1 Adaptive Ranging Framework

This section focuses on adaptive ranging approaches that implement a perception/action framework such as in [130, 131] to adjust the bandwidth of the transmitted waveforms, enabling the adaptive ranging system to overcome weather changes while minimizing the spectral footprint. This framework is generally used in the literature for clutter rejection (e.g. [132, 133]), and target recognition and tracking in challenging conditions (e.g. [134, 135]), nevertheless it has not previously been used for cooperative inter-node ranging.

As mentioned earlier, (5.16) allows us to determine the minimum achievable variance for a non-biased time delay estimator. Thus, knowing the SNR and waveform parameters is enough to determine the CRLB. However, in practice it is not possible to attain the CRLB due to spurious signals, quantization errors, and hardware-induced nonlinearities. Also, the SNR is constantly

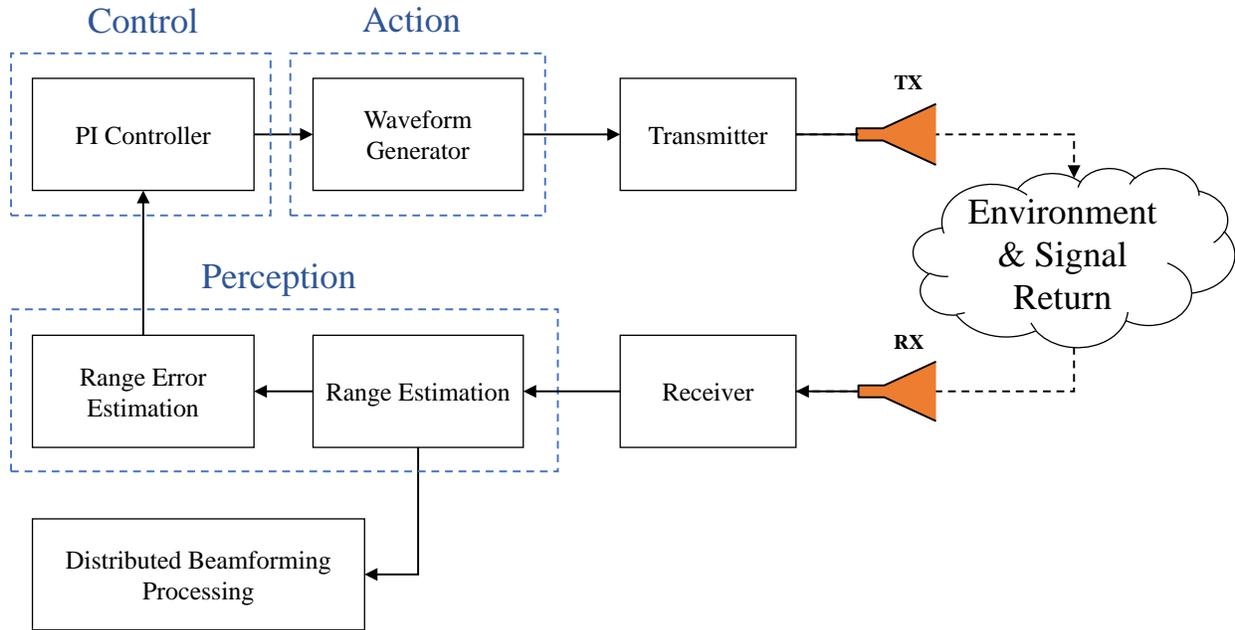


Figure 5.18: Adaptive inter-node ranging architecture that relies on the standard deviation of the range estimates. From [10] © 2021 IEEE.

changing in practice, due to the array dynamics, interference, and antenna gain, among others. All these changes to the channels make it impossible to select a constant bandwidth for the ranging waveform that minimizes the spectral footprint and achieve the desired accuracies. This section focuses on two-tone signals due to all the benefits that were mentioned earlier in this chapter. To overcome all the dynamic changes, the tone separation is modified continuously to prevent any considerable drop in accuracy and to minimize the allocated bandwidth. It is important to note that even though two-tone waveforms are spectrally sparse in nature, and other signals can be transmitted between the two-tones, this section considers the effects on systems that cannot perform frequency hopping, which for this case, consuming larger bandwidths leads to overlapping frequencies and consequently reduction in performance especially if the affected signals are communications signals (since they tend to have lower power than sensing signals).

As depicted in Fig. 5.18, the adaptive ranging approach is based on a perception/action framework. The transmitter (TX) and receiver (RX) are placed on the node that is estimating the range of its target, the node can be designated as either primary or secondary. The signal returns con-

stitute the reflected pulses from the target (primary or secondary). Based on the framework that was introduced in Chapters 1 and 2, it can be assumed that this adaptive ranging system is placed on a secondary node and the targeted node is the primary. Also, the target can be represented by an active repeater, or a corner reflector, which represents the case where no amplification is done to the transmitted ranging pulses. At first the radar transmits a ranging pulse with a desired initial bandwidth. The reflected pulses from the target are then received. The received returns can have the same carrier as the transmitted pulses or they can have another carrier in the case where a repeater is used. These returns (one or multiple returns) are then processed in the perception step where certain properties such as standard deviation are extracted. Range estimation is also performed in the perception step, which is used at a later stage for phase alignment among the nodes in the array. Afterwards, the extracted information is fed to a control step that determines the desired waveform modifications. These desired modifications are carried to the action step, where the transmitted waveform is adjusted based on the desired action. Finally, a new set of ranging pulses with the modified characteristics are retransmitted. This cycle runs continuously to achieve the desired ranging performance with minimal bandwidth. The three adaptive ranging processes operate as follows:

Perception Process The perception process focuses on extracting the information of interest from the received signals. These information consist of both identifying the range of the target, and determining the desired metrics that are needed for the decision making in the control process; for the proposed approach in this section, the needed metric is the standard deviation of the range estimates. The standard deviation is calculated using

$$\sigma_R = \sqrt{\frac{\sum(R_i - \bar{R})^2}{I}}, \quad (5.55)$$

where R_i is the i th range of a batch of I range estimates, and \bar{R} is the mean of the range estimates in that batch. In this section, A processing interval refers to the process of capturing, determining, and processing I range estimates, than evaluating both the control and action processes.

The ranging is achieved as described in Section 5.2 using two-tone waveforms with varying bandwidth. Interpolation was used for peak estimation. Although in many cases the biases obtained from interpolation are beyond the acceptable error range, this is not an impediment since the same adaptive ranging approach can be implemented by either modifying the transmitted waveform or the peak estimator.

Control Process In this step, the calculated standard deviation is compared to a reference desired value, and the difference is fed to a controller that in turn modifies the bandwidth of the transmitted waveform in order to get closer to the desired ranging accuracy. In this work, the closed-loop Proportional-Integral (PI) controller, which is derived from the PID controller is used. This widely used controller has a relatively simple architecture is generally used for robotics application. PID controllers have also their share in sensing systems, as it has been used for multiple applications such as imaging [136], and radar missile guidance [137, 138]. PID controllers output a correction signals based on the observed error (difference between the controller input and the reference), the correction is a combination of the proportional, integral, and derivative actions. The general discrete form of a PID controller is given by

$$x[n] = x[n - 1] + K_p \left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t} \right) e[n] - \left(1 + \frac{2T_d}{\Delta t} \right) e[n - 1] + \frac{T_d}{\Delta t} e[n - 2] \right] \quad (5.56)$$

where $x[n] = \frac{1}{2\delta f}$ is the output of the controller, which in this work is inversely proportional to the frequency separation of the tones in the pulsed two-tone waveform. This choice was made since the standard deviation of the range measurements is a function of $\frac{1}{2\delta f}$ as shown in (5.16). The error term $e[n]$ represents the difference between the measured time delay standard deviation and the reference value, in the next paragraphs. The proportional gain is represented by K_p , $\Delta t = t_n - t_{n-1}$ is the time interval, T_i is the integration time, and T_d is the derivative time. A simplified representation of the controller is shown in Fig. 5.19.

The proportional action in the PID controller is tuned by the proportional gain K_p , which maps the input error to a desired output using a linear fashion. An instantaneous large error leads to an instantaneous high action on the output, and vice versa. On the other hand, the integral action,

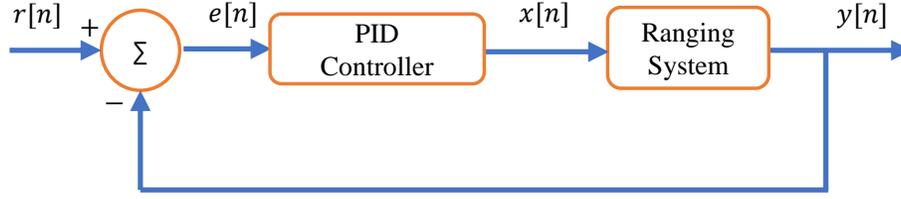


Figure 5.19: A simplified representation of the used PID controller. In this figure, $y[n]$ represents the system output, which is the standard deviation of the time-delay estimates of multiple consecutive ranging pulses.

which is controlled by the integration time T_i removes any steady state error that would be observed in the case where only a Proportional (P) controller was used. As it can be inferred, the integral action has an exponential behavior, the longer the error lingers, the higher is the integral action. Thus for high T_i values, the system will have overshoots, especially if the observed errors has a tendency to last for a long duration. Finally, the derivative action is lined to the derivative time T_d . The derivative action monitors and damps the integral action to reduce the overshoots. The derivative action counteracts any abrupt action from the other two parts of the controller, ensuring a smooth operation. Nevertheless, in unstable environment, the derivative action tends to add to the error instead of minimizing it. This behavior can drive the system to instability, and in many cases, such as in this dissertation, the derivative action can be disregarded.

Every part of the controller has a specific action and all parts are important for different situations. The proportional controller is always required for an appropriate response. Similarly, the integral action is needed to remove any steady state error. However, the derivative action can be disregarded in this work to ensure stability in harsh environments, such as rain. The derivative action can be removed by simply setting T_d to zero in (5.56). The PI controller can be expressed as

$$x[n] = x[n - 1] + K_p \left[\left(1 + \frac{\Delta t}{T_i} \right) e[n] - e[n - 1] \right] \quad (5.57)$$

As explained, PID controllers have three gain parameters (K_p , T_i , and T_d) that determine the behavior of the controller. Determining the correct values for these parameters is crucial for a smooth and stable operation of the controller. Tuning PID controllers can be done through various

methods as explained in [139], and these methods are generally classified in three groups:

- Tuning using plant model: in this case, an accurate representative model of the plant being controlled is needed. The model is usually expressed in terms of one or multiple transfer functions. The controller parameters are chosen by examining the closed-loop poles.
- Tuning using approximate model: prior to selecting the closed-loop behavior, a model describing the plant needs to be approximated. There are multiple techniques for doing so, as estimating a model using a first order model with delay. Also, the model could be inferred by collecting pair of inputs and outputs, and then the model can be estimated using a desired system identification technique. However, inaccurate representation of the model is common for such approaches since in many cases not all the system behaviors are described in the collected input/output data.
- Online tuning: This method is based on tuning the controller parameter online without relying on models. Prior tuning, there should be a knowledge on the behavior of the plant and its limits.

For the presented application, the ranging system cannot be described using accurate transfer functions, which makes the online tuning methods a logical candidate. The Ziegler-Nichols PID tuning method [67] is commonly used for online tuning. During the tuning phase, only the proportional gain K_p is modified while both T_i and T_d are set to zero throughout the tuning process. K_p is first set to zero, afterwards, small increments are added until it reaches an optimal value K_u , that makes the system critically stable. During this operation, the output should show oscillations with constant frequency and magnitude. At this point, the oscillation period T_u is obtained and the controller gains are selected. For the case of PI controllers, the weights are selected using

$$K_p = 0.45K_u, \quad (5.58)$$

$$T_i = \frac{T_u}{1.2}, \quad (5.59)$$

$$T_d = 0. \quad (5.60)$$

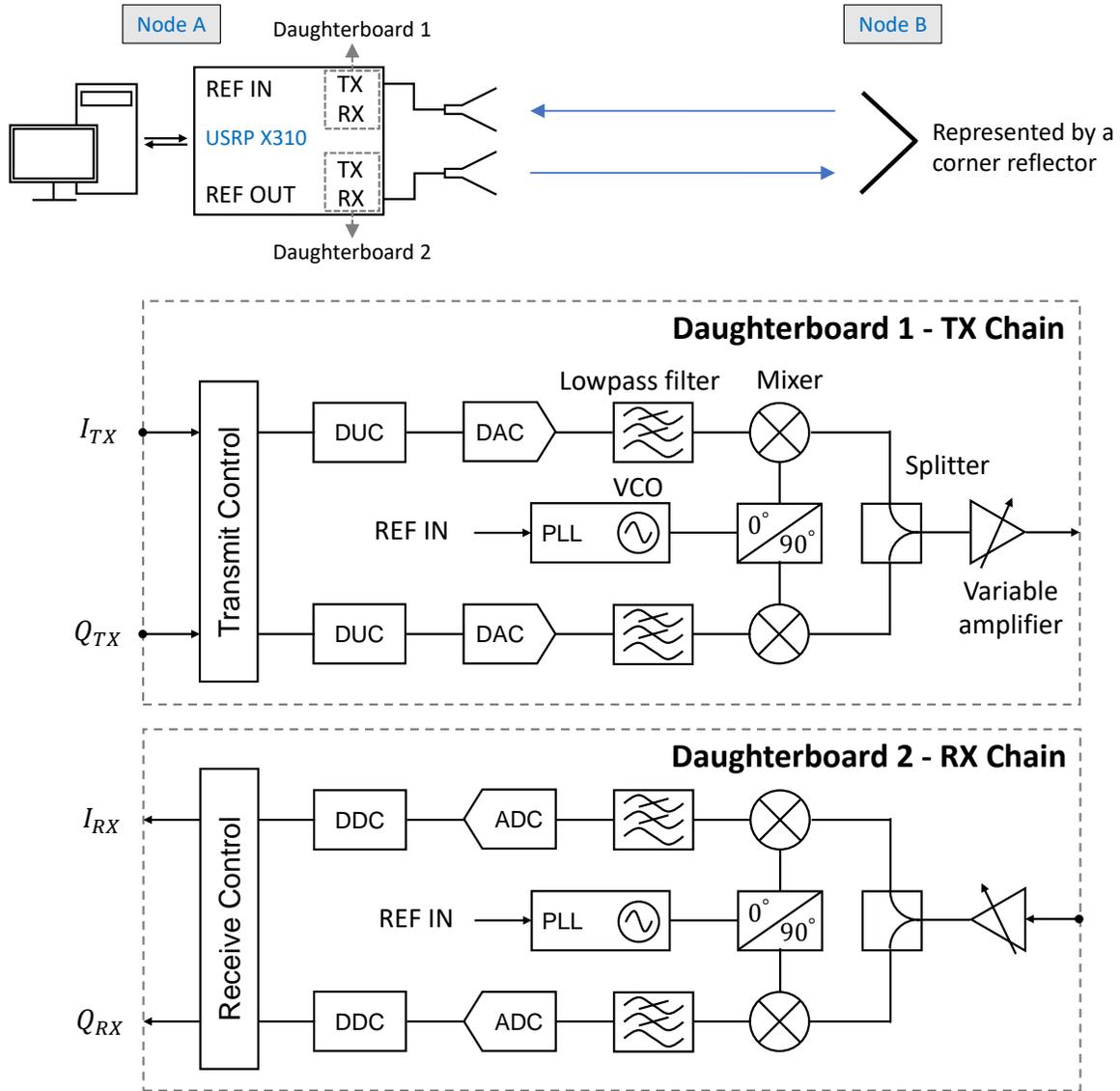


Figure 5.20: Block diagram representing the hardware used for the indoor adaptive ranging experiment. From [12] © 2021 IEEE.

Action Process In this last step, the controller output, which dictates the new desired tone separation through $x[n] = \frac{1}{2\delta f}$, is used to design the new ranging waveform. In this step the new tone f_2 is calculated and the new ranging waveform is generated in real time for continuous operation of the radar. One SDR is sufficient for this task since it has the capabilities to generate the new waveform, transmit it, receive it back, and process it, all in real time. The block diagram of the used hardware is shown in Fig. 5.20.



Figure 5.21: Experimental setup for adaptive ranging system measurements.

5.3.1.2 Adaptive Indoor Ranging Experiment

Experimental results were gathered using a USRP X310 SDR with two UBX 160 daughterboards. The carrier frequency was set to 4.2 GHz. The transmitting and receiving daughterboards were connected to two horn antennas with an operating range of 3.85 GHz - 5.85 GHz and a gain of 15 dBi each. The maximum allowed tone separation was set to 3 MHz with a sampling frequency equal to 10 MHz and pulse width equal to 1.2 ms. The pulse repetition interval (PRI) was set to 2 ms which is long enough for most of the applications. The transmitted pulses were reflected using a corner reflector placed 4.5 m away from the antennas. The preprocessing SNR (prior to the matched filtering) and standard deviation were calculated in windows of 40 pulses. Fig. 5.21 shows the experimental setup.

As described earlier, the Ziegler-Nichols tuning method was applied and Fig. 5.22 shows the ranging system response to the gain K_v . In this tuning process the reference standard deviation of time estimates was set to 7.5×10^{-3} m; this value was selected since it is close to the maximum achievable frequency, in order to allow the PI to have appropriate response. The sampling frequency was set to 10 MHz and the maximum tone separation was limited by the sampling

© 2021 IEEE. Section 5.3.1.2 is reprinted with minor modifications, with permission, from Section V of "S. R. Mghabghab, and J. A. Nanzer "Adaptive Internode Ranging for Coherent Distributed Antenna Arrays," IEEE Transactions on Aerospace and Electronic Systems, vol. 56, no. 6, pp. 4689-4697, 2020".

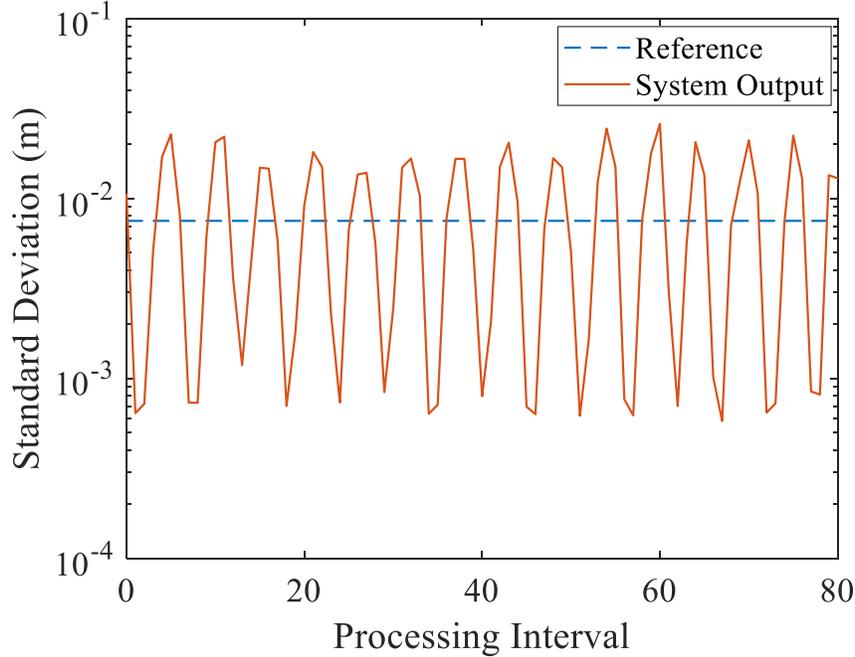


Figure 5.22: Ranging system behavior to the proportional gain K_u .

time. This proportional gain was achieved for $K_p = K_u = 50000$ with an oscillation period of $T_u = 6.125$ processing intervals. A processing interval is equal to the duration of the 40 received pulses (1.2 ms each) plus the required processing time; the total processing interval was equal to 2 s. This time latency was due in part to data transfers between the SDR and the host computer running the adaptive processor, and was not optimized for this work. Once K_u and T_u were obtained, the PI controller gains were set to $K_p = 22500$ and $T_i = 5.1$ processing intervals. If $e[n]$ was a function of the ranging standard deviation instead of the standard deviation of the time-delay estimates, different values for K_u and T_u would have been obtained.

Two tests were conducted to evaluate the adaptive ranging approach; where the standard deviation of the range estimates was evaluated for various SNR conditions. In the first test, the adaptively controlled waveform was compared to two static waveforms with tone separations of 5 kHz and 3 MHz (the minimum and maximum allowed tone separations), as shown in Fig. 5.23. At first, the SNR of the received pulses is at around 50 dB, afterwards, the transmitted power was reduced every 50 processing intervals. The SNR was estimated by first measuring the noise power when no pulses were transmitted, then, the received pulse power was measured, and at the end, the

SNR was obtained by calculating the ratio. For the adaptive ranging system, it is expected that the tone separation will increase in response to the decreasing SNR. At first, with high SNR, the tone separation is close to 5 kHz and with decrease in the received power, the bandwidth increases until reaching the maximum tone separation of 3 MHz. As for the cases where the tone separation is kept constant, when the tone separation is 5 kHz, the accuracy is close to the reference but continually degrades as the SNR decreases. For the 3 MHz tone separation waveform, the accuracy is better than necessary until the 200th processing interval, when the SNR degrades too far. At points prior to the 200th processing interval, the waveform uses far more bandwidth than needed for the desired accuracy, needlessly occupying spectrum. The adaptively controlled waveform starts at a narrow tone separation that gradually increases as the SNR degrades. The resulting standard deviation closely follows the reference value until the 200th processing interval, when the SNR degrades so far that the necessary tone separation is beyond the 3 MHz limit. The adaptive ranging system thus achieved the desired ranging accuracy without occupying unnecessary spectrum.

In the second experiment, the transmitted power was set to a constant value with an SNR of 37 dB for the received signal, and the reference point was modified to show how the adaptive ranging system behaves under these conditions. Fig. 5.24 shows the accuracy of the adaptive ranging system in following the desired reference standard deviation along with the modifications applied on the tone separation to achieve this adaptive task. With even the specified minimal tone separation of 3 MHz, the system was able to adapt to a reference standard deviation of 3 mm; the minimum achievable standard deviation was 1.5 mm when the reference was set to 0.3 mm, which dictated a tone separation beyond the 3 MHz limit.

The received signal in Fig. 5.24 has relatively high SNR values when considering non-cooperative ranging systems (around 37 dB), however it is important to note that since in the proposed system architecture an active repeater is used on every node to amplify and transmit back the received signal to its source, only one-way free-space path losses (FSPL) are incurred instead of two-ways. Considering that the active repeater is retransmitting the pulse at the same initial power, the use of active repeaters helps keeping the operation in the moderate to high SNR region, where the

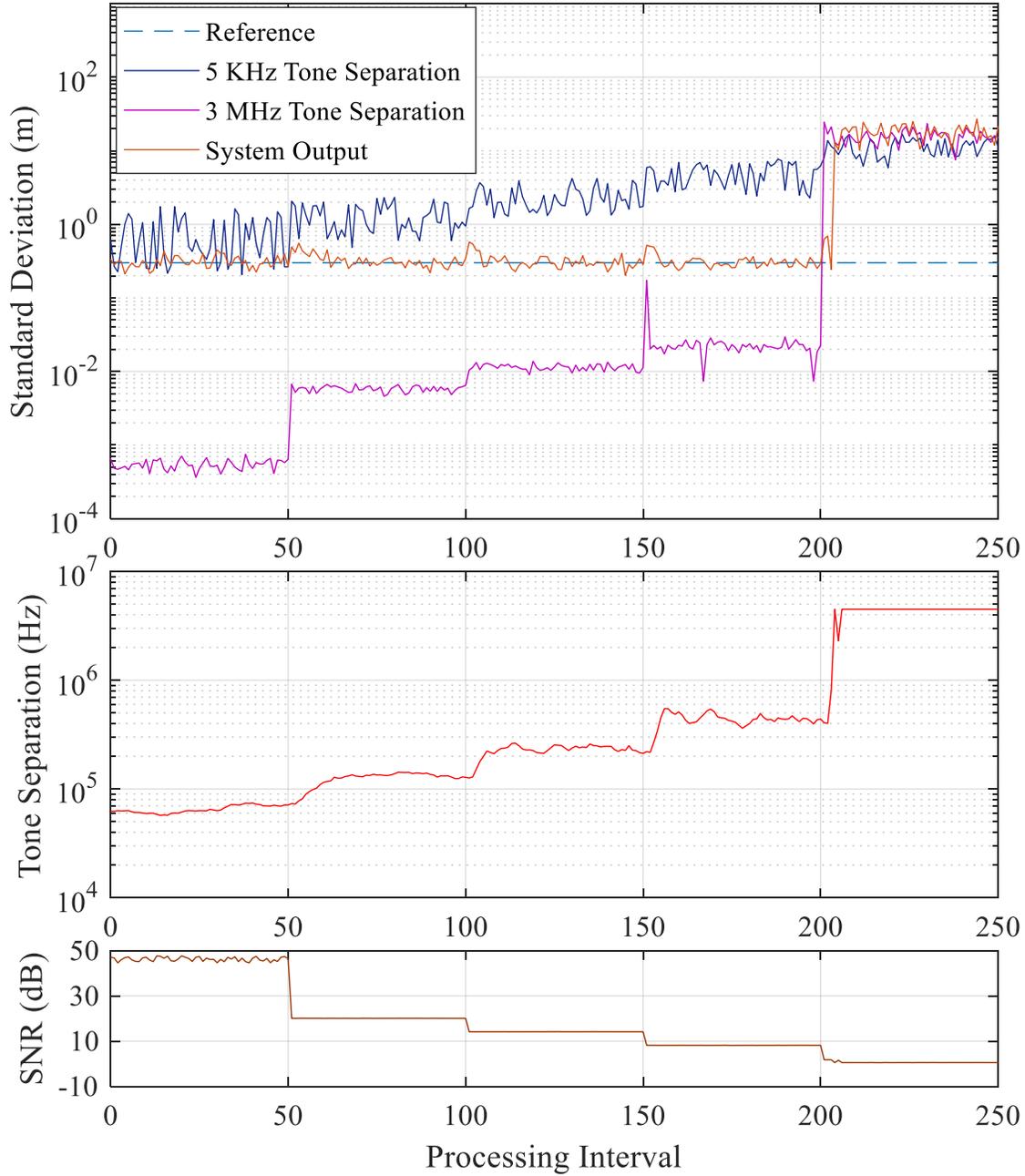


Figure 5.23: The behavior of the adaptive ranging system for a varying SNR.

accuracy of ranging system is dependent on the tone separation. The FSPL for isotropic antennas can be found from the Friis transmission formula, from which is obtained

$$\text{FSPL} = \left(\frac{4\pi Rf}{c} \right)^2, \quad (5.61)$$

with f representing the transmitted carrier frequency of the ranging signals.

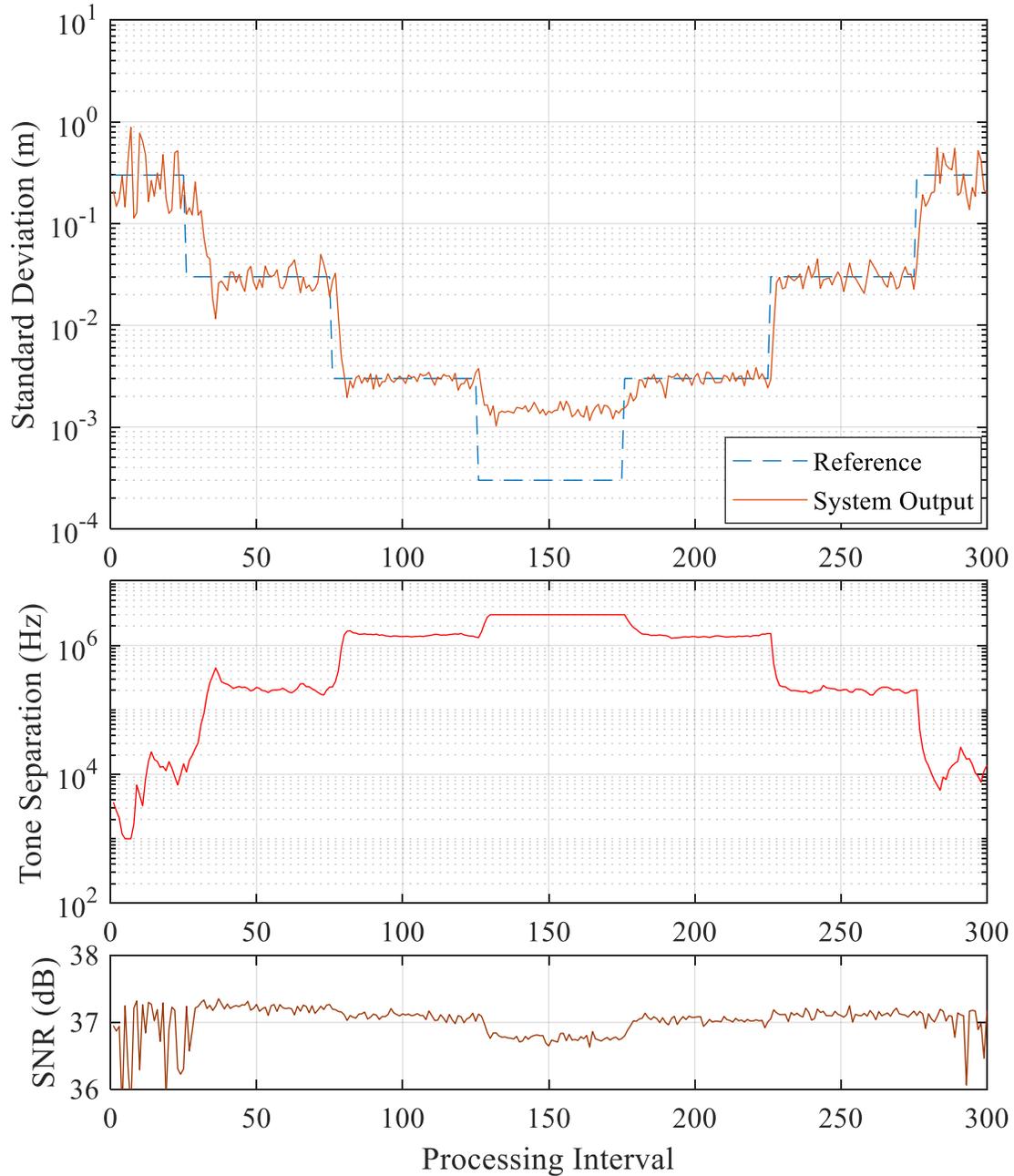


Figure 5.24: The behavior of the adaptive ranging system for a varying reference point.

5.3.1.3 Adaptive Outdoor Ranging Experiment

In this section, the adaptive ranging framework is explored outdoors over a 90 m link. The corner reflector that was used for the indoor testing was replaced by an active repeater to generate a similar behavior to what can be expected in an operational open-loop coherent distributed array. In addition, wireless frequency synchronization was implemented between the radar and repeater. A

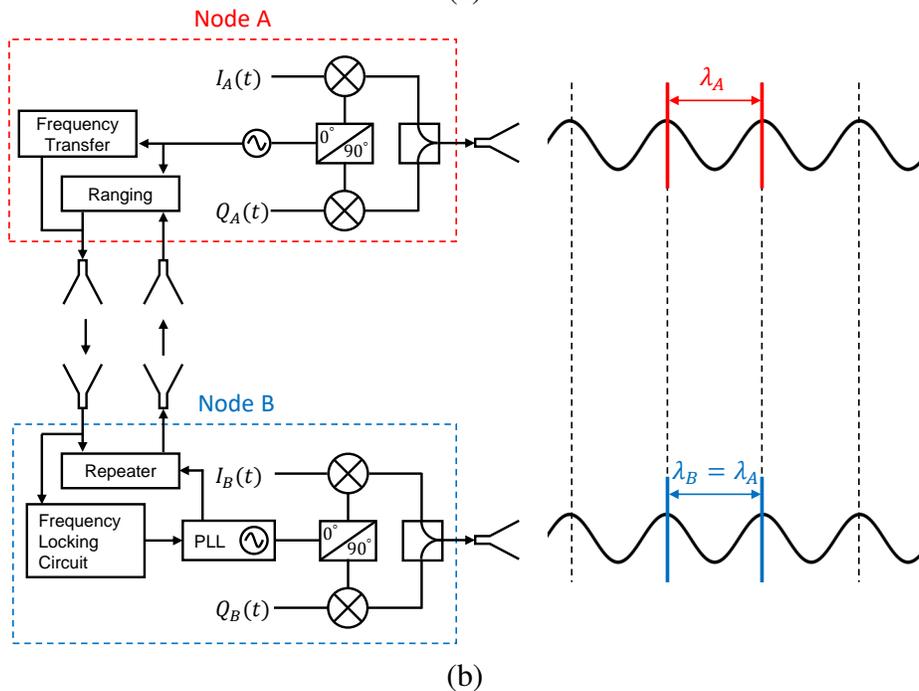
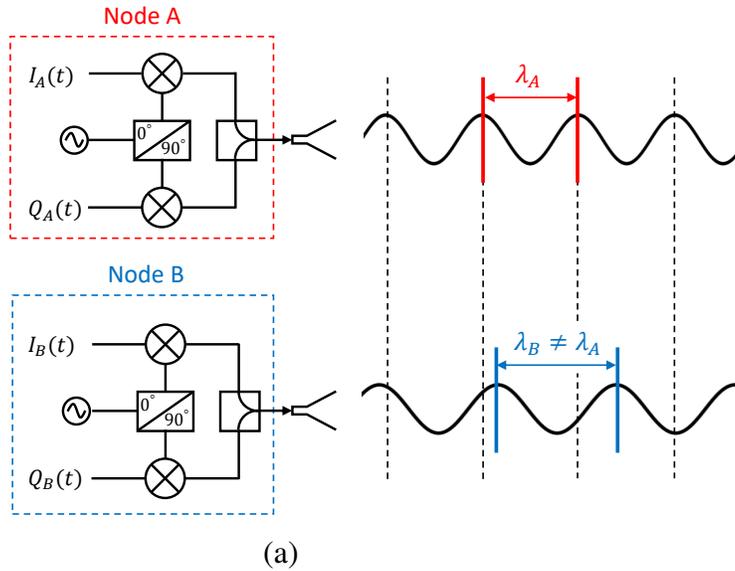


Figure 5.25: (a) Without phase and frequency synchronization, the nodes in the array cannot align their phases, as it is impossible to prevent the frequencies on each node to relatively drift, and even when the frequency seems to be stable, it is not possible to transmit coherent phases. (b) Once both phase and frequency are synchronized, it is possible to transmit a common coherent frequency to the target, and ranging is part of localization that enables phase alignment. From [10].

block diagram summarizing the importance of phase (through localization, which depends on time delay) and frequency synchronization for a coherent operation between the transmitting/receiving nodes is shown in Fig. 5.25.

Although this chapter does not focus on wireless frequency synchronization, a brief description of the used circuitry and approach is given. As mentioned in Chapter 1, wireless frequency synchronization, or frequency synchronization in general is essential for the operation of coherent distributed arrays, whether closed loop or open loop is implemented, otherwise it is not possible to transmit coherent signals to the target. In addition to that, wireless frequency synchronization is essential for cooperative ranging, more importantly when an active repeater is used especially with different carrier frequencies for the reception and transmission of the ranging signals. Using different carrier frequencies mitigates multipath and self-induced interference. In this case, Frequency locked repeaters minimize phase and frequency errors, and this can be explained in the following. Let us consider the case where a baseband signal (ranging pulse) with a frequency f_b is modulated on a carrier f_{c1} . This ranging pulse is transmitted from the radar and received at a repeater that downconverts it using the carrier f'_{c1} . f'_{c1} is nominally f_{c1} , and it is generally equal to f_{c1} if both the radar and repeater are frequency locked, otherwise it cannot be equal due to the relative frequency drift. If the general case is considered, the downconverted frequency on the secondary node represented by the repeater becomes $f_b + f_{c1} - f'_{c1}$. Similarly, this baseband frequency is upconverted again to $f_b + f_{c1} - f'_{c1} + f'_{c2}$ so it can be retransmitted back to the radar. Once received back by the radar, the downconverted frequency becomes $f_b + f_{c1} - f'_{c1} + f'_{c2} - f_{c2}$. Ideally, if both nodes are frequency locked, with $f_{c1} = f'_{c1}$ and $f_{c2} = f'_{c2}$, the recovered signal will have a frequency f_b . Nevertheless, when this is not the case, the recovered signal will have an offset frequency that will prevent the radar from estimating the correct range of the target. In addition, even the same bands were used for transmission and reception, $f_{c1} = f_{c2}$ and $f'_{c1} = f'_{c2}$, frequency synchronization is necessary as it significantly reduces the instantaneous phase and frequency errors, which leads to an improved ranging accuracy.

The block diagram of the wireless frequency synchronization circuit is shown in Fig. 5.26. Detailed explanation and analysis of this circuit is presented in Chapter 4. Briefly, this adjunct circuit receives a two-tone CW signal from the primary node, with a tone separation equal to the desired reference input frequency, which was equal to 10 MHz for the used SDRs. The circuit

amplifies and splits the received signals and mix them back to generate a 10 MHz output. The 10 MHz output frequency is amplified further to reach the desired power, which for the Ettus USRP X310 SDR was between 0 and 15 dBm. Finally, the amplified signals are fed to the reference input to lock the frequency of the secondary node to the that of the primary node.

As shown in Fig. 5.26, the outdoor adaptive ranging approach was achieved using three SDRs. Node A, which represents the primary node, was implementing the adaptive ranging algorithm in addition to transmitting the two-tone wireless frequency synchronization signals. Node B, a secondary node, was represented by an active repeater, connected to a self-mixing adjunct circuit, that allows it to lock its frequency to that of the primary node. All the carriers in this outdoor test were using the Industrial, Scientific, and Medical (ISM) frequency bands. The two tones of the wireless frequency synchronization signal were transmitted at 910 MHz and 920 MHz using one SDR. The signals were only composed of the two carrier frequencies, with no other modulation, to minimize any sort of noise in the extracted reference input frequency. These signals were fed to the frequency locking circuit that was equipped with a cavity-based band pass filter to ensure that no other signals than the two desired tones are being mixed together. The 10 MHz output of this adjunct self-mixing circuit was fed to the reference input of the repeater in order to ensure that both nodes are operating at the exact same frequencies. Once node B is locked to node A, the radar on node A starts transmitting its ranging signals at a 2.45 GHz carrier. The same adaptive framework that was used in Section 5.3.1.2, was used in this outdoor testing, and the adaptive approach is summarized in Fig. 5.27. Afterwards, node B receives these ranging signals, amplifies them, and retransmits them back to node A at a 5.8 GHz carrier. The antennas of the two-nodes were 90 m apart. The outdoor antennas can be seen in Fig. 5.28. Wide band grid antennas with an operational frequency from 600 MHz to 6.5 GHz were used, covering the entire desired bandwidth for this test. The antenna gain was 15 dBi to 26 dBi, depending on the frequency band, with 15 dBi corresponding to the lowest frequency band. All the antennas and the SDRs were equipped with surge protection devices. The hardware that was used for nodes A and B was placed indoors and is shown in Figs. 5.29 and 5.30.

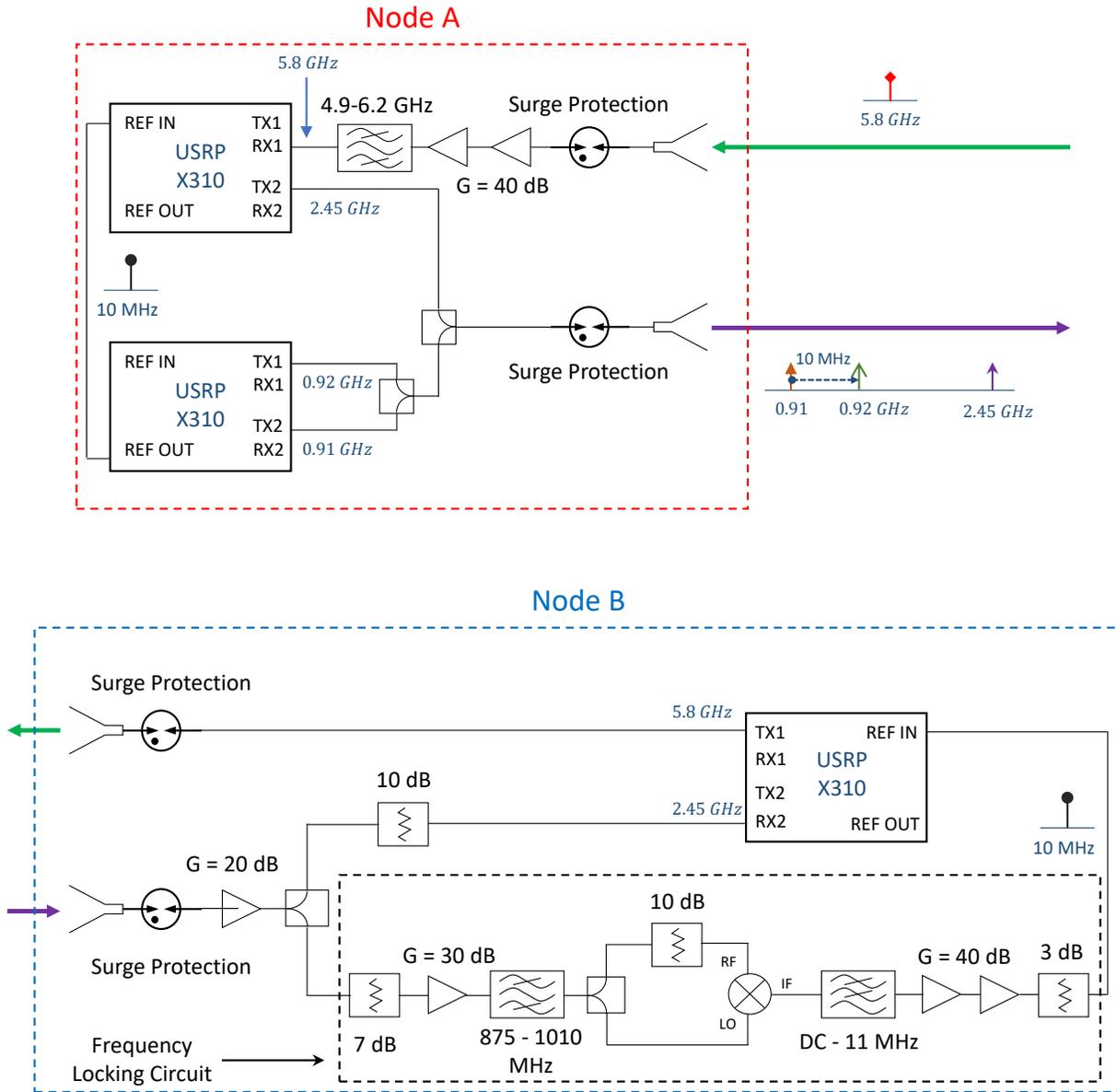


Figure 5.26: Block diagram for the outdoor adaptive ranging experiment, where wireless frequency synchronization was implemented. Node A is the primary node, while node B represents the secondary node. Wireless frequency synchronization two-tone CW signals was transmitted at 910 MHz and 920 MHz, and the ranging two-tone adaptive waveform was transmitted initially at 2.45 GHz carrier and it was received at 5.8 GHz carrier. From [13] © 2021 IEEE.

The transmitted power was no more than 15 dBm for every carrier, and due to the cable losses, the maximum Effective Isotropic Radiated Power (EIRP) was less than 36 dBm. The major attenuation resulted from the connection between the SDRs and the antennas. Since the attenuation in

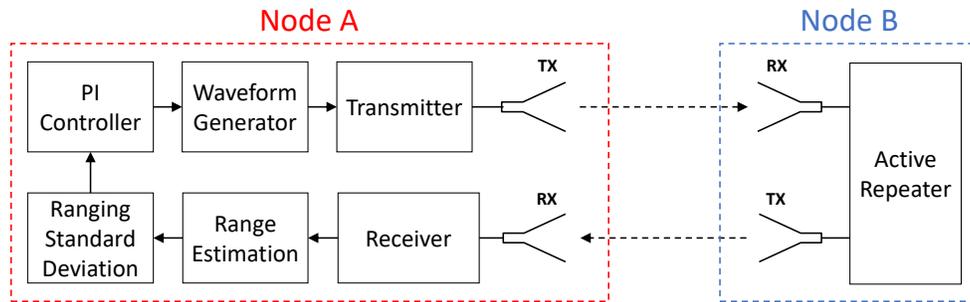


Figure 5.27: The outdoor adaptive inter-node cooperative ranging architecture. The same approach as in Fig. 5.18 was used, with the addition of a wireless frequency locked repeater. From [10].

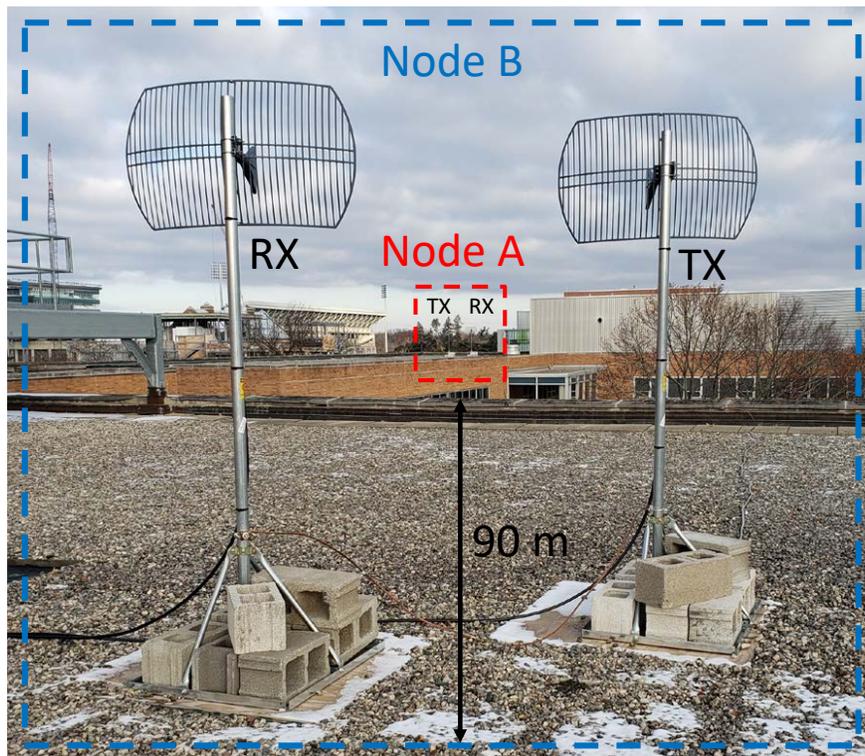


Figure 5.28: Each nodes was equipped with two antennas, one for transmit (TX) and one for receive (RX). The antennas on every node were separated by around 2 m, and the separation between the antennas of each node was 90 m. From [10].

general is not desirable, and for long distances, it is essential to ensure that a minimal attenuation is observed, the outdoor-rated LDF4-50A 1/2 cable was used to connect the antennas to the SDRs. Each connection was around 20 m. The attenuation is dependent on the type of the used cable and the transmitted frequency; for a 20 m LDF4-50A 1/2 cable, the average attenuation is as follows: 1.5 dB for the band 902 MHz to 928 MHz, 2.4 dB for the band 2.4 GHz to 2.5 GHz, and 4 dB



Figure 5.29: The hardware that was used to operate node A, including two SDRs.

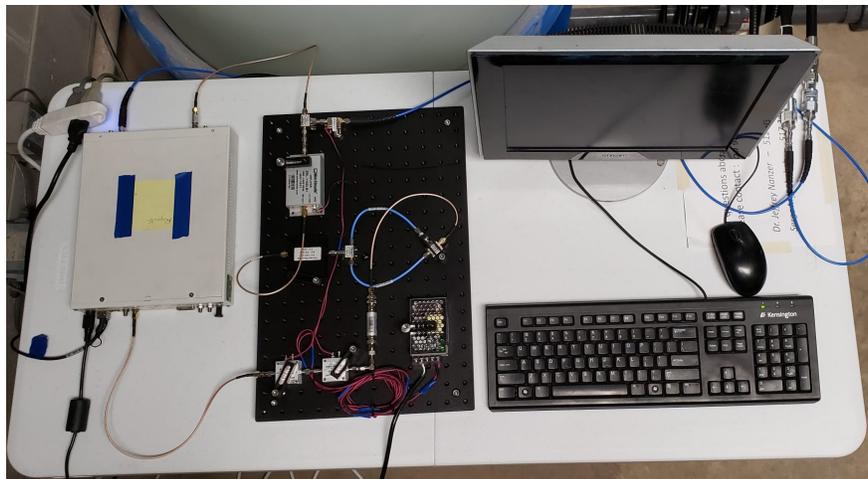


Figure 5.30: The hardware that was used to operate node B, including one SDR and the adjunct self-mixing circuit.

for 5.725 GHz to 5.875 GHz. Similarly, the antenna gain is dependent on the frequency, and the estimated values were as follows: 16 dBi for the band 902 MHz to 928 MHz, 22 dBi for the band 2.4 GHz to 2.5 GHz, and 25 dBi for 5.725 GHz to 5.875 GHz.

This ranging system was designed to enable monitoring the adaptive radar in varying weather conditions. The performance of the radar was affected by a multitude of uncertainties including varying SNR and interference levels, vibration, rain, among others. All these effects were assessed for the adaptive radar with the framework of Fig. 5.27. In this section the standard deviation was calculated using 200 consecutive range estimates. This calculated standard deviation was used as a feedback to the system. The transmitted baseband frequencies for the two-tone signal are

$f_1 = 20 \text{ kHz}$ and $20 \text{ kHz} \leq f_2 \leq 7.52 \text{ MHz}$. f_2 was determined by the output of the controller, and it allowed a bandwidth range between 0 and 7.5 MHz. The frequency of the disambiguation pulse was set to 1.875 MHz, equivalent to the fourth of the maximum possible bandwidth. The ranging pulse width and the disambiguation pulse width were both constant and equal to $143.7 \mu\text{s}$ and 533 ns respectively. A duration of $159.7 \mu\text{s}$ was allocated for each of the two pulses, giving a total time of $319.4 \mu\text{s}$. However, the ranging and disambiguation pulses were logged every 105 ms to allow continuous data logging and to prevent any buffer underflow or overflow. All the signals processing was performed real-time in LabVIEW.

The targeted standard deviation was set to 10 mm; this value was selected since it allows the controller to operate smoothly in the assigned bandwidth, which was 7.5 MHz. Lower or higher target values can be selected depending on the application, however, this target value allows the reader to observe the full benefits of using the proposed adaptive ranging approach, since on average a 3.5 MHz (half of the bandwidth) was needed to achieve a 10 mm standard deviation for the available SNR. Before calculating the standard deviation, 40 groups of 5 consecutive range estimates were assembled, the 5 estimates were averaged to obtain the final range estimate. This approach of averaging 5 consecutive estimates improves the standard deviation of the final estimate by a factor of $\sqrt{5}$. Once calibrated, the PI controller gain were $K_p = 10\mu$ and $T_i = 3.3$.

Weather conditions were recorded using a Davis 6250 Vantage Vue weather sensor, The weather sensor was attached to the mast of the receive antenna of node A as shown in Fig. 5.31. This sensor was able to measure and log the rainfall, pressure, humidity, wind direction, wind speed, and temperature once every minute. Using the Davis 6250 Vantage Vue sensor allowed me to monitor and display some of the weather effects on the ranging performance.

Evaluation of some of the external and weather effects on the ranging performance was done for 24 hours in total, and the data was captured from June 1, 2020 23:56 until June 2, 2020 23:56. 4090 processing intervals were present in the 24 hours interval. In the first outdoor experiment, the adaptive approach was not running, and ranging was done using fixed frequencies. f_1 was set to 20 kHz, and f_2 was set to 3.5 MHz. The ranging standard deviations during the 24 hours are



Figure 5.31: an image showing the weather sensor Davis 6250 Vantage Vue that was attached to the receive antenna of node A.

shown in Fig. 5.32a. Although the ranging waveform was constant, it can be seen that the standard deviation was ranging from 0.005 m up to 0.05 m (the standard deviation was calculated based on an averaging of 5 range estimates) in a 24 hour period, and these extremes can vary even further depending on the external conditions. Generally, high wind speed increase the antennas vibrations, leading to a degraded performance. High humidity and rain rate increase the attenuation, leading to a decrease in the SNR. Moreover, nearby transmitters such as WIFI networks induce time-varying interference. With interference power varying based on the network consumption and the relative location of the surrounding transmitters. Although the effects of some of the mentioned factors can be estimated for a given time, other effects from factors such as interference cannot be estimated in advance. The Effects of SNR, wind speed, humidity, and rain rate are visible in 5.32. The drop in SNR from 550 to 1600 processing intervals had a visible negative effect on the standard deviation. High wind speeds at around 300 processing intervals lead to an increase in the ranging standard deviation, while a decrease in wind speed at values beyond 3000 processing intervals contributed to the decrease in ranging standard deviation. Similarly, high humidity and rain rate from 550 up to 1000 processing intervals lead to higher ranging standard deviations.

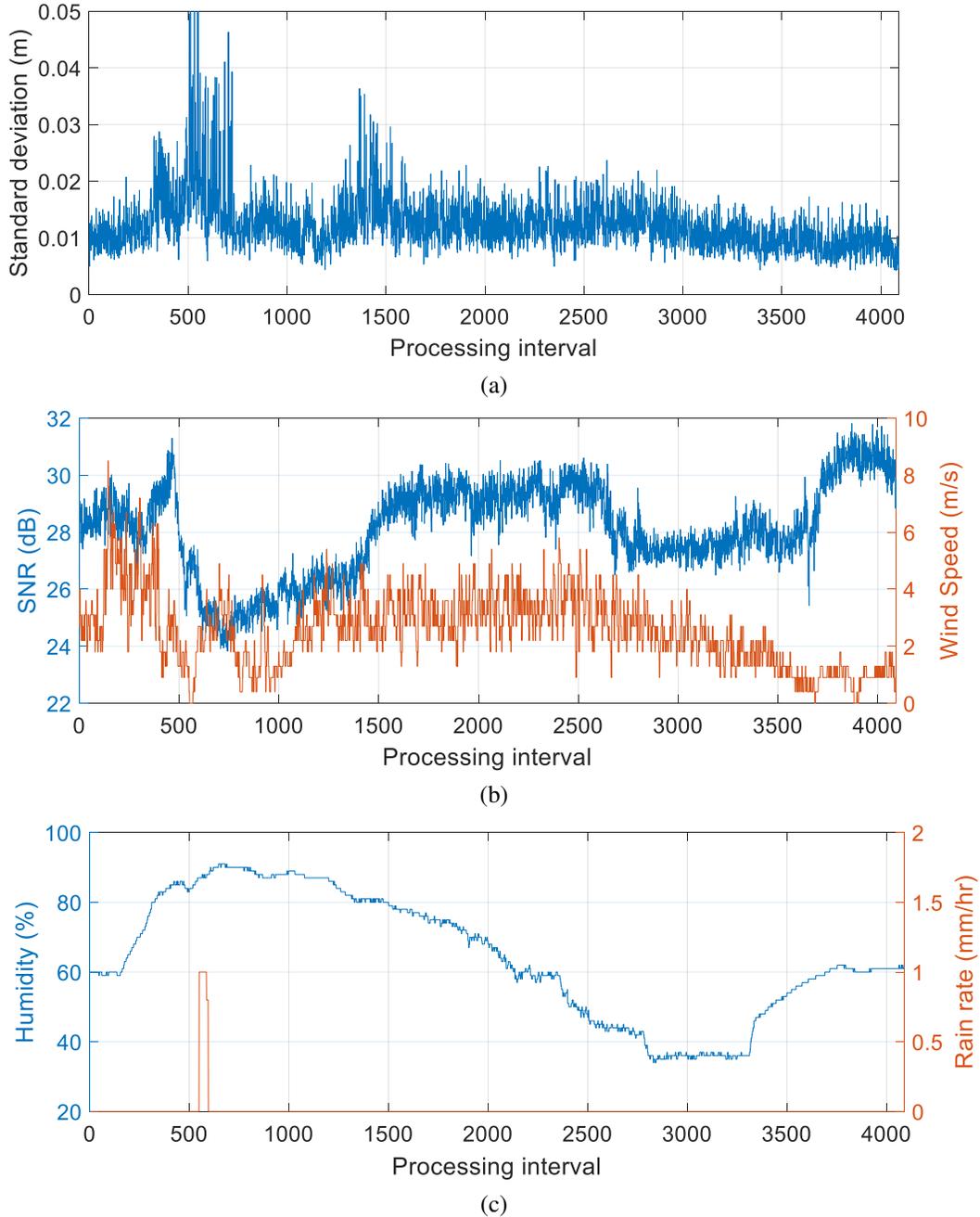


Figure 5.32: (a) Ranging standard deviation for a 24 hours interval, where $f_1 = 20$ kHz and $f_2 = 3.5$ MHz. The effects of SNR changes and wind speed were shown in (b). Also the contribution of humidity and rain rate was presented in (c).

Since in this dissertation we are interested in the beamforming gain, the ranging standard deviation results from 5.32a (treated as RMSE) were used to determine the coherent frequency that can be achieved for a desired beamforming gain in a two-node distributed array system. A coherent

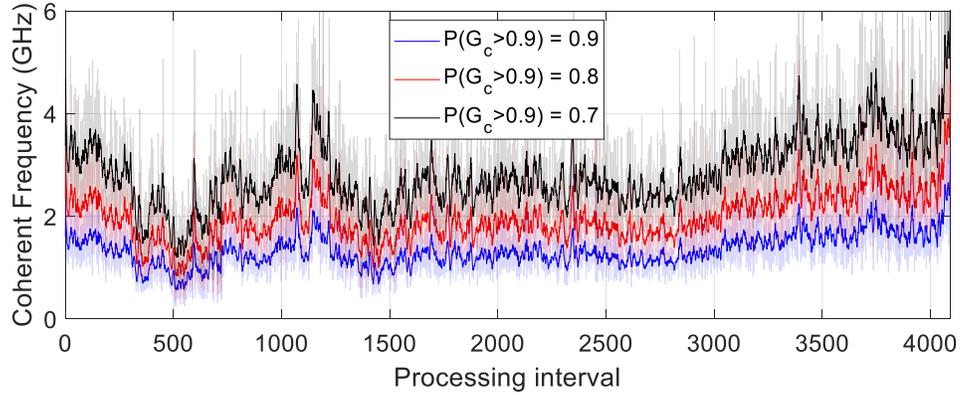


Figure 5.33: Achievable coherent frequency using the obtained ranging standard deviation. Three probabilities for achieving coherent gain of at least 0.9 were investigated. The lines with solid colors represent the moving average with a window of 10, while the faded lines represent the individual estimates.

gain of 0.9 was of interest, and the probabilities of 0.9, 0.8, and 0.7 to achieve this desired coherent gain for a desired carrier frequency were evaluated. The clarity of the achievable coherent frequency for the desired coherent gain was improved by plotting the actual values of the 4090 individual estimates using faded colors and their smoothed values using clear solid colors. The smoothed values were obtained using a 10-point average window, and the results are shown in Fig. 5.33. It can be seen that for $P(G_c > 0.9) = 0.9$, the maximum coherent beamforming frequency that can be supported varied between 0.7 GHz and 2.7 GHz. This shows the advantage of using an adaptive ranging waveform that ensures a coherent transmission for a desired carrier frequency for all weather conditions with minimal bandwidth for the ranging signals.

Once the achievable coherent frequencies were evaluated for a two-tone waveform with constant frequency, the adaptive ranging approach was tested for comparison. As mentioned earlier, the target ranging standard deviation was selected as 10 mm. The effects of weather conditions on the adaptive ranging approach were recorded for 24 hours, from June 3, 2020 03:05 until June 4, 2020 03:05. Fig. 5.34 show the same parameters as in Fig. 5.32, with the addition of the adaptive upper frequency f_2 of the adaptive ranging waveform. It can be seen that f_2 was adaptively modified to counteract the environmental and external conditions and ensure a stable performance for the ranging. The variation in both SNR and wind speed had the highest effect on the f_2 . The

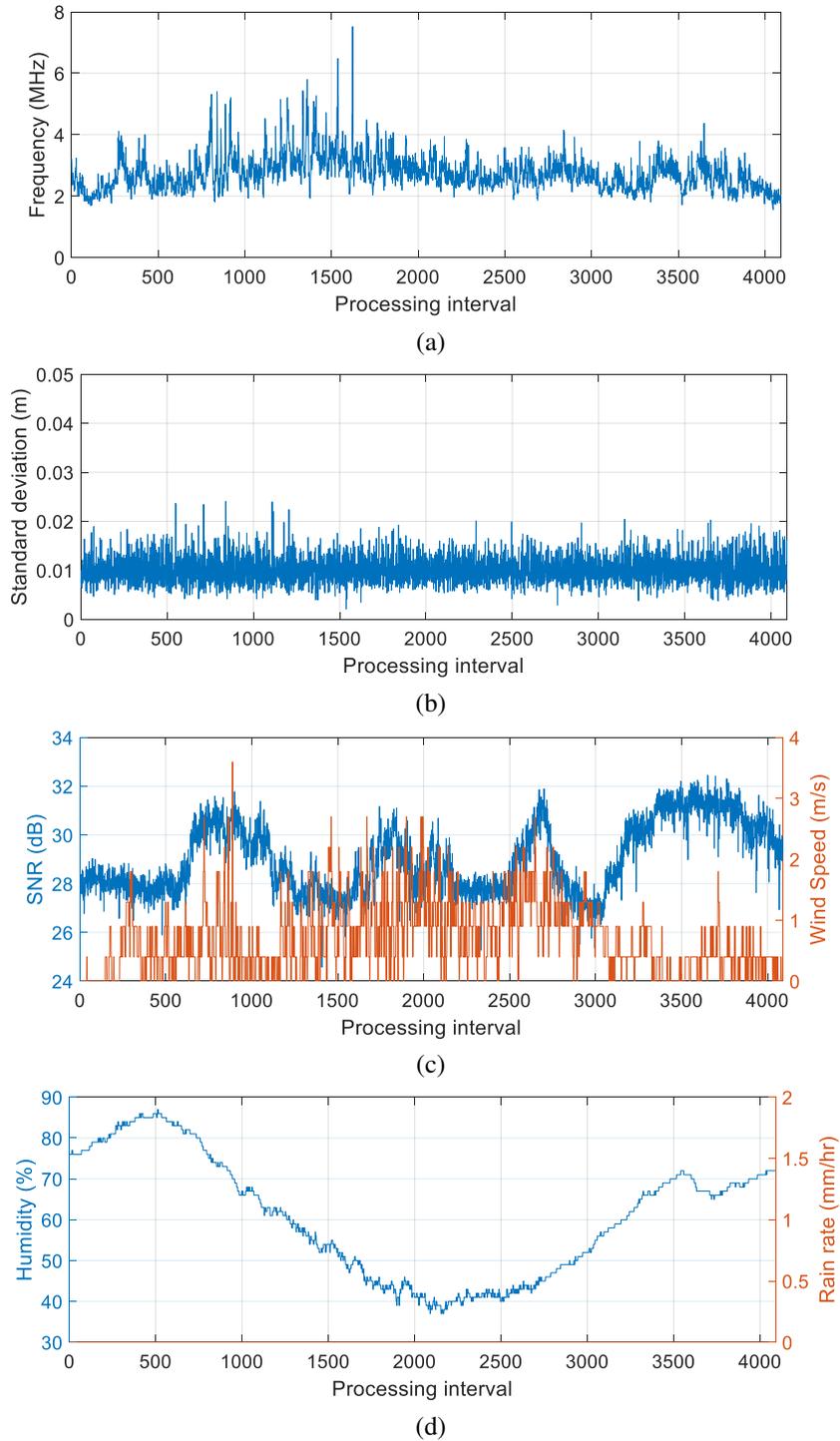


Figure 5.34: (a) The adaptively modified second tone f_2 of the transmitted ranging pulses. (b) Ranging standard deviation for a 24 hours interval, where $f_1 = 20$ kHz and f_2 was adaptively modified. The SNR changes and wind speed are shown in (c), and the humidity and rain rate are presented in (d).

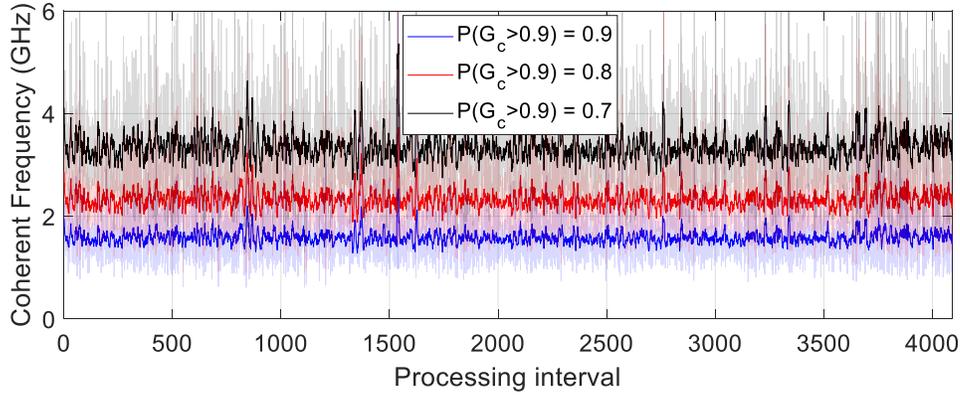


Figure 5.35: Achievable coherent frequency using the obtained ranging standard deviation in the adaptive framework. Three probabilities for achieving coherent gain of at least 0.9 were investigated. The lines with solid colors represent the moving average with a window of 10, while the faded lines represent the individual estimates.

achievable coherent beamforming frequency were reevaluated for the standard deviation values in Fig 5.34, and the results were displayed in Fig 5.35. It can be seen that the stability of the achievable coherent frequencies was considerably improved, this was possible due to the dynamic update of the waveform bandwidth. By setting the target ranging standard deviation to 10 mm, the maximum achievable coherent frequencies probabilities of 0.9, 0.8, and 0.7 to achieve a 0.9 coherent gain were 1.5 GHz, 2.2 GHz, and 3 GHz.

A longer evaluation of the adaptive framework was done with a period of 7 days, the data was captured from June 16, 2020 16:53 until June 23, 2020 16:53. Fig. 5.36 displayed the same parameters as in Fig. 5.34, with the only difference that in Fig. 5.36 the data was captured over a seven-day period. 28,630 processing intervals were captured in total. The achievable coherent frequencies over all this period are displayed in Fig. 5.37.

5.3.2 Using Only One Ranging Pulse as Feedback

We have seen in Section 5.3.1 the framework for achieving adaptive ranging to maintain a target ranging accuracy in the event of any change in the environment. One of the drawbacks in the proposed approach could be the long time before a modification is applied to the transmitted bandwidth. This delay is due to the time needed to calculate the standard deviation, where in section

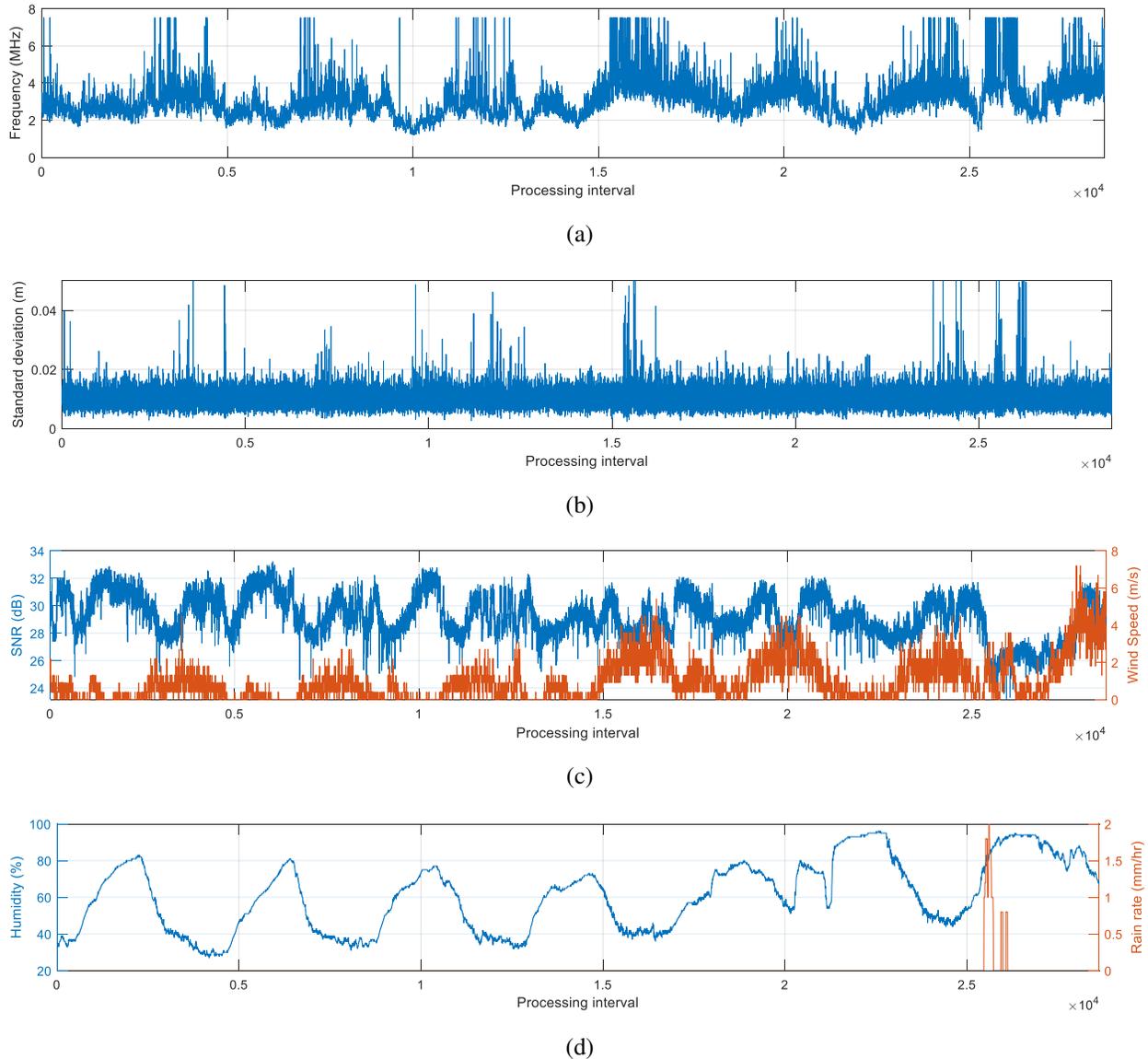


Figure 5.36: (a) The adaptively modified second tone f_2 of the transmitted ranging pulses. (b) Ranging standard deviation for 1 week interval, where $f_1 = 20$ kHz and f_2 was adaptively modified. The SNR changes and wind speed are shown in (c), and the humidity and rain rate are presented in (d).

5.3.1.3, 200 range estimates were used to estimate the standard deviation. The delay that is introduced by waiting for 200 range estimates before applying any change to the waveform is usually acceptable, since the required processing time before an update is on the order of milliseconds. However, speeding up this task can be beneficial, especially in fast changing environments. This can be done by modifying the perception process to reduce the time required before the controller

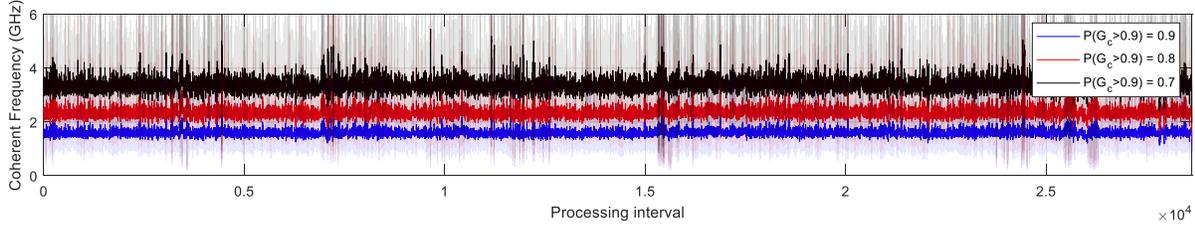


Figure 5.37: Achievable coherent frequency using the obtained ranging standard deviation in the adaptive framework. Three probabilities for achieving coherent gain of at least 0.9 were investigated. The lines with solid colors represent the moving average with a window of 10, while the faded lines represent the individual estimates.

can determine the new update to the waveform. The modification can be done by selecting a metric other than the standard deviation of the range estimates to serve as feedback to the controller. In this section we implement an adaptive ranging framework that relies on the SNR of the received pulses rather than the standard deviation. Based on the observed SNR, the second tone of the ranging waveform is modified to maintain a desired ranging accuracy while minimizing the waveform bandwidth. On one hand, this approach is fast and can account for any drop in SNR, on the other hand, the presented approach cannot act against wind induced degradation, since it cannot detect the effect of vibration.

5.3.2.1 SNR estimation from One Pulse

Accurate SNR estimation is essential for the proposed adaptive ranging algorithm and plays a significant role in wireless system generally. SNR is used to characterize wireless systems, signal processing algorithm, and communications performance, among others. SNR can be used for decision making, as it is intended to be used in this work or other works such as in [140], where it was used to adaptively modify the modulation of Quadrature Amplitude Modulation (QAM) signals. SNR can generally be used to estimate the bit-error-rate (BER), or the sensing accuracy.

Various techniques can be used to estimate the SNR of a given signal. These techniques include split-symbol moments estimator [141], maximum-likelihood estimator [142], squared signal-to-noise variance estimator [143], second- and fourth-order moments [144], and signal-to-variation

ratio [145], among others. A comparison of these SNR estimators is shown in [146]. For blind SNR estimation, eigenvalue decomposition [147] can be used. Although this approach is applicable for most of the cases, it suffers from long processing time due to its covariance matrix calculation and it can be applicable only when multiple pulses are captured. These impediments makes this estimator less desirable, especially that in this work we are interested in fast SNR estimation. SNR estimation using eigenvalue decomposition is summarized in the following.

Eigenvalue decomposition-based SNR estimator can estimate the power of the signal and noise without prior knowledge of the received waveform, making it a perfect candidate for SNR estimation for RF signals. In essence, this algorithm starts by forming a matrix \mathbf{X} from L received pulses with N samples each. \mathbf{X} is represented by

$$\mathbf{X} = \begin{bmatrix} \chi_{1,1} & \chi_{1,2} & \cdots & \chi_{1,L} \\ \chi_{2,1} & \chi_{2,2} & \cdots & \chi_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ \chi_{N,1} & \chi_{N,2} & \cdots & \chi_{N,L} \end{bmatrix}. \quad (5.62)$$

The covariance matrix, which is the most time consuming operation for this estimator, is computed from

$$\mathbf{R}_x = \frac{1}{N} \mathbf{X} \mathbf{X}^H \quad (5.63)$$

where \mathbf{X}^H is the Hermitian of the matrix \mathbf{X} . Singular-value decomposition is used to estimate the eigenvalues λ_L of the matrix \mathbf{R}_x . Once obtained, the eigenvalues are sorted by decreasing order. Since the desired signal, which is represented by L pulses, is correlated, the highest eigenvalue λ_1 typically represents the power of the received signal. In some cases, for complex waveforms, λ_2 , and/or λ_3 , etc., can represent part of the received pulse. Assuming that only λ_1 represents the power of the received pulse, $\lambda_2 - \lambda_L$ represent the power of the noise. The noise level can be estimated using

$$\gamma^2 = \frac{1}{L-1} \sum_{l=2}^L \lambda_l. \quad (5.64)$$

The signal power level is calculated using

$$P_s = \frac{\lambda_1 - \gamma^2}{L}. \quad (5.65)$$

In the case where the transmitted signal is a continuous wave or it does not have any non-zero parts, the SNR is calculated by

$$\text{SNR}_{dB} = 10 \log_{10} \left(\frac{P_s}{\gamma^2} \right). \quad (5.66)$$

For pulsed signals, which is the case in the presented work, (5.66) can be modified to accommodate for the used pulse width by

$$\text{SNR}_{dB} = 10 \log_{10} \left(\frac{P_s}{\gamma^2} \right) + 10 \log_{10} \left(\frac{\text{PRI}}{T} \right). \quad (5.67)$$

This way, the power of the noise will be only considered in the area where the captured signal contains the ranging pulse.

5.3.2.2 Ranging Standard Deviation Estimation from SNR

The proposed SNR estimation approach is based on comparing the matched filter output of the received ranging pulse to that of the transmitted pulse. This step allows us to determine the signal power level. As for the noise level, it is estimated by observing the received signals between every two ranging pulses. By estimating the signals level in these regions, the noise or interference levels can be estimated. Once both the signal RMS amplitude and the noise RMS amplitude are obtained, SNR can be easily estimated. The advantage of this approach is that it is computationally inexpensive, since most of the computation, which is represented by computing the matched filter outputs, can either be done before the ranging algorithm starts running in real-time (for the case of matched filter output of the transmitted pulse or autocorrelation), or is already done when estimating the range (for the case of matched filter output of the received pulse). This SNR estimator can only be applicable if the signal of interest is already known, which is the case for radars.

To visualize the presented algorithm, let us consider that we have a transmitted signal S_{RX} with $f_1 = 10$ kHz and $f_2 = 1.01$ MHz as represented in Fig. 5.38. The received signal S_{TX} will have a

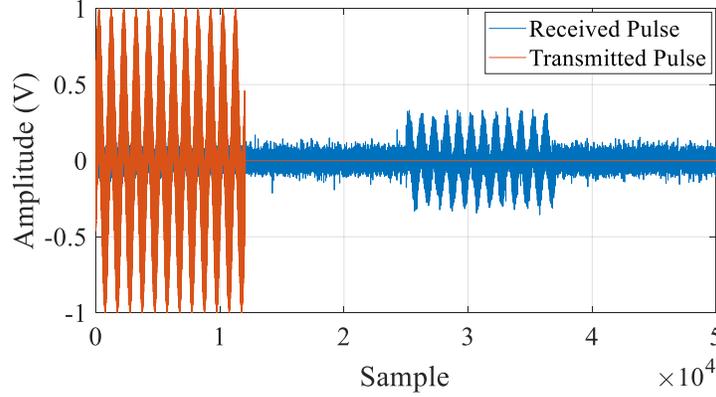


Figure 5.38: An example of transmitted (orange) and received (blue) two-tone signal with 11 dB SNR. From [12] © 2021 IEEE.

smaller amplitude with added noise, which we assume in this section that is AWGN. The matched filter output of the received pulse can be evaluated using

$$S_{out}[m] = \sum_{m=-\infty}^{\infty} h[n - m]S_{RX}[m], \quad (5.68)$$

where $h[n] = S_{TX}^*[-n]$ is the linear filter that maximizes the SNR of the received signal. The autocorrelation is computed similarly by replacing $S_{RX}[m]$ with $S_{TX}[m]$ in (5.68). The autocorrelation can be estimated before the ranging starts, since we will be defining the transmitted waveform, which reduces the computational time. As for (5.68), the matched filter output is obtained when performing the time-delay estimation, thus, similarly to the autocorrelation step, no extra computation is needed. When computing the autocorrelation and cross-correlation, the only parameters needed for the SNR estimation is the magnitude of their highest peak. The autocorrelation and cross-correlation of the signals in Fig. 5.38 are shown in Fig. 5.39.

Once (5.68) is performed, the time delay is estimated for the ranging step, thus, the area that only contains the noise components can be determined using the knowledge of the pulse width and PRI. Smaller noise area can be selected to account for any time-delay estimation error. Once the area that only contains noise is determined, the RMS noise amplitude A_N is directly obtained.

The SNR can be calculated once A_S is determined, and this can be done by comparing the peak of the matched filter output with the peak of the autocorrelation of the transmitted signal.

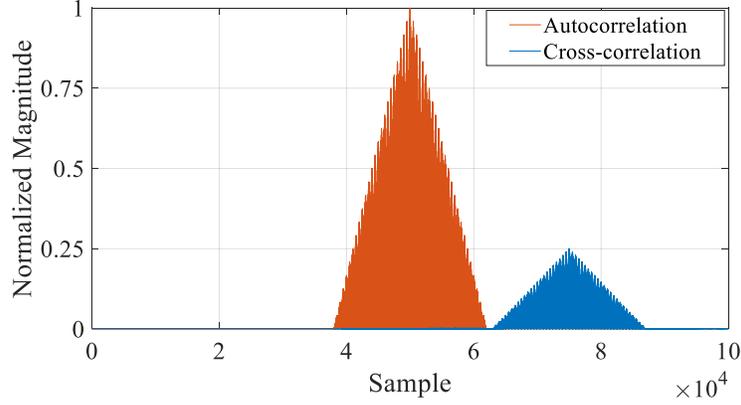


Figure 5.39: A simulated example of the autocorrelation and cross-correlation of the pulsed two-tone signals that are shown in Fig. 5.38, where the ratio of amplitudes of the received to transmitted pulses is $K = 0.25$. From [12] © 2021 IEEE.

Generally, a simplified representation of the received signal can be obtained using

$$S_{RX}[n] = K \cdot S_{TX}[n] + w[n], \quad (5.69)$$

where K is a scaling factor ranging from 0 to 1, and w represents the AWGN noise in the received signal. Combining (5.68) and (5.69) gives

$$S_{out}[m] = K \sum_{m=-\infty}^{\infty} h[n-m]S_{TX}[m] + \sum_{m=-\infty}^{\infty} h[n-m]w[m], \quad (5.70)$$

where

$$\sum_{m=-\infty}^{\infty} h[n-m]S_{TX}[m] = \sum_{m=-\infty}^{\infty} S_{TX}^*[m-n]S_{TX}[m] \quad (5.71)$$

is the autocorrelation of the transmitted pulse. The term

$$\sum_{k=-\infty}^{\infty} h[n-m]w[m] \quad (5.72)$$

is the cross-correlation of the transmitted pulse with the received noise. Since both terms are uncorrelated, $\sum_{m=-\infty}^{\infty} h[n-m]w[m]$ tends to zero for long pulses with sufficient integration time or after averaging multiple captured pulses, yielding

$$S_{out_{avg}}[m] = K \sum_{m=-\infty}^{\infty} S_{TX}^*[m-n]S_{TX}[m]. \quad (5.73)$$

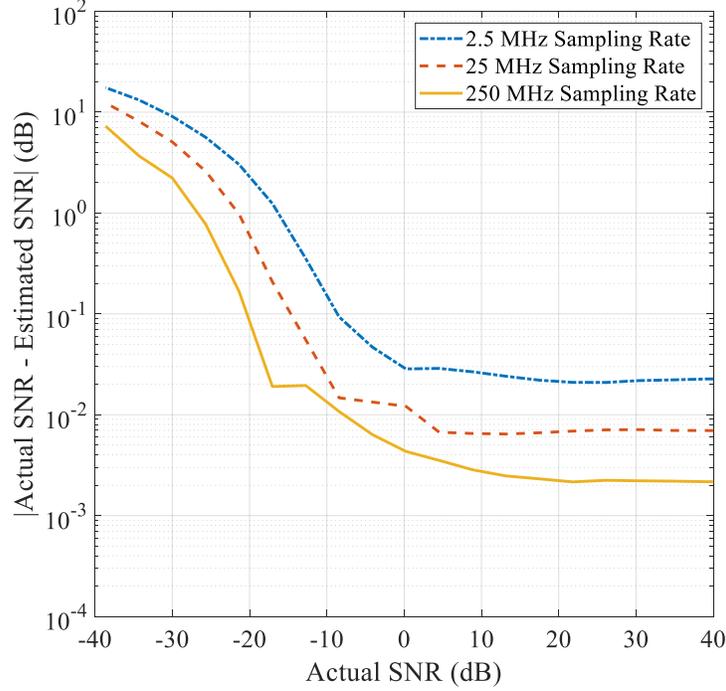


Figure 5.40: Estimation accuracy of the eigenvalue decomposition-based SNR estimator. Three sampling rates were selected to evaluate the change in accuracy. From [12] © 2021 IEEE.

Once both $S_{out_{avg}}$ and $\sum_{m=-\infty}^{\infty} S_{TX}^*[m-n]S_{TX}[m]$ are calculated, K can be estimated by comparing the magnitude of both peaks, where K is the ratio of peaks. After determining K , A_S is computed from

$$A_S = K \sqrt{\frac{1}{N} \sum_{n=0}^N (S_{TX}[n])^2}. \quad (5.74)$$

The SNR estimate is thus given by

$$\text{SNR}_{dB} = 10 \log_{10} \left(\left[\frac{A_S}{A_N} \right]^2 \right). \quad (5.75)$$

The accuracy of this SNR estimator is dependent on the ability to estimate accurately the peak magnitude, which is possible if the sampling rate is high enough. Moreover, the estimation accuracy of the peak can be further improved by utilizing interpolation, sinc fitting, zero crossing of Hilbert transform, and FFT pruning [148], among others. These are the same techniques that can be used to estimate the refined peak location, which were discussed in Section 5.2.

The SNR estimation accuracy of the proposed approach (using the matched filter output) was compared to that of the eigenvalue decomposition approach through simulation. The transmitted

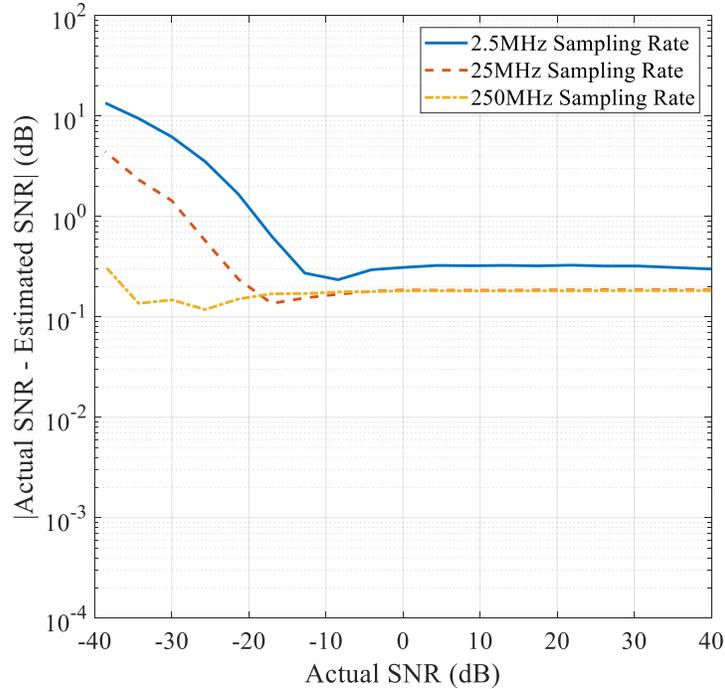


Figure 5.41: Estimation accuracy of the matched filter-based SNR estimator. One of the two tones had a $\pm 90^\circ$ (same results for both cases) phase shift in comparison to the filter; which leads to the lowest accuracy since only the in-phase signal is being considered. Three sampling rates were selected to evaluate the change in accuracy. From [12] © 2021 IEEE.

signals were based on a two-tone waveform with 1.2 ms pulse width, $f_1 = 10$ kHz, and $f_2 = 1.01$ MHz. 100 captures with 5 ms duration each were processed for every case. The SNR was estimated for the case where only the in-phase part of the signal was captured. By considering only the in-phase part of the signal, the initial phase of the received pulse plays a big role in the achievable accuracy for the proposed method. In the case where we are matched filtering the complex signal with both in-phase and quadrature parts, the received phase of a two-tone pulse hardly affects the peak amplitude of the matched filter output. Fig. 5.40 shows the performance of the eigenvalue decomposition-based SNR estimator, whereas Figs. 5.41 and 5.42 show the performance of the matched filter-based SNR estimator. In Fig. 5.42, the received phase was set equal to the transmitted phase, which gives the highest accuracy, whereas in Fig. 5.41, the received phase of the lowest tones was shifted by $\pm 90^\circ$, giving the lowest possible estimation accuracy. Sampling rates of 2.5 MSps, 25 MSps, and 250 MSps were evaluated. The error of both estimation

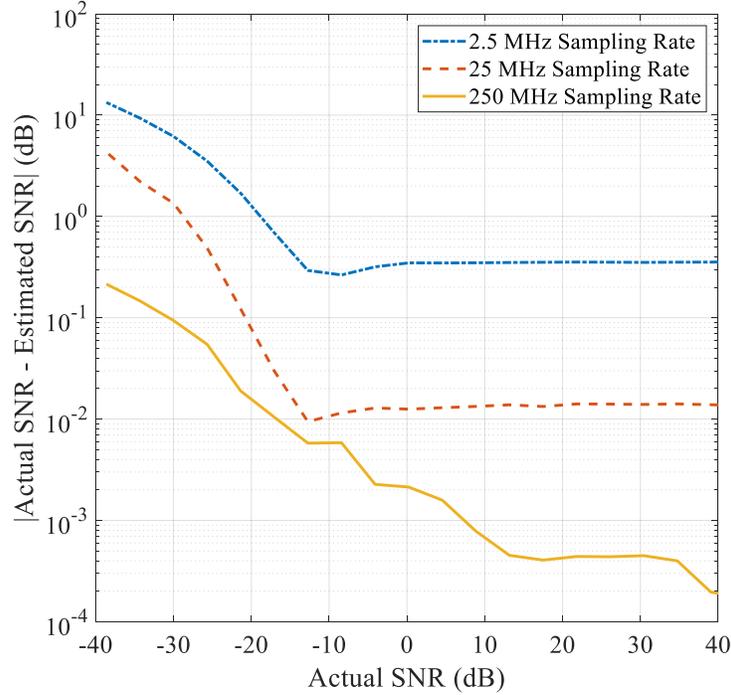


Figure 5.42: Estimation accuracy of the matched filter-based SNR estimator. The initial phases of both the signal and filter were equal, leading to the highest accuracy. Three sampling rates were selected to evaluate the change in accuracy. From [12] © 2021 IEEE.

algorithms was calculated using $E = |\text{Actual SNR} - \text{Estimated SNR}|$, where both the actual and estimate values were in dB.

The performance of the eigenvalue decomposition approach is superior at 2.5 MSps with an approximate of 10 dB improvement over the matched filter SNR estimation approach. However, when the phases are aligned, as in Fig. 5.42, the matched filter approach gives a much higher accuracy than the eigenvalue decomposition approach for the 250 MSps case, especially at high SNR values, where more than 10 dB improvement was observed. This behavior is expected, since at lower sampling rate, the peak of the matched filter output is discretized, and often a difference between the magnitude of the actual peak and the magnitude of the discretized point closest to the actual peak is observed. Nevertheless, by employing the peak estimators that were presented in Section 5.2, a refined estimate of the main lobe is obtained, leading to accuracies similar to what was observed for the 250 MSps case, even with low sampling rates. Moreover, the cooperative ranging approach operates typically in relatively high SNR regions, and in this case the SNR



Figure 5.43: Experimental setup for testing the proposed SNR estimation method that relies on the matched filter output of the received pulses. The transmitter was connected to the receiver via 2 m coaxial cable with an additional 30 dB attenuator to ensure that the received power is within the acceptable range.

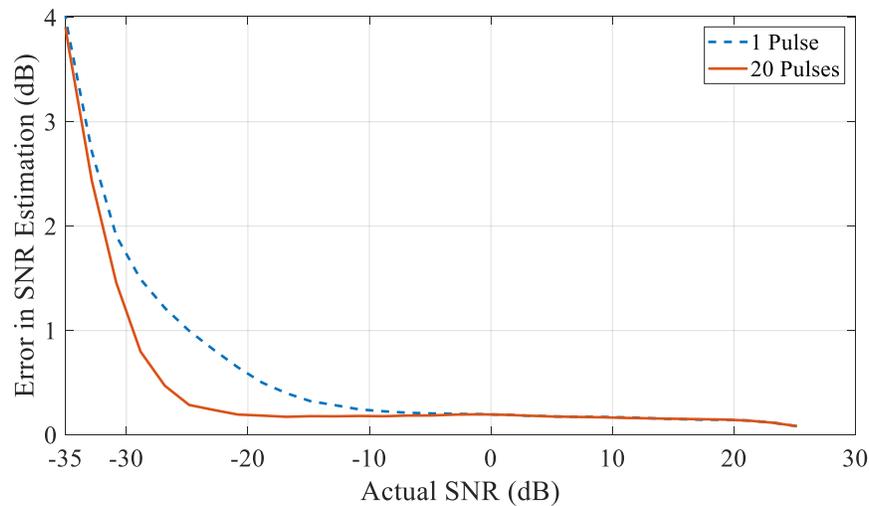


Figure 5.44: Measured SNR estimates using the matched filtering approach for a single pulse and 20 sequential pulses in a SDR-based system. The SNR was controlled by modifying the transmitted power. From [12] © 2021 IEEE.

estimation errors for both algorithms is very low, and can be ignored.

The matched filter-based SNR estimator was tested experimentally using the setup in Fig. 5.43. An Ettus USRP X310 SDR with two UBX 160 daughterboards was used. The transmitting port and the receiving port were both on separate daughterboards to minimize the interference. A 2 m cable was used to connect the two ports and to ensure that the SNR is being tested in a controlled environment where it is possible to obtain the actual transmitted power (within a small error margin). A 30 dB attenuator was added to ensure that the receive power does not exceed the

maximum power threshold. The noise power was held constant throughout the experiment and the transmitter power was adjusted electronically to modify the SNR.

Less than 2 dB error was observed at SNR values above -30 dB as shown in Fig. 5.44. Using 1 pulse to estimate the SNR yielded good estimates with no difference in comparison to using averaging of SNR values from 20 pulses once the actual SNR was beyond -5 dB. Generally, averaging the SNR estimate of 20 pulses improves the standard deviation of the estimates by a factor of $\sqrt{20}$. This improved SNR estimation accuracy over any other estimator is due to the knowledge of the received signals, since the pulses are used for the radar. The processing gain from the matched filter was 31 dB, which explains the ability of this SNR estimator to estimate accurately SNR values down to -31 dB. The 0.25 dB constant error from -20 dB up to 26 dB is due to the instability in noise power of the system.

5.3.2.3 SNR-Based Perception Design

As mentioned earlier, the aim of this section is to minimize the processing time in the perception step that was introduced initially in Fig. 5.18. This can be done by utilizing minimal information to deliver a feedback to the controller, which in this section is achieved by using the SNR of the received signal instead of the standard deviation of the range estimates. Consequently, the new block diagram representing the adaptive framework is given in Fig. 5.45. However, the ranging accuracy still needs to be the deciding factor in the adaptive loop, but instead of extracting it from consecutive range estimates, the SNR estimates are used to estimate it. This procedure reduces considerably the processing time, since a feedback can be given to the controller once one ranging pulse is received, instead of waiting for tens or hundreds of pulses before an action can be made.

The ranging standard deviation is estimated using the estimated SNR from one pulse as explained in Section 5.3.2.1. The estimated SNR is fed to a range accuracy estimator block that once calibrated, only relies on the estimated SNR. The ranging accuracy estimator is based on the CRLB that is defined in (5.16) for moderate to high SNR. The preprocessing SNR is embedded in **SNR**

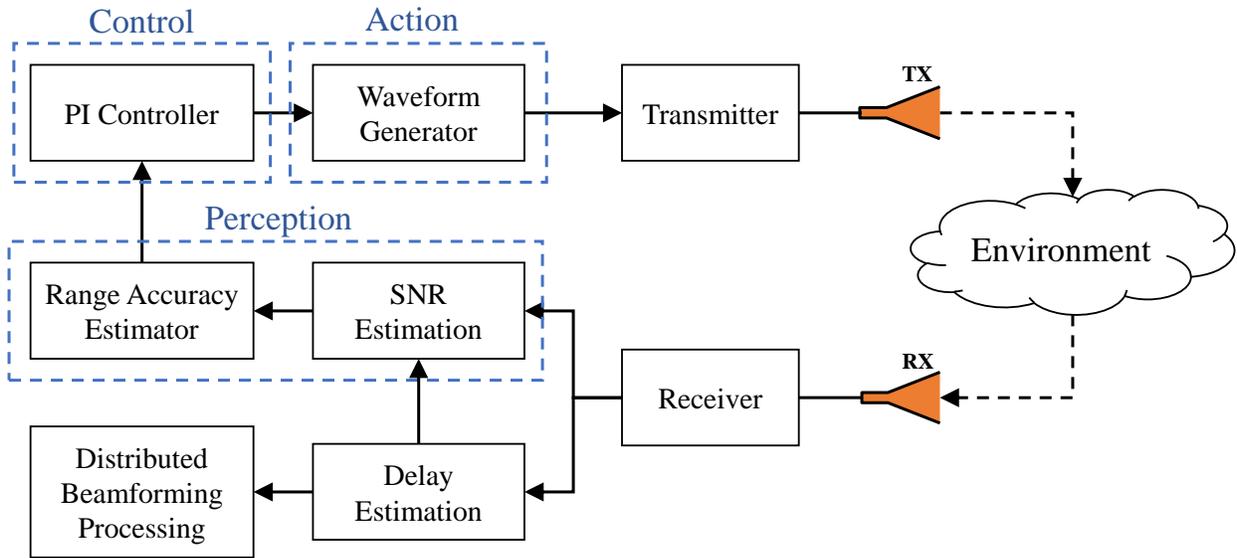


Figure 5.45: Adaptive ranging is necessary when high ranging accuracy and low spectral footprint are needed. This can be done by implementing three processes that ensures that only the desired bandwidth is used. A perception process is used to estimate the SNR of the received ranging pulses and convert this SNR to an estimated standard deviation. The controller compares the estimated standard deviation to a desired value and sends the necessary feedback to the action block that generates the desired waveform. This generated waveform is used for ranging, until a new waveform is generated for the next loop. From [12] © 2021 IEEE.

which is defined in this dissertation as

$$\mathbf{SNR} = \frac{2E}{N_0} = 2 T BW_r \mathbf{SNR}. \quad (5.76)$$

In the majority of the cases the variance (and standard deviation consequently) of the time-delay estimates cannot reach the CRLB, due to quantization errors, noise, nonidealities of the system, distortion, the use of repeaters, among other factors. However, the variance of the time-delay estimates is inversely proportional to the SNR and it can be predicted using the SNR estimates by adding a scaling factor to (5.16), so that the CRLB curve follows the actual variance of the system. The scaling factor is obtained through initial calibration of the system, where the performance of ranging in the desired system is evaluated over a variety of SNR values.

The scaling factor that is applied to the CRLB curve so that it follows the observed variance is determined using an experimental setup similar to Fig. 5.43. The scaled CRLB curve is compared to the calculated variance for multiple SNR values as shown in Fig. 5.46. As can be seen, the

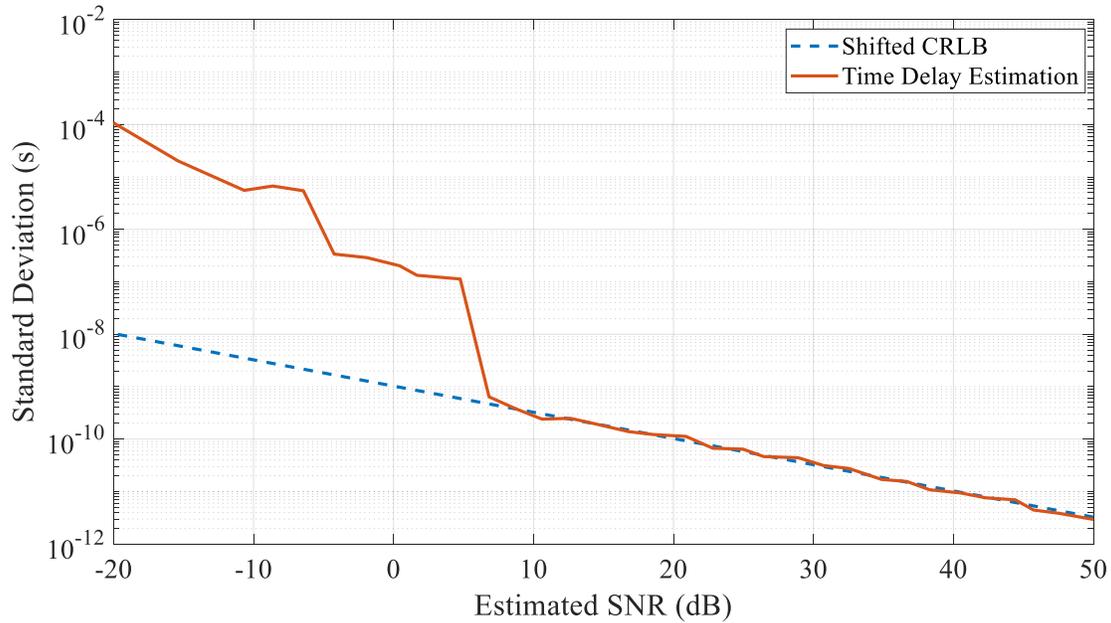


Figure 5.46: Measured standard deviation of the time delay estimator versus the scaled CRLB curve, demonstrating that the CRB from (5.16) can be scaled using a constant parameter to match the observed standard deviation. This experiment was done where no vibration or interference were present. From [12] © 2021 IEEE.

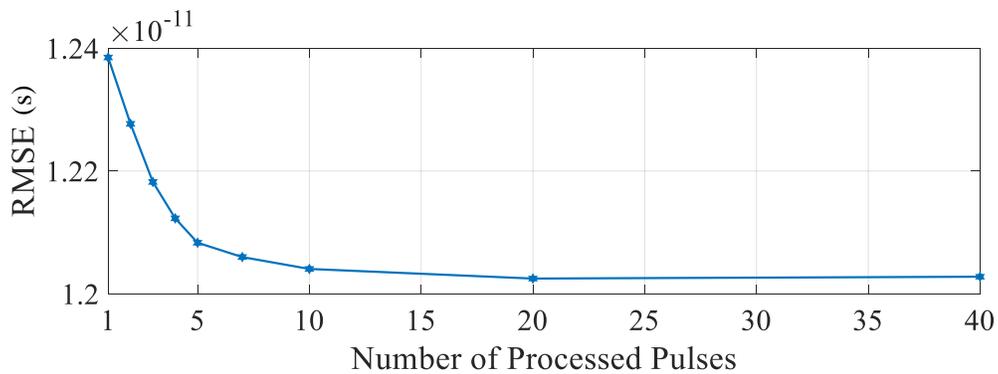


Figure 5.47: Measured RMSE of ranging accuracy versus SNR (averaged over 100 measurements) showing only a 3.7 ps improvement when more than one pulse is used. From [12] © 2021 IEEE.

two curves track with minimal errors for SNR values larger than 6 dB, which is going to be the case for cooperative ranging systems. The approximation is extendable to even lower values once the time-bandwidth product is increase by modifying the sampling rate or the pulse width. This scaled CRLB curve is used later to estimate the time delay variance/standard deviation using only the SNR estimate of one ranging pulse.

The effect of using only one pulse versus multiple pulses to estimate the SNR (using the pro-

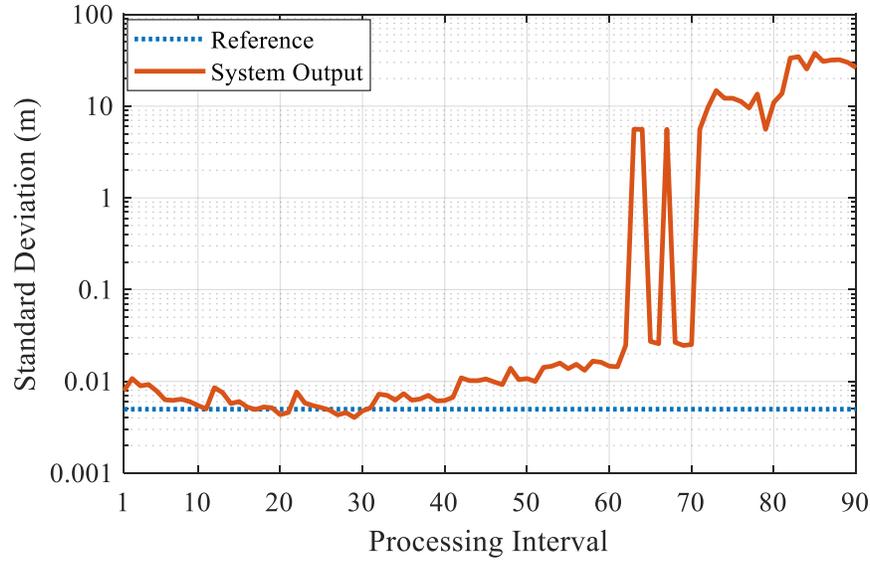


Figure 5.48: Experimental setup for the adaptive ranging system with the perception process that relies only on the estimated SNR. The two horn antennas were used for transmission and reception, and the corner reflector represented the target. From [12] © 2021 IEEE.

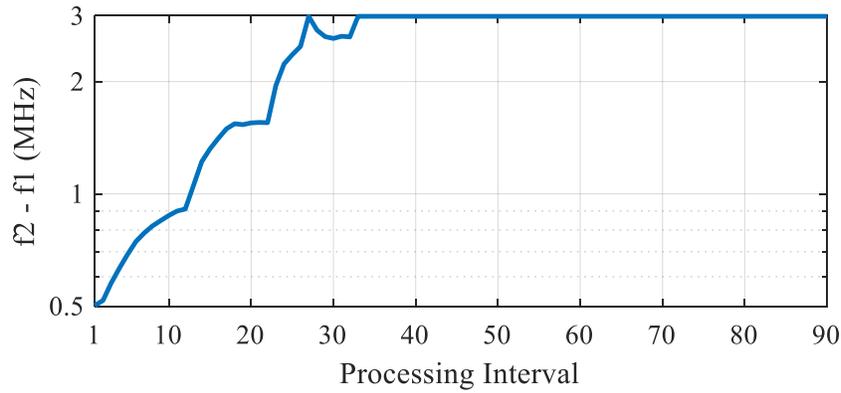
posed algorithm) on the estimated standard deviation for the time-delay estimates is analyzed in Fig. 5.47. As can be seen a minimal improvement is observed when calculating the RMSE between the adjusted CRLB curve and the actual standard deviation for varying number of pulses. The difference in RMSE between using 1 or 40 pulses to estimate the SNR was only 3.7 ps which is relatively negligible. This shows the robustness of this proposed modification to the perception process.

5.3.2.4 Adaptive Ranging Results

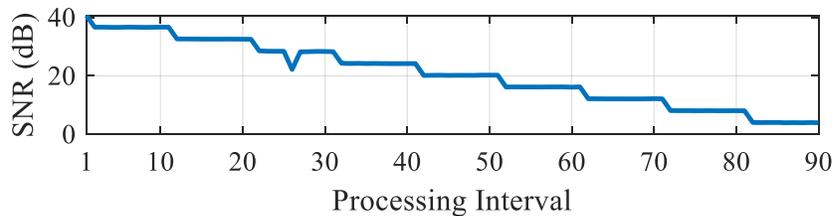
The proposed adaptive ranging algorithm using the modified perception process was tested experimentally using the Ettus USRP X310 SDR with UBX 160 daughterboards. Two horn antennas with 15 dBi gain and operational range from 3.85 GHz to 5.85 GHz were used for transmission and reception as shown in Fig. 5.48. A 21.5 dB low noise amplifier (LNA) was connected on the receive port. The target was represented by a corner reflector placed 4.5 m away from the antennas. The pulse width was set to 1.2 ms and the maximum allowed tone separation was set to 3 MHz. The lower tone was transmitted at a 10 kHz baseband. The sampling rate was set to 10 MSps. The target ranging standard deviation was set to 5 mm in this experiment, which corresponds to a time-



(a)



(b)



(c)

Figure 5.49: Performance of the adaptive ranging approach with modified perception: (a) ranging performance; (b) tone separation; (c) SNR. The SNR was modified by changing the transmitted power. The adaptive system was able to keep up with the change in SNR until a higher bandwidth than what is available was needed. From [12] © 2021 IEEE.

delay standard deviation of 33.3 ps. This value was solely chosen so that it can show clearly the operation of the proposed adaptive ranging approach. Similarly to Section 5.3.1.2, the parameters of the PI controller were tuned using the Ziegler-Nichols tuning method. The gain parameters for

this architecture were as follows: $K_p = 10, 125$ and $T_i = 4.25$.

The transmitted power was adjusted electronically every 10 processing intervals (processing loop cycles) to evaluate the performance of the controller. As can be seen in Fig. 5.49c, the SNR initially was 40 dB, and it was reduced constantly until reaching an SNR close to 4 dB. Fig. 5.49b shows all the assigned bandwidths by the controller to counteract the drop in SNR. The adaptive ranging performance was displayed in Fig. 5.49a, where it was compared to the target value, which was 5 mm. As expected, the bandwidth kept on increasing with every decrease in SNR until reaching a point where the adaptive system cannot keep up with the decrease in SNR. This point, which is shown after the 30th processing interval, is reached when the desired bandwidth exceeds the allowed value, which was set to 3 MHz. Between 60 and 70 processing intervals, the SNR reached a point where it is not possible for the matched filter to always detect the main lobe, as a result, the standard deviation was alternating between low and high values. This can be fixed by using the disambiguation pulse as explained in Section 5.2. The same behavior was observed in Fig. 5.46, but at a different SNR threshold value. This is the case because a different waveform was used (different bandwidth).

With the results in Fig. 5.49, it is possible to conclude that the proposed adaptive ranging approach is feasible for the SNR values that can be attained in cooperative systems. Furthermore, real-time implementation is feasible, especially when the standard deviation is estimated using only one ranging pulse. Thus, the limiting factor for the controller processing time is in this case the PRI of the ranging pulse, which is the best case scenario in terms of processing time.

CHAPTER 6

BEAMFORMING EXPERIMENTS

Coherent beamforming using centralized open-loop coherent distributed arrays is possible once the electrical states of the secondary nodes are synchronized to that of the primary node. In this dissertation, I am investigating the coherent beamforming of CW signals, which alleviates the need to synchronize the time on the distributed nodes. Thus, only the frequency and phase need to be synchronized. Frequency synchronization is done using the proposed adjunct self-mixing circuit in Chapter 4. As for phase alignment, in this chapter I consider the cases where the angle of the primary node relative to a secondary node is given, and only the distance needs to be estimated in order to determine the desired correction to the relative phase shift on a given secondary node.

6.1 Effects of the Displacement of Nodes on the Beamforming Performance

As seen in Chapter 4, any change in distance between the synchronization antennas induces a relative phase shift on the secondary nodes due to the frequency locking circuit, which can be estimated using (4.19). In addition, a second phase shift $\Delta\phi_{c_2}$ is obtained due to the change in distance between the coherent transmitters of both nodes. This observed phase shift on the secondary nodes is estimated from

$$\Delta\phi_{c_2} = -\frac{2\pi\Delta d_T \sin(\theta_0)}{\lambda_c}, \quad (6.1)$$

where Δd_T is the displacement distance separating the transmitter of the primary node and the transmitter of the secondary node of interest, and θ_0 represents the beamsteering angle. Thus, the total phase shift on the secondary node due to the relative displacement is estimated using

$$\Delta\phi_c = \Delta\phi_{c_1} + \Delta\phi_{c_2}. \quad (6.2)$$

Once this phase shift is estimated, it can be subtracted from the transmitted phase of the secondary node of interest to align the phases of the nodes in the array.

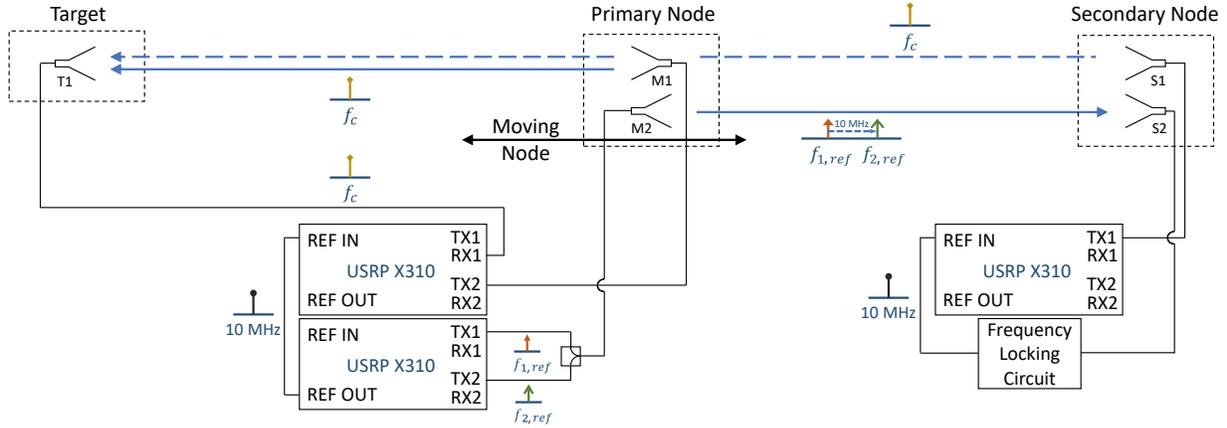


Figure 6.1: Block diagram of the setup used to measure the performance of the distributed two-node beamforming experiment. From [14] © 2021 IEEE.

As seen from (4.19) and (6.1), or (6.2), the phase shift caused by the change in distance of separation between the array is maximized when $\theta_0 = 90^\circ$, and is equal to 0 when $\theta_0 = 270^\circ$. This is the case because at $\theta_0 = 90^\circ$, $\sin(\theta_0) = 1$, leading to $\Delta\phi_{c_2} = \Delta\phi_{c_1}$ (assuming that $\Delta d_{IN} = \Delta d_T$). On the other hand, when $\theta_0 = 270^\circ$, $\sin(\theta_0) = -1$, leading to $\Delta\phi_{c_2} = -\Delta\phi_{c_1}$ which makes both phase shifts add destructively (assuming that $\Delta d_{IN} = \Delta d_T$). Thus, the most challenging scenario is when $\theta_0 = 90^\circ$, which is the case that is being studied in this chapter.

The phase shifts $\Delta\phi_{c_1}$, $\Delta\phi_{c_2}$, and $\Delta\phi_c$ were tested using three experiments. The block diagram representing the used hardware and connections is shown in Fig. 6.1, while the experimental setup is shown in Fig. 6.2. In the first experiment, only the frequency synchronization antenna, which is represented by the antenna M2 was moved to test the effect of $\Delta\phi_{c_1}$ on the beamforming performance or coherent gain. Afterwards, only the coherent transmitter, represented by M1, was displaced to evaluate the effect of $\Delta\phi_{c_2}$ on the coherent gain. Finally, the entire primary node (antennas M1 and M2) was displaced in the third experiment to show the effect of both of the phase shifts, which are represented by $\Delta\phi_c$.

The antennas M1 and S1 from the primary and secondary nodes respectively were used for the transmission of the coherent signals. The coherent signals were captured using T1, which was fed to an Ettus USRP X310 SDR. Antennas M1, S1, and T1 were log-periodic antennas with a frequency range of 1.35 GHz to 9.5 GHz. Standard Gain horn antennas with frequency range

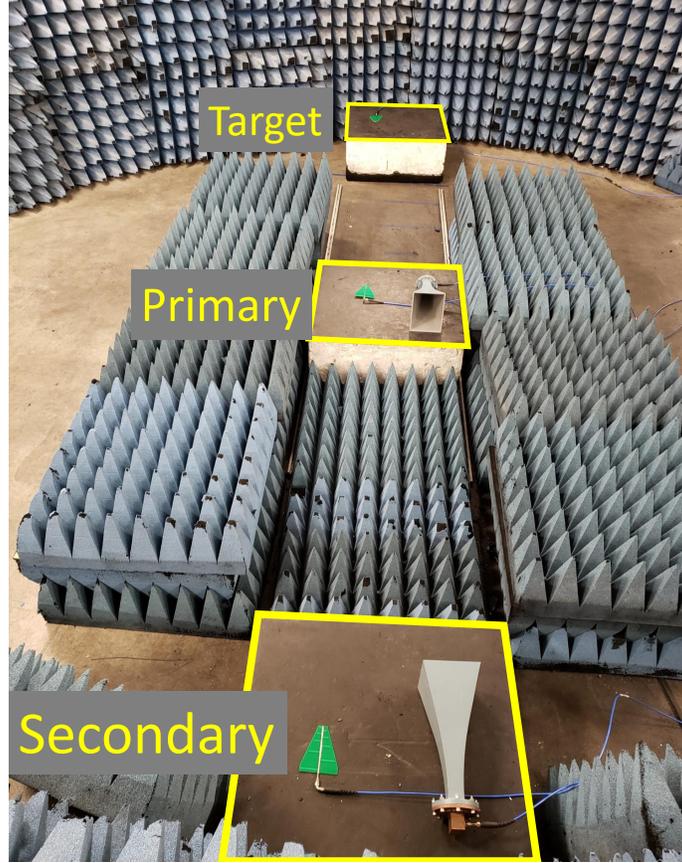


Figure 6.2: Image of the experimental setup in the semi-enclosed arch range showing the distributed two-node beamforming experiment. From [14] © 2021 IEEE.

3.85 GHz to 5.85 GHz were used for frequency synchronization, these antennas were represented by M2 and S2 in Fig. 6.1.

In the three conducted experiments, the frequencies were selected as follows: $f_{1,ref} = 3.995$ GHz, $f_{2,ref} = 4.005$ GHz, and $f_c = 1.5$ GHz. The signals transmitted at these frequencies were all CW, and a 100 kHz CW baseband signal was modulated onto the carrier f_c , which was later recovered by the receiver to evaluate the beamforming performance. The beamformed signals at 1.5 GHz had a wavelength $\lambda_c = 0.2$ m, thus, the antennas on the primary node were only moved by 0.2 m in total to observe the full effect of wireless frequency synchronization on the beamforming performance. In the experiment that is depicted in Fig. 6.2, the secondary node was placed 5 m away from the target, while the primary node was placed in between the two nodes with an initial separation of 1.7 m from the secondary node. The antennas of the primary node were moved by a total of 0.2 m

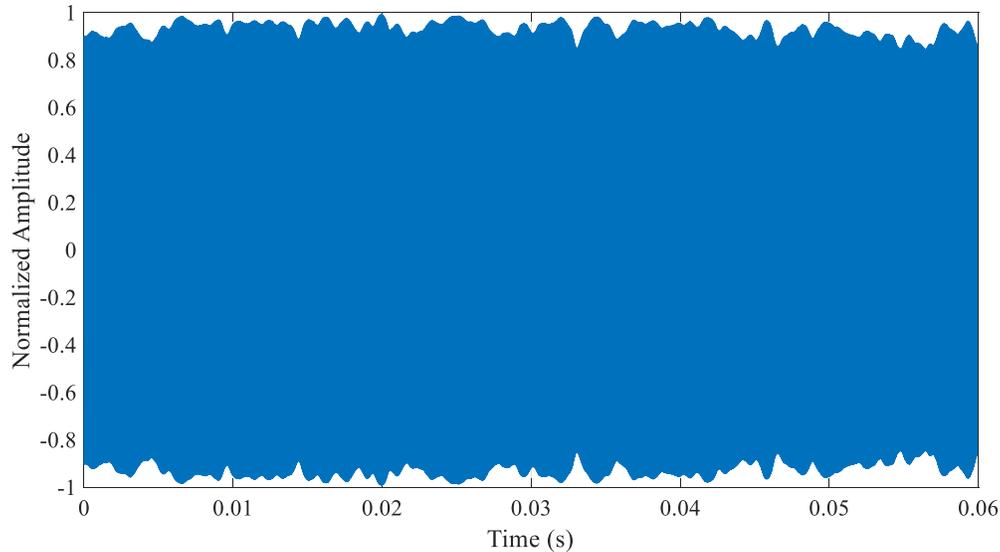


Figure 6.3: Beamformed signals from two transmitting nodes at a 1.5 GHz carrier with a 100 kHz CW baseband signal. The secondary node was wireless frequency synchronized to the primary node using the adjunct self-mixing circuit, and the received phases were manually adjusted to achieve the highest possible coherent gain at the receiver.

towards the target. The carrier was transmitted in these experiments with a 17 dBm output power and the beamforming performance was captured for every 2.5 cm increments. At first, the phases of the two received signals from the primary and secondary nodes were calibrated manually to perfectly align the phases at the starting positions. Once the phases are aligned, the wireless frequency synchronization approach allowed the received signals to add up coherently. A capture of the received signal is shown in Fig. 6.3. As can be seen, the amplitude of the received signals has some ripples and it is not constant, this is due to the instantaneous phase and frequency errors due to the wireless frequency synchronization. As explained in Chapter 3, during the synchronization of the secondary node, the phase noise of the locked VCO is affected by the phase noise of the received synchronization signals, the PLL phase noise, and the phase noise of the VCO. The combination of all this noise produce some instantaneous phase and frequency errors which are mainly dependent on the quality of the received synchronization signals and the used hardware. A better performance can be obtained if VCOs with higher phase stability were used.

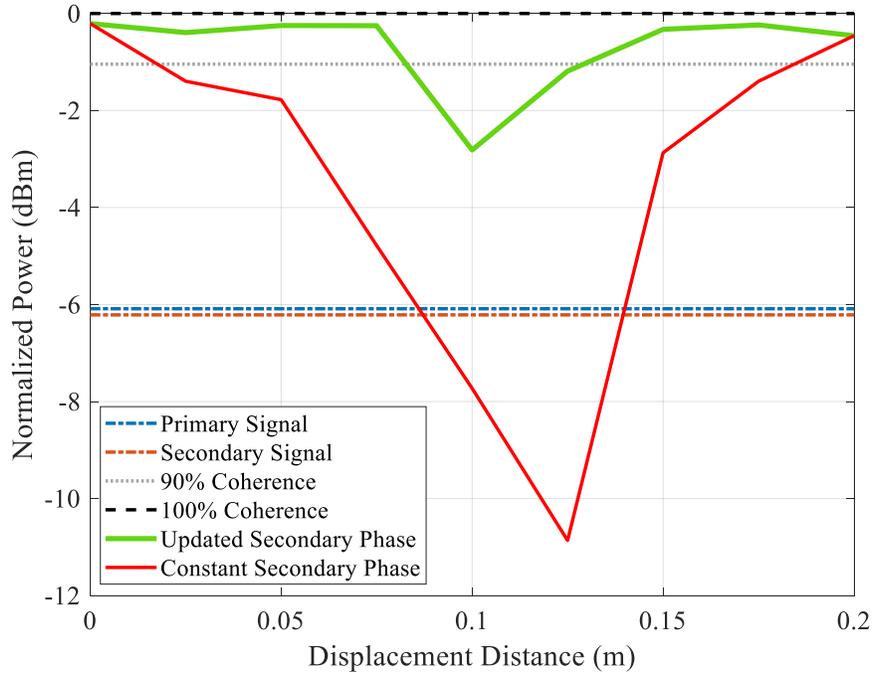


Figure 6.4: Experimental results showing the coherent summation of the primary and secondary node signals for the case where the two-tone transmitter M2 was displaced. The transmitted signals were connected to the oscilloscope via cable and the data was recorded for different displacement distances. From [14] © 2021 IEEE.

6.1.1 Frequency Locking Circuit Receiver Displacement

The first experiment aimed to prove that by using (4.19) it is possible to estimate the phase shift produced by the displacement of the frequency synchronization antennas. Thus, only the antennas S2 and M2 were displaced. The antennas T1, S1, and M1 were replaced by coaxial cable to ensure that when the primary node is moved, only the phase shift produced by the wireless frequency synchronization process is observed. M2 was moved towards the target over a total distance of 20 cm which is equal to the wavelength of the coherent signal. The beamforming performance is shown in Fig. 6.4. In this experiment, the coherent signals from the primary and secondary nodes had a similar amplitude, which stayed constant throughout the experiment since the transmitters of these two nodes were replaced by coaxial cables. A 6 dB increase in power was observed when the phases were perfectly aligned. The phase shift caused by the motion of the primary node was estimated using (4.19) and the resultant phase shift was deducted from the transmitted phase of

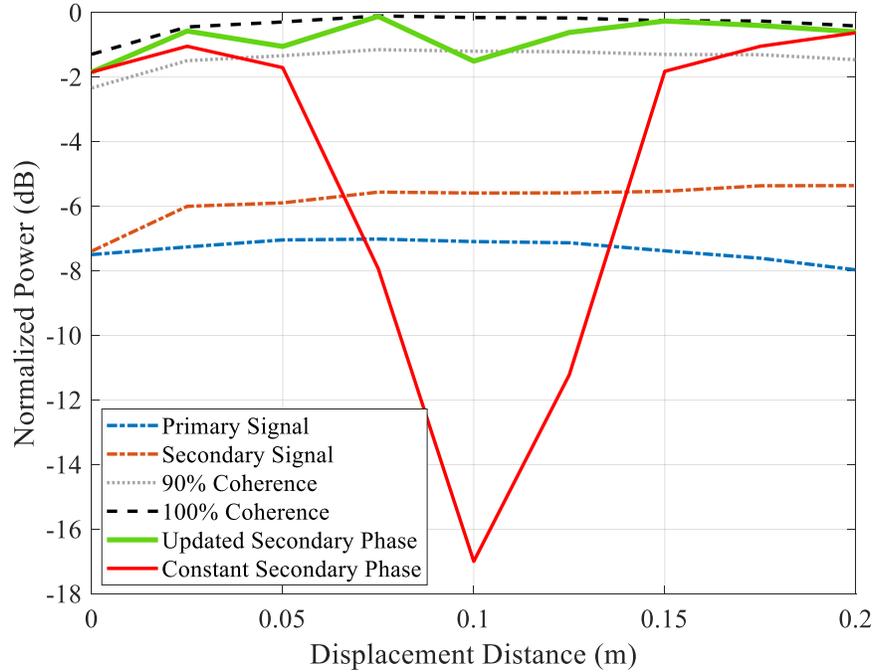


Figure 6.5: Coherent beamforming results showing the combined signal power of the two transmitted signals for multiple displacement distances of the primary node transmitter M1. From [14] © 2021 IEEE.

the secondary node to correct for the anticipated phase shift. However, when no correction was performed, the performance degraded considerably, and a 10.5 dB decrease in power was observed in the area around 0.1 m separation, which corresponds to the distance where the phase shift is close to 180° . However, the minimum power was observed at 12.5 cm displacement distance instead of 10 cm, which is a result of multipath interference. The effects of multipath interference are also visible in the case of the updated secondary phase (where the correction was done), where a degradation of the received power was observed at 10 cm.

6.1.2 Primary Transmitter Displacement

Opposite to Section 6.1.1, the phase shift produced by moving the transmitters is evaluated in this section. This was done by using log-periodic antennas for T1, S1, and M1. The phase shift produced by the motion of the synchronization antennas was suppressed by keeping S2 and M2 stationary while the M1 was displaced. The array was transmitting in the end-fire configuration,

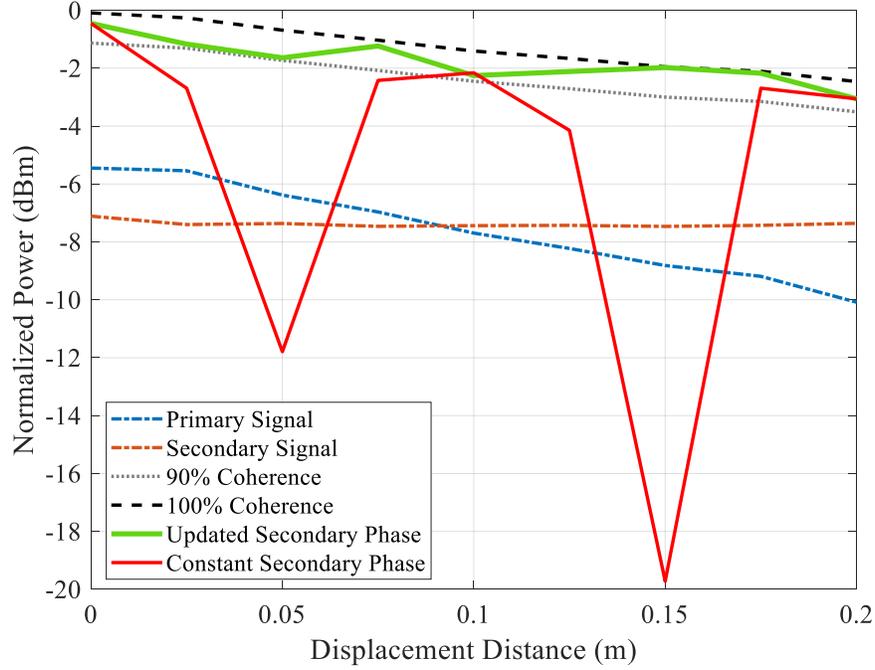


Figure 6.6: Coherent beamforming results showing the combined signal power of the two transmitted signals for multiple displacement distances of the primary node. From [14] © 2021 IEEE.

which is done by selecting $\theta_0 = 90^\circ$. The results of this experiment are shown in Fig. 6.5. As can be seen, the coherent gain was kept above 0.9 for most of the positions in the case where the phase was adjusted using (6.1). These results show that it is possible to track the phase shift produced by moving the transmitters using the conventional analysis for coherent distributed arrays in the case where the adjunct self-mixing circuit was used for frequency synchronization.

6.1.3 Full Node Displacement

Now that the phase shift produced by moving individual antennas on the primary node was evaluated, this section shows the effect of moving all the antennas on the primary node simultaneously. Similar to Sections 6.1.1 and 6.1.2, in this section the antennas of the primary node were moved by a total distance of 0.2 m, which corresponds to the wavelength of the transmitted signal. Beamforming was achieved at the beamsteering angle $\theta_0 = 90^\circ$, which corresponds to the most sensitive case scenario. The estimated phase shift from (6.2) was used to correct the phase on the secondary

node so that it is possible to obtain a coherent summation of signals at the target. As can be seen from Fig. 6.6, when the phase correction was applied, it was possible to maintain a coherent gain above 0.9 for all the recorded values. Since in this experiment both of the antennas of the primary node were displaced simultaneously, and the antennas were beamforming at $\theta_0 = 90^\circ$, two minimas were obtained for the case where no correction was applied for the phase of the secondary node. This is the case since the contribution from both terms in (6.2) was additive. This experiment proves that coherent beamforming is achievable using the presented approach once the primary node is localized by the secondary node. In this experiment, localization is achieved by only measuring the separation between the two nodes since the angle is given.

6.2 Coherent Beamforming using Frequency Locking Circuit and Ranging

In addition to the experimental setup in Section 6.1, the secondary node is equipped with a radar in this section so that the location of the primary node can be estimated in real time before applying any phase correction to the transmitted phase from the secondary node. The primary node is equipped with a repeater so that it can be seen by the secondary node as a point source. This is essential since accurate range estimates are required to estimate and compensate for the obtained phase shift, especially that two-tone waveforms (which are used for ranging in this section) are highly ambiguous.

6.2.1 Range Estimation

the distance between the two nodes was estimated using the two-tone ranging waveform. The sampling frequency was set to 25 MSps, while the ranging frequencies were selected as $f_1 = 500$ kHz and $f_2 = 4.5$ MHz and the pulse repetition interval had a duration of 1 ms with a 50% duty cycle. It should be noted that with a sampling rate of 25 MSps a larger f_2 can be chosen which would improve the ranging standard deviation. However, it was found that with upper tones close to the Nyquist frequency, distortion in the discrete matched filter output was too coarse to accurately detect small changes in position. By reducing the value of f_2 , the estimation ability is

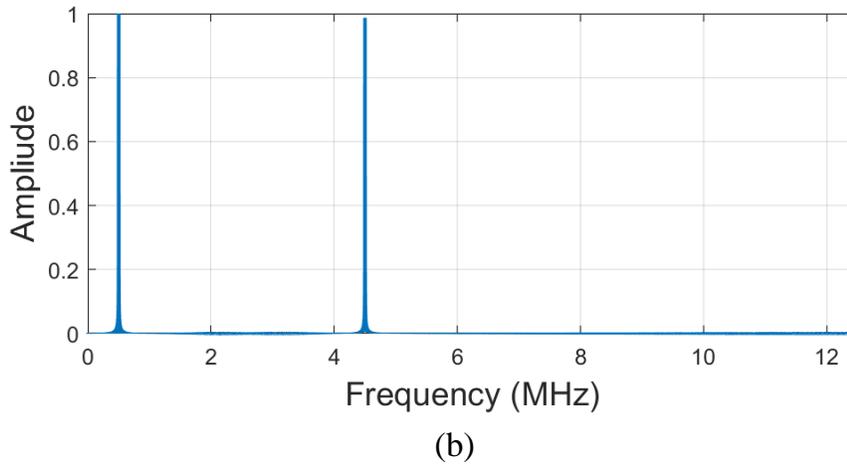
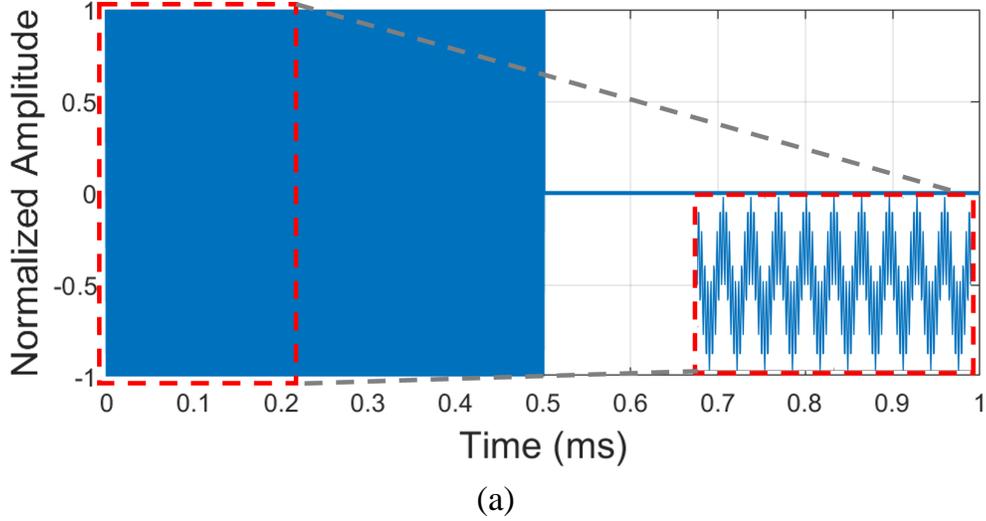


Figure 6.7: (a) Time domain representation of the localization waveform. (b) Frequency content of the localization waveform.

reduced but results in a more finely sampled representation of the matched filter and therefore the bias is reduced. The time and frequency domain of the ranging waveform is shown in Fig. 6.7.

The processing gain from using a matched filter is equivalent to the time-bandwidth product $T BW_n$, where T is equal to 0.5 ms in this experiment, and BW_n is the noise bandwidth which, since no filtering was performed, was 25 MHz. The processing gain is equivalent to 41 dB. The average preprocessing SNR was on average equal to 30 dB, giving a total post processing SNR of 71 dB. Using these values, the CRLB of the time delay estimates is equal to $\sigma_t = 9.88510^{-23} s^2$, equivalent to a lower bound for the positional RMSE of $\sigma_R = 1.5$ mm.

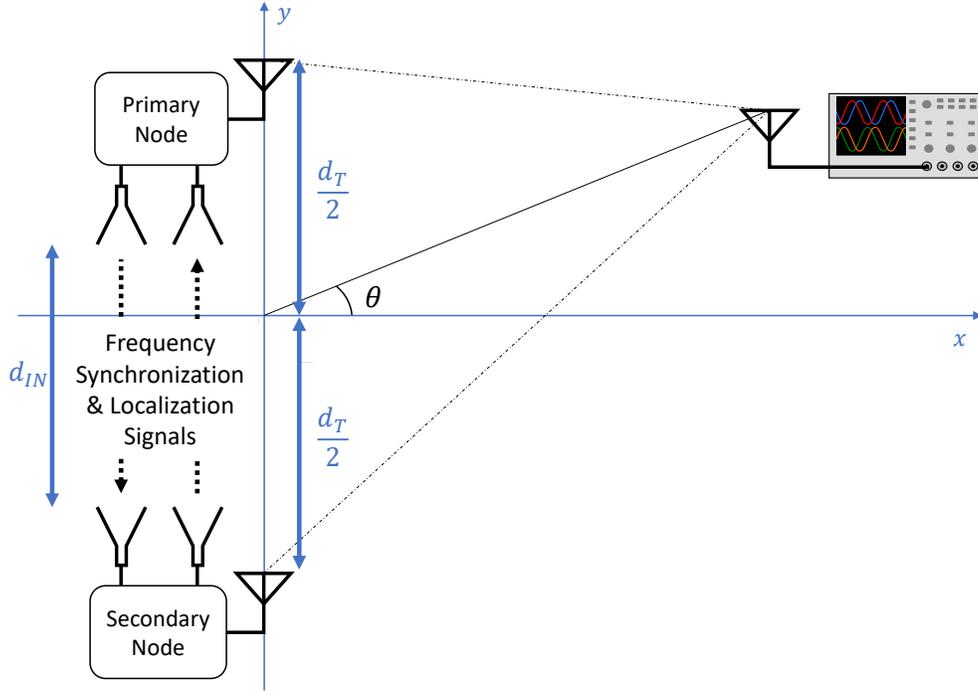


Figure 6.8: Schematic of the distributed array used in this study, consisting of a primary and secondary node, beamforming to an arbitrary angle.

To evaluate the required localization accuracy, the coherent gain G_c is evaluated parametrically as a function of the localization standard deviation. The probability to achieve a coherent gain above a certain threshold $P(G_c \geq X)$ is evaluated using 50,000 Monte Carlo simulations where the random distances Δd_{IN} and Δd_T are equal, which is the case portrayed in Fig. 6.8. Similar settings to the Monte Carlo simulations in Chapter 2 were selected. Fig. 6.9 shows the simulated results where the standard deviation of the range estimates, σ_{Δ_d} , is expressed as a function of the wavelength of the coherent signal. The threshold values $X = 0.6, 0.7, 0.8,$ and 0.9 were investigated for a probabilities $P(G_c \geq X) = 90\%$.

As can be seen, and as discussed earlier in this chapter, higher ranging accuracy is needed at $\theta_0 = 90^\circ$ to prevent the coherent gain from considerably degrading. The minimum RMSE requirement for the range estimate to achieve $P(G_c \geq 0.9) = 90\%$ at $\theta_0 = 90^\circ$ is equal to $0.03 \lambda_c$. Knowing that, the achievable coherent frequencies using these requirements are given by

$$f_c \leq \frac{0.03 c}{\sigma_x}. \quad (6.3)$$

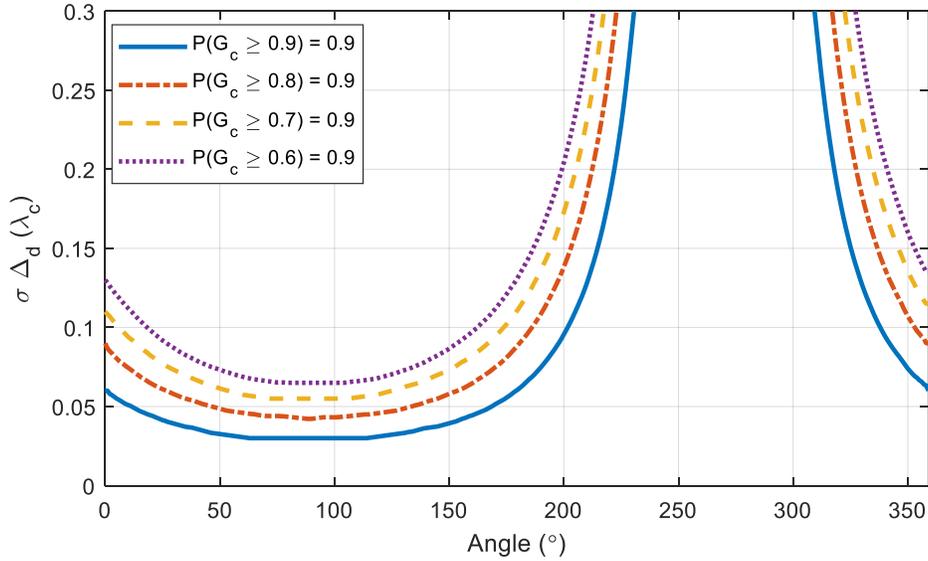


Figure 6.9: 50,000 Monte Carlo simulation showing various threshold values for achieving coherent gain. The probabilities were estimated as a function of angle and localization standard deviation. The strictest requirements for localization uncertainty appear at 90° .

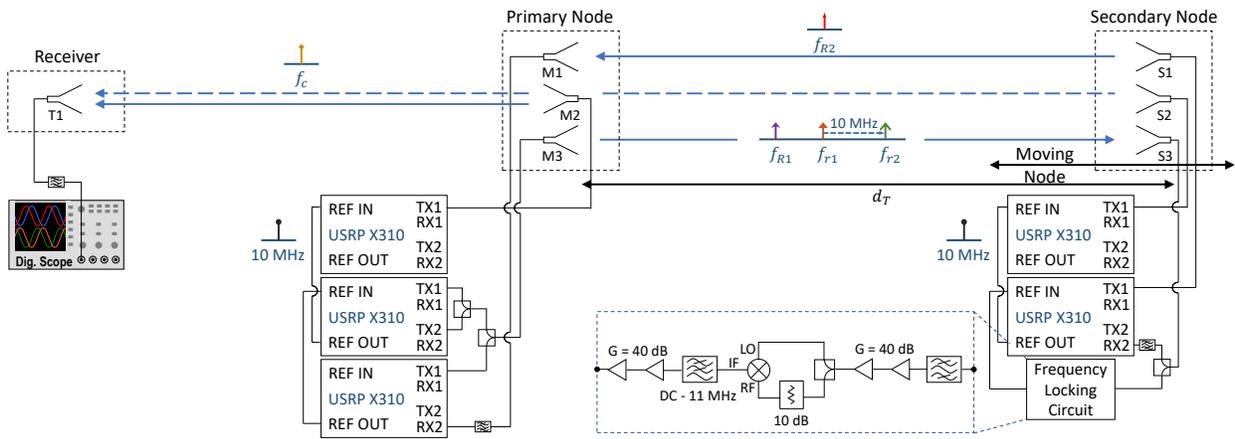


Figure 6.10: Schematics showing the connections and setup of the distributed nodes and target. This setup was used for the wireless beamforming experiment. From [15] © 2021 IEEE.

In this section, the coherent carrier frequency is selected as 1.5 GHz. Thus, to ensure that $P(G_c \geq 0.9) = 90\%$, the maximum allowed RMSE is 6 mm, which is higher than the CRLB of the selected ranging waveform and the achievable SNR in the proposed experiment (which is on average equal to 30 dB).

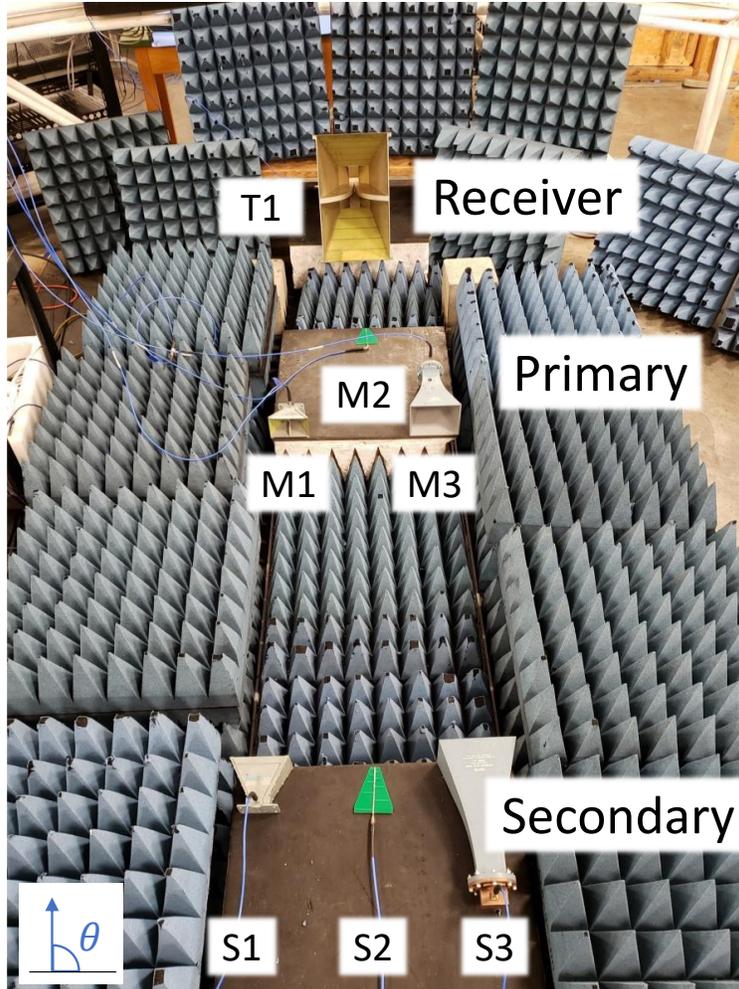


Figure 6.11: Open-loop coherent distributed array setup showing the target, primary, and secondary nodes. The antennas in this setup were connected to multiple SDRs, an Oscillator, and a frequency locking circuit. From [15] © 2021 IEEE.

6.2.2 Distributed Beamforming Experiment

To demonstrate the proposed approach, the setup described by the block diagram in Fig. 6.10 was realized. The antennas that were representing the two-node array are shown in Fig. 6.11. As can be seen, the steering angle θ_0 was set to 90° . The experiment was held in a semi-enclosed arch range to minimize the unwanted reflections. Horn antennas were used for frequency and phase synchronization while log-periodic antennas were employed for the transmission of the coherent signals. M1 and S1 had an operational frequency range of 2–18 GHz, while M3 and S3 had an operational frequency range of 3.95–5.85 GHz. The log-periodic antennas, represented by M2

and S2 had an operational frequency range of 1.35–9.5 GHz. The receiver was equipped with the horn antenna T1, which had an operational frequency of 0.5–6 GHz. The MSOX92004A Infiniium oscilloscope represented the target, which was used to monitor the received power for various scenarios.

The primary node was equipped with an active repeater that receives the ranging signals from the cooperative radar, amplifies them, and retransmits them back at a second band. The use of this repeater allows the primary node to be represented as a point source, which is essential for this experiment since an accurate range measurement is necessary to eliminate the phase shifts in the array. By transmitting back the ranging signals at a second band, it is ensured that the secondary node will be only observing the primary node. In addition, the power amplification on this repeater reduces the propagation losses to $1/R^2$ instead of $1/R^4$ as seen in traditional radars. The two carriers f_{R1} and f_{R2} that were used for ranging are at 5 GHz and 3 GHz respectively. At first, the ranging pulses were transmitted from S1 at a 3 GHz carrier. Once captured by M1, they were amplified and retransmitted back from M3 at a 5 GHz carrier. Finally, the captured ranging pulses by S3 were used to estimate the range by employing matched filtering and interpolation (1000 points were interpolated between every two samples from the matched filter output). The frequency of the secondary node was locked to the two incoming tones at f_{r1} and f_{r2} , which were selected as 4.3 GHz and 4.31 GHz respectively.

Before beamforming at the target, the system was calibrated to remove any initial static phase offset; this is a one time calibration that is needed at the system startup. Once calibrated, the coherent signal, which is represented by a 1.5 GHz CW signal, was transmitted from the two nodes to the target, represented by the oscilloscope. The oscilloscope collected the signal for two cases: with and without phase correction. The amplitude of the beamformed signals was captured for both cases with and without phase alignment, and it was compared to the amplitude of the individual signals. In both cases, frequency synchronization was achieved. The data was recorded with 2.5 cm increments, and for every location, 1,500 cycles of the beamformed signals were collected to estimate the amplitude. The secondary node was displaced in total by 20 cm,

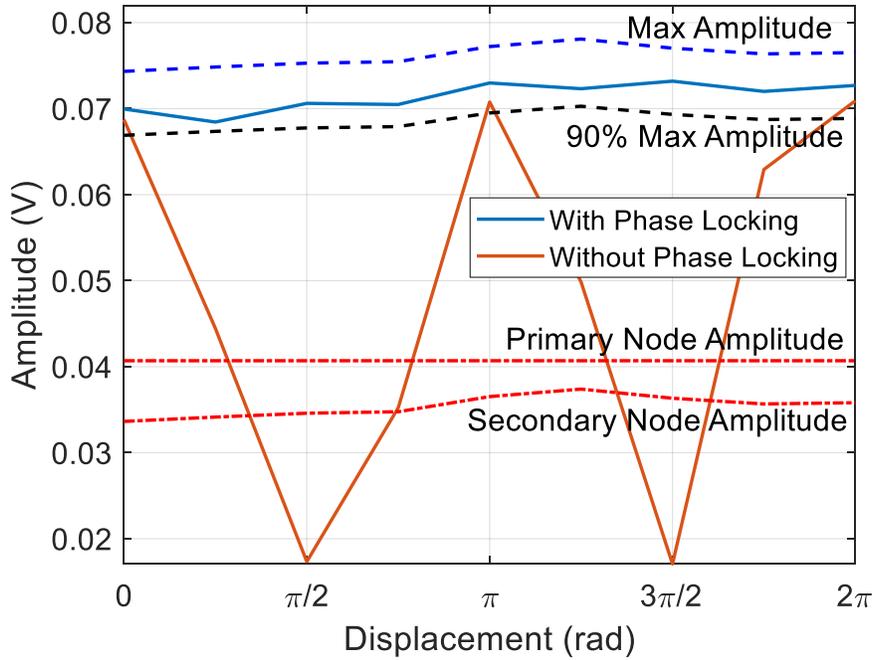


Figure 6.12: Experimental results showing the coherent beamforming results for the case where the secondary node was displaced. In this scenario, the signals were transmitted using log-periodic antennas, and their were received using a horn antenna. From [15] © 2021 IEEE.

which is equal to the wavelength of the coherent signal λ_c . The results are shown in Fig. 6.12. Similarly to the results in Section 6.1.3, when no phase correction was applied, two minimas were observed when measuring the amplitude of the beamformed signals. Whereas, the amplitude of the beamformed signals was always above 90% of the maximum achievable amplitude when phase correction was applied.

This experiment was the first to proved that an open-loop coherent distributed array is achievable. However, this experiment represents the simplified scenario where the angle between the nodes is given and only the range needs to be estimated. As a next step, localization technique(s) need to be investigated since in general the angle of the primary node is not given.

CHAPTER 7

TWO-DIMENSIONAL LOCALIZATION TECHNIQUE THAT ENABLES BEAMFORMING

It has been shown in Chapter 6 that open-loop beamforming is possible in the case where the secondary nodes only need to estimate the range of the primary node. However, in general, the primary node can be located anywhere around the secondary nodes, thus it is important in some cases to estimate the location of the primary node, rather than its range. Two-dimensional localization of the primary node is addressed in this chapter. Localization is achieved by employing the TOA estimation approach [77, 149].

7.1 TOA Using Only One Secondary Node

To synchronize the phase of every secondary node relatively to the phase of the primary node, every secondary node has to be able to localize the primary node. TDOA can be used to localize the primary node. This can be done by having the primary node transmit specific signals to all the secondary nodes at once using predefined time stamps. However, to achieve accurate localization, which is crucial to the operation of the distributed array, all the nodes need to be perfectly time-aligned. In many cases, especially when accurate localization is needed, the time alignment requirements surpass the achievable experimental results. This can be alleviated by focusing on TOA for localization rather than TDOA. TOA can be performed by first having all the secondary nodes transmit ranging signals towards the primary node, once these signals are received, they are amplified and retransmitted back at a second channel similarly to how ranging was achieved in the previous chapters. This was tested using the architecture in Fig. 7.1. When testing the localization approach, the secondary node(s) were frequency locked using coaxial cables, however, wireless frequency synchronization can be realized as explained in the earlier chapters.

The advantage of using the approach in Fig. 7.1 is that it is scalable as long as the ranging waveform is scalable. Since in this approach it is possible to choose any desired ranging wave-

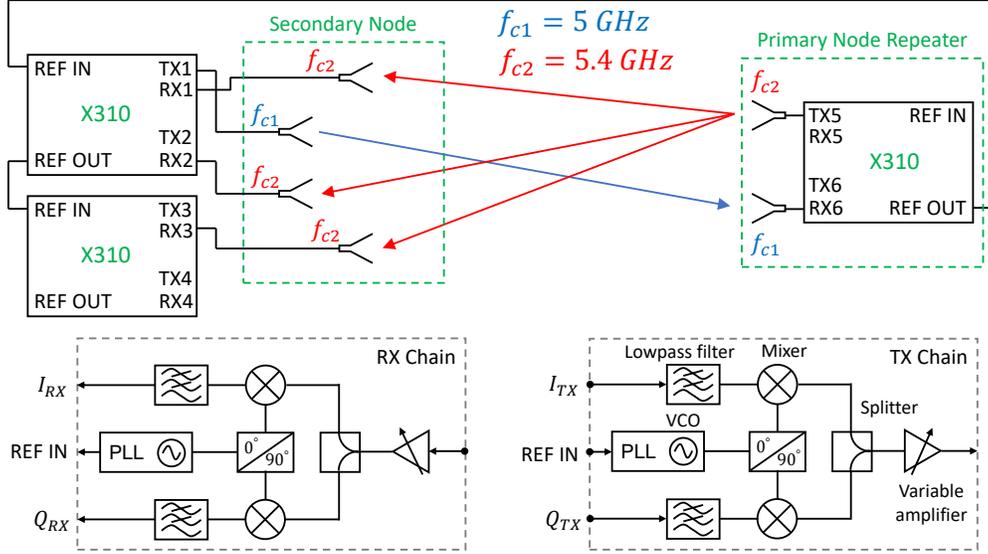


Figure 7.1: Diagram showing the proposed TOA experiment, where the secondary node localizes the primary node using TTSFW signals. From [1] © 2021 IEEE.

form, one can use TTSFW to achieve scalable, high accuracy range estimates, which is the case in this experiment. In addition, it does not require accurate time alignment between the primary and secondary node. Furthermore, in general it is better to use multilateration algorithms for localization compared to estimating the range and angle of the primary node separately. This is the case mainly because it is hard to achieve high accuracy angle estimation generally (below 0.1° error).

Regarding the multilateration approach used, the approximate maximum likelihood (AML) estimator was selected to find the location of the primary node [150]. In TOA, the time of arrival of the ranging pulse from the primary node (or the range of the primary node) at the receiving anchors (or receivers in our case) is used to form circles with radii equal to the estimated range. Optimally, the primary node would be at the point of intersection. However, due to noise, the radii have range offsets causing the circles to rarely intersect. This is resolved by employing a maximum likelihood (ML) estimator. The ML estimator evaluates the variance of the time delay estimates at every receiver and then select the best estimate for the point of intersection. As explained in [150], the AML estimator solves the following equation to determine the location of the primary node

$$2 \begin{bmatrix} \sum g_i x_i & \sum g_i y_i \\ \sum h_i x_i & \sum h_i y_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum g_i (s + k_i - \delta_i^2) \\ \sum h_i (s + k_i - \delta_i^2) \end{bmatrix}, \quad (7.1)$$

with

$$g_i = \frac{x - x_i}{r_i (r_i + \delta_i)}, \quad (7.2)$$

$$h_i = \frac{y - y_i}{r_i (r_i + \delta_i)}, \quad (7.3)$$

where (x_i, y_i) are the known locations of the receivers, (x, y) is the location of the transmitter, δ_i is the estimated distance between the target and the receiver i , r_i is the true distance between the target and the receiver i , $s = x^2 + y^2$, and $k_i = x_i^2 + y_i^2$. (7.1) is solved iteratively while being treated as set of linear equations.

The CRLB of TOA is derived in [83] and can be calculated from the inverse of the Fisher Information Matrix of the position space J_{pos} . For a 3 anchors system, J_{pos} is given by

$$J_{pos} = \sum_{i=1}^3 \begin{bmatrix} \frac{(x-x_i)^2}{\sigma_i^2 r_i^2} & \frac{(x-x_i)(y-y_i)}{\sigma_i^2 r_i^2} \\ \frac{(x-x_i)(y-y_i)}{\sigma_i^2 r_i^2} & \frac{(y-y_i)^2}{\sigma_i^2 r_i^2} \end{bmatrix}, \quad (7.4)$$

where σ_i^2 is the ranging variance at the sensor i , which can be minimized by transmitting high power signals, wideband signals, accurate ranging waveforms, and signals with long durations. Using 3 anchors allows us to localize the transmitter in 2 dimensions. The first element of the CRLB matrix which is obtained from solving (7.4) has the lower bound for the variance of x while the last element has the lower bound for the variance of y . The elements on the off diagonal represent the lower bound for the covariance of x and y . Thus by knowing the variance of the range estimates it is possible to calculate the CRLB for the localization algorithm.

7.2 Experimental Results

In the setup that is shown in Fig. 7.2, TX1 was set at the origin (0,0), and the (x_i, y_i) coordinates in meters for RX1, RX2, and RX3 were (0.345, 0.384), (-0.408, -0.114), (-0.737, 0.384). In our case, the estimated δ_i is the distance between the primary node, and the center of TX1 and RXi on the secondary node. This leads to an adjusted location (or virtual location) for the anchors, which in our case would be (0.173, 0.192), (-0.204, -0.057), (-0.368, 0.192). This experiment was achieved using Ettus Research X310 SDRs, each equipped with two UBX 160 daughterboards. Three horn

Table 7.1: Standard deviation (STD) of the estimated locations from experimental measurements versus CRLB in meters. From [1] © 2021 IEEE.

| (x,y) | STD (x, y) | $\sqrt{\text{CRLB}} (x, y)$ |
|---------------|------------------|-----------------------------|
| (0, 3.6) | (0.0495, 0.0058) | (0.048, 0.004) |
| (-0.15, 3.6) | (0.05, 0.0035) | (0.048, 0.0032) |
| (0.15, 3.6) | (0.052, 0.0076) | (0.048, 0.0055) |
| (-0.12, 3.75) | (0.051, 0.0051) | (0.05, 0.0033) |

antennas with 15 dBi gain were connected to TX1, TX5, and RX6, while RX1, RX2, and RX3 were connected to three 10 dBi gain horn antennas. The TTTSFW ranging waveform had $N = 2$, $f_1 = 500$ kHz, $BW = 9$ MHz, $T = 49.9 \mu\text{s}$, and $T_r = 24.95 \mu\text{s}$. Throughout the experiment, the average ranging standard deviations of the 3 receivers were $\sigma_1 = 6.08$ mm, $\sigma_2 = 6.1$ mm, and $\sigma_3 = 4.58$ mm respectively. For these values, the CRLBs for x and y can be computed using (7.4), and the results are shown in Fig. 7.3.

The localization CRLB values can be used to analyze the achievable coherent gain for distributed phased arrays. Ten thousand Monte Carlo simulations were evaluated for arrays of sizes $N = 2$ and 6 nodes to analyze the achievable coherent gain for a 1 GHz coherent frequency. The simulation results are shown in Fig. 7.4. The average achievable coherent gain was evaluated based on the spread area of the nodes. The maximum spread of the secondary nodes was set to ± 5 m for x and y coordinates, where the primary node was designated as the reference point and the secondary nodes were spread uniformly around the primary node. As can be seen, A lower average coherent gain is achievable when the average spacing between the primary and secondary nodes increases.

The comparison between the values in Fig. 7.3, and the experimental results is shown in Table 7.1 for four points. Since directive horn antennas were used in this experiment, the evaluated locations were all in the broadside to the antennas of the primary node.

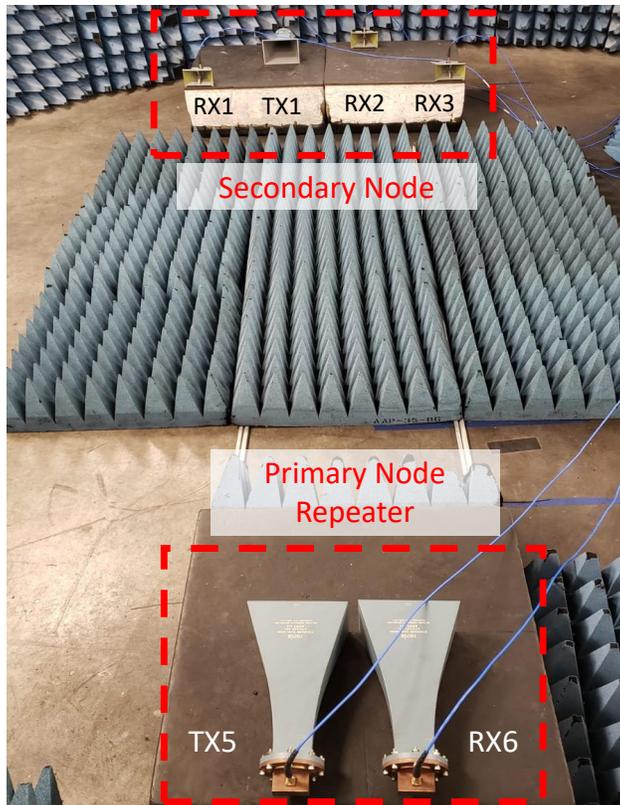


Figure 7.2: Experimental setup. From [1] © 2021 IEEE.

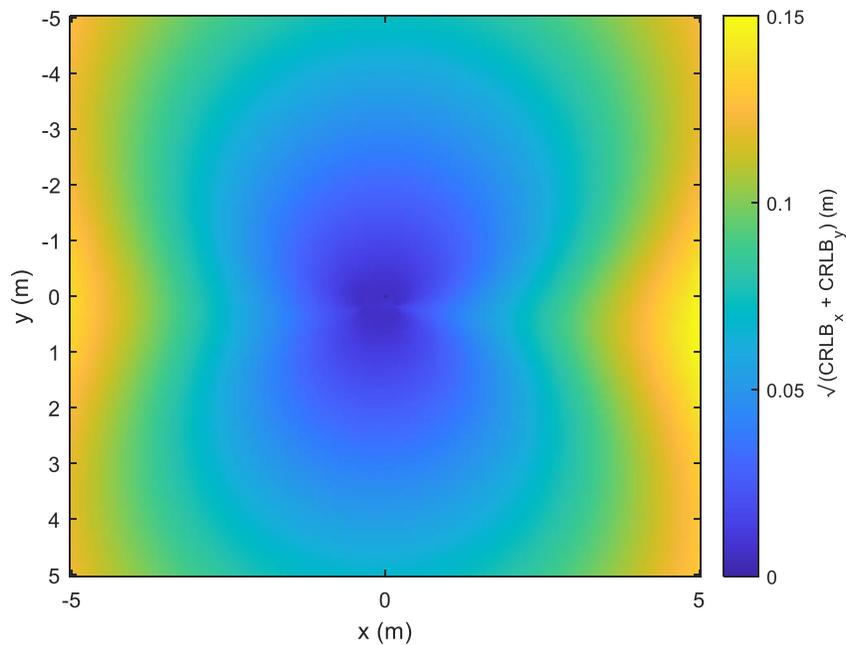


Figure 7.3: Square root of the summation of lower bounds for estimating x and y coordinates of the target at multiple locations. From [1] © 2021 IEEE.

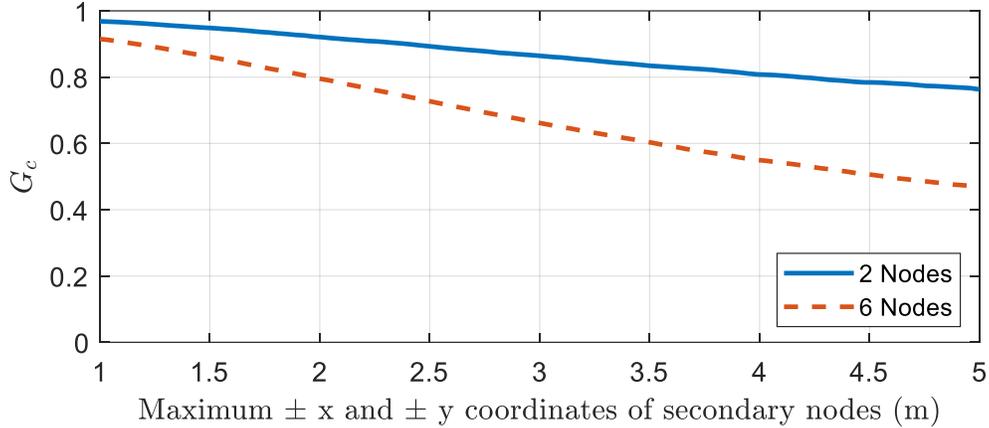


Figure 7.4: Ten thousand Monte Carlo simulations showing the average achievable coherent gain for 1 GHz coherent signals using the CRLB results for the localization estimates. Arrays of 2 and 6 nodes were analyzed. From [1] © 2021 IEEE.

7.3 Discussion

The standard deviation values in Table 7.1 are in general very low, which is desirable for the coherent operation of the distributed antenna array. However, one of the main drawbacks in the experimental setup is the use of horn antennas instead of omnidirectional antennas, such as dipole antenna. Horn antennas are directive and their phase center changes depending on the receive angle. Thus, whenever the transmit and receive antennas are not perfectly aligned, the gain drastically drops, and the location of the phase center slightly differ, which leads to biases in the localization estimates. On the other hand, omnidirectional antennas suffer from low gain, which will impact the localization accuracy and lead to an increase in the standard deviation by at least an order of magnitude. In addition, omnidirectional antennas are highly prone to multipath interference, leading to highly biased range estimates.

Throughout the localization experiment, biases on the order of centimeters were observed. These biases were not only caused by the shift in phase centers, which usually is not supposed to have a very high effect for the chosen positions of the primary node, they were also a result of the multipath interference, which is in general the major contributor to the bias in range estimates. Thus, to enable accurate, unbiased localization, multipath interference needs to be minimized. Multipath interference is not addressed in this dissertation as the targeted application for the pre-

sented wireless frequency and phase synchronization approaches is beamforming using a swarm of drones or satellites where multipath interference is minimal.

CHAPTER 8

CONCLUSION

Open-loop beamforming is an emerging and fairly unexplored area. In this work, I initially focused on determining some of the system requirements that enable open-loop operations. Since these systems require inter-node coordination, the electrical states need to be synchronized without relying on external communications. Phase alignment was one of the major focus of this work, where every node in the array needs to adjust its phase relatively to an internal reference in the array to enable coherent summation of the transmitted signals in the far field. Centralized architectures were of interest, in which the secondary nodes need to synchronize their parameters relatively to a reference, the primary node. Generally, phase alignment is possible in such architectures if all the secondary nodes can localize the primary node with localization errors orders of magnitude less than the transmitted wavelength. The requirements for range/angle estimation or localization were investigated and clearly linked to the achievable beamforming performance, using the coherent gain as a metric of interest.

In addition to the requirements for localization, I investigated the requirements for reliable frequency locking in nodes that can take advantage of PLLs. The pulse repetition intervals for frequency synchronization signals, the quality of VCOs and PLLs, and the type of transmitted signals were analyzed. The bound on the achievable coherent gain was evaluated in terms of the frequency synchronization parameters. I showed the benefits of employing continuous frequency/phase locking instead of using pulsed synchronization signals.

Once all the requirements for phase and frequency synchronization were studied, accurate range estimation techniques were explored. Two-tone ranging pulses were used for accurate range estimation. Time of flight approach was used for ranging, and a disambiguation pulse was added to disambiguate the output of the matched filter. Adaptive ranging that relies on PI controllers was introduced and tested in indoor and outdoor settings. Adaptive ranging enables high-ranging accuracies that meet the system requirements, while minimizing the spectral footprint and reducing

interference on surrounding communications bands.

I demonstrated a novel frequency synchronization technique that relies on phase locking using an adjunct self-mixing circuit and PLL. Frequency synchronization was achieved using a centralized approach, where the primary node transmit frequency synchronization signals constituted of two-tone CW signals with a tone separation of 10 MHz, and the secondary nodes lock their frequencies to the incoming signals. The 10 MHz output was demodulated using an adjunct self-mixing circuit and was fed to an internal PLL on every secondary node. The phase shift due to frequency locking was tracked and accounted for while aligning the transmitted coherent signals.

I demonstrated and implemented the first open-loop beamforming at 1.5 GHz. The experiment was implemented using one primary node and one secondary node. The secondary node was locking its frequency to that of the primary node and it was adjusting its phase depending on the range between the two nodes. Beamforming was achieved using CW signals that alleviate the need for time synchronization and it was done end fire to the array, which is the most challenging direction for beamforming as it requires the highest ranging precision. In addition, I investigated a localization technique that enables the localization of the primary node in the case where both the range and angle of the primary node are unknown.

This dissertation lays the ground work for open-loop distributed phased arrays by addressing the fundamental synchronization requirements for phase and frequency. Open-loop beamforming is an emerging area of interest as it enables the beamforming of not only communications signals, but also sensing signals, which is a luxury that conventional coherent distributed arrays do not possess. The target of this dissertation is to increase the performance or transmit power from mobile and dynamic arrays such as drone swarms or satellite constellations. Using the presented synchronization techniques, such systems are now able to perform long range communications and sensing by combining the individually transmitted power from all the elements in a coherent manner without relying on any external systems.

APPENDICES

APPENDIX A

**MATLAB RANGING REQUIREMENTS FOR COHERENT
DISTRIBUTED ARRAYS**

```

clear
% close all

c = 3e8;
f = 0.3e9;
wavelength = c / f;
Fs = f*10;
t = 0:1/Fs:1e-7;
Ovar = (wavelength * 0.1 * pi / 180)^2;
% Ovar needs to be set to zero in the case where only ranging errors are
% considered and not angle estimation errors
Rvar = (wavelength * 0.01)^2;

trials = 10000;
% number of secondary nodes (4 options):
secondaryNBR = [1 3 9 99];

% stdRanging_0 = logspace(1, 5, 50)/5e5;
% stdRanging_0 = linspace(0, 0.3, 150);
% maximum x and y coordinates:
max_xy = linspace(1, 100, 100);
prevjj = 1;
fprintf('Secondary option , and max xy option: 0, 0\n');
for ii = 1 : length(secondaryNBR)
    for jj = 1 : length(max_xy)
        fprintf(['\b', 1, 3 + max(ceil(log10(abs(ii))),1)+ ...
                max(ceil(log10(abs(jj))) + (prevjj ==length(max_xy))*(ceil(...
                log10(abs(prevjj))+1),1)), num2str(ii), ', ', ', num2str(jj), '\n']);
    end
end

```

```

prevjj = jj;
for tt = 1 : trials
    for kk = 1 : secondaryNBR(ii)

        % remove the cases where the secondary nodes are within
        % 0.5 m in x and y
        tst = rand;
        if tst < 0.5
            x_actual = unifrnd(0.5, max_xy(jj));
            y_actual = unifrnd(0.5, max_xy(jj));
        else
            x_actual = unifrnd(-max_xy(jj), -0.5);
            y_actual = unifrnd(-max_xy(jj), -0.5);
        end

        x = x_actual;
        y = y_actual;

        % transforming x and y coordinates to range and angle
        R = sqrt(x^2 + y^2);
        O = atan2d(y,x);

        % variable transformation for the variance
        Xvar = Rvar * cos(O)^2 + Ovar * R^2 * sin(O)^2;
        Yvar = Rvar * sin(O)^2 + Ovar * R^2 * cos(O)^2;
        XYcov = (Rvar - Ovar * R^2) * sin(O) * cos(O);

        x_generated = normrnd(x_actual, sqrt(Xvar), 1);
        y_generated = normrnd(y_actual, sqrt(Yvar), 1);

        sepDist_actual = sqrt((x_actual - 0)^2 + (y_actual - 0)^2);
        sepDist_generated = sqrt((x_generated - 0)^2 + (y_generated - 0)^2);
    end
end

```

```

secondaryOrientation_actual = atan(y_actual/x_actual);
secondaryOrientation_generated = atan(y_generated / ...
    x_generated);

orientAngle = 2*pi*rand;

% Uncomment in the case where no frequency locking circuit
% is considered
secondary_sig(1:length(t),kk) = sin(2*pi*f*t + 2*pi *(0 ...
    + sin(orientAngle + secondaryOrientation_actual)) * ...
    sepDist_actual / wavelength - (2*pi * (0 + ...
    sin(orientAngle + secondaryOrientation_generated))...
    * sepDist_generated / wavelength));

% Uncomment in the case where frequency locking circuit
% is considered
secondary_sig_fl(1:length(t),kk) = sin(2*pi*f*t - 2*pi ...
    * (1+sin(orientAngle+secondaryOrientation_actual)) ...
    * sepDist_actual / wavelength + (2*pi * (1 + ...
    sin(orientAngle + secondaryOrientation_generated)) ...
    * sepDist_generated / wavelength));

end

% signal from primary node:
Master_sig = sin(2*pi*f*t);
% one waveform is selected by: Master_sig(1,:)

Summ_rmsAmp(tt) = (rms(sum(secondary_sig(:,1:...
    secondaryNBR(ii)),2) + Master_sig'))^2;
Summ_rmsAmp_fl(tt) = (rms(sum(secondary_sig_fl(:,1:...
    secondaryNBR(ii)),2) + Master_sig'))^2;

```

```

end

% estimation of the beamformed signal amplitude and comparison with
% the 90% threshold
reference_rmsAmp = 0.9 * (rms((1+secondaryNBR(ii)).* Master_sig))^2;
P_90Coherence(jj , ii) = sum(Summ_rmsAmp >= ...
    reference_rmsAmp)/length(Summ_rmsAmp);
P_90Coherence_fl(jj , ii) = sum(Summ_rmsAmp_fl >= ...
    reference_rmsAmp)/length(Summ_rmsAmp_fl);

Coherence(jj , ii) = mean(Summ_rmsAmp ./ ...
    (rms((1+secondaryNBR(ii)).* Master_sig))^2);
Coherence_fl(jj , ii) = mean(Summ_rmsAmp_fl ./ ...
    (rms((1+secondaryNBR(ii)).* Master_sig))^2);

end

end

% plotting:
figure(1)
hold off
plot(max_xy, smooth(P_90Coherence(:,1)))
hold on
plot(max_xy, smooth(P_90Coherence(:,2)))
plot(max_xy, smooth(P_90Coherence(:,3)))
plot(max_xy, smooth(P_90Coherence(:,4)))
legend([int2str(secondaryNBR(1)+1) ' nodes no FLC'], ...
    [int2str(secondaryNBR(2)+1) ' nodes'], [int2str(secondaryNBR(3)+1) ...
    ' nodes'],[int2str(secondaryNBR(4)+1) ' nodes'], 'Location', 'southeast')
grid on
xlabel('Maximum \pm x and \pm y coordinates of secondary nodes (m)')
ylabel('P(G_c > 0.9)')
ylim([0, 1])

```

```

figure(2)
hold off
plot(max_xy, smooth(P_90Coherence_fl(:,1)))
hold on
plot(max_xy, smooth(P_90Coherence_fl(:,2)))
plot(max_xy, smooth(P_90Coherence_fl(:,3)))
plot(max_xy, smooth(P_90Coherence_fl(:,4)))
legend([int2str(secondaryNBR(1)+1) ' nodes'],[int2str(secondaryNBR(2)+1)...
    ' nodes'], [int2str(secondaryNBR(3)+1) ' nodes'],[int2str(...
    secondaryNBR(4)+1) ' nodes'], 'Location', 'southeast')

```

```

grid on
xlabel('Maximum \pm x and \pm y coordinates of secondary nodes (m)')
ylabel('P(G_c > 0.9)')
ylim([0, 1])

```

```

figure(3)
hold off
plot(max_xy, smooth(Coherence(:,1)))
hold on
plot(max_xy, smooth(Coherence(:,2)))
plot(max_xy, smooth(Coherence(:,3)))
plot(max_xy, smooth(Coherence(:,4)))
legend([int2str(secondaryNBR(1)+1) ' nodes no FLC'],[int2str(...
    secondaryNBR(2)+1) ' nodes'], [int2str(secondaryNBR(3)+1) ' nodes']...
    ,[int2str(secondaryNBR(4)+1) ' nodes'], 'Location', 'southeast')

```

```

grid on
xlabel('Maximum \pm x and \pm y coordinates of secondary nodes (m)')
ylabel('$\bar{G}_c$', 'Interpreter', 'latex')
ylim([0, 1])

```

```

figure(4)

```

```

hold off
plot(max_xy, smooth(Coherence_fl(:,1)))
hold on
plot(max_xy, smooth(Coherence_fl(:,2)))
plot(max_xy, smooth(Coherence_fl(:,3)))
plot(max_xy, smooth(Coherence_fl(:,4)))
legend([int2str(secondaryNBR(1)+1) ' nodes'],[int2str(secondaryNBR(2)+1)...
    ' nodes'], [int2str(secondaryNBR(3)+1) ' nodes'],[int2str(...
    secondaryNBR(4)+1) ' nodes'], 'Location', 'southeast')
grid on
xlabel('Maximum \pm x and \pm y coordinates of secondary nodes (m)')
ylabel('$\bar{G}_c$', 'Interpreter', 'latex')
ylim([0, 1])

%% testin the formul for summation of sine waves
clear
% close all

c = 3e8;
f = 0.3e9;
wavelength = c / f;
Fs = f*10;
t = 0:1/Fs:1e-7;
% OvarFactor = 0.1;
OvarFactor = [2, 1, 0.7, 0.5, 0.3, 0.2, 0.1, 0.07, 0.05, 0.03, 0.025, ...
    0.02, 0.01];

% RvarFactor = 0.01;
RvarFactor = [0.04, 0.01, 0.001];

OvarVec = (wavelength .* OvarFactor .* pi ./ 180).^2;
RvarVec = (wavelength .* RvarFactor).^2;

```

```

    trials = 5000;
% trials = 100;
% secondaryNBR = [1 3 9 99];
secondaryNBR = 9;

% stdRanging_0 = logspace(1, 5, 50)/5e5;
% stdRanging_0 = linspace(0, 0.3, 150);
max_xy = linspace(1, 100, 100);
for vv = 1 : length(RvarVec)
    Rvar = RvarVec(vv);

for uu = 1 : length(OvarVec)
    Ovar = OvarVec(uu);

    prevjj = 1;
    fprintf('Secondary option, and max xy option: 0, 0\n');
    for ii = 1 : length(secondaryNBR)
        for jj = 1 : length(max_xy)
            fprintf([ repmat('\b', 1, 3 + max(ceil(log10(abs(vv))),1)+ ...
                max(ceil(log10(abs(uu)))) + (prevjj ==length(max_xy))*...
                (ceil(log10(abs(prevjj)))+1),1)), num2str(vv), ', ', ', ...
                num2str(uu), '\n' ]);
            prevjj = jj;
            for tt = 1 : trials
                for kk = 1 : secondaryNBR(ii)

                    tst = rand;
                    if tst < 0.5
                        x_actual = unifrnd(0.5, max_xy(jj));
                        y_actual = unifrnd(0.5, max_xy(jj));
                    else

```

```

        x_actual = unifrnd(-max_xy(jj), -0.5);
        y_actual = unifrnd(-max_xy(jj), -0.5);
    end

    x = x_actual;
    y = y_actual;

    O = atan2d(y,x);
    R = sqrt(x^2 + y^2);

    Xvar = Rvar * cos(O)^2 + Ovar * R^2 * sin(O)^2;
    Yvar = Rvar * sin(O)^2 + Ovar * R^2 * cos(O)^2;
    XYcov = (Rvar - Ovar * R^2) * sin(O) * cos(O);

    x_generated = normrnd(x_actual, sqrt(Xvar), 1);
    y_generated = normrnd(y_actual, sqrt(Yvar), 1);

    sepDist_actual = sqrt((x_actual - 0)^2 + (y_actual - 0)^2);
    sepDist_generated = sqrt((x_generated - 0)^2 + ...
        (y_generated - 0)^2);

    secondaryOrientation_actual = atan(y_actual/x_actual);
    secondaryOrientation_generated = ...
        atan(y_generated/x_generated);

    orientAngle = 2*pi*rand;

    % Uncomment in the case where no frequency locking
    % circuit is considered
    secondary_sig(1:length(t),kk) = sin(2*pi*f*t + 2*pi * ...
        (0 + sin(orientAngle + secondaryOrientation_actual))...
        * sepDist_actual / wavelength - (2*pi * (0 + sin(...

```

```

orientAngle + secondaryOrientation_generated)) * ...
sepDist_generated / wavelength));

% Uncomment in the case where frequency locking circuit is considered
secondary_sig_fl(1:length(t),kk) = sin(2*pi*f*t ...
    - 2*pi * (1 + sin(orientAngle + ...
secondaryOrientation_actual)) * sepDist_actual / ...
wavelength + (2*pi * (1 + sin(orientAngle + ...
secondaryOrientation_generated)) * sepDist_generated ...
/ wavelength));
end

Master_sig = sin(2*pi*f*t);
% one waveform is selected by: Master_sig(1,:)
Summ_rmsAmp(tt) = (rms(sum(secondary_sig(:,1:...
secondaryNBR(ii)),2) + Master_sig'))^2;
Summ_rmsAmp_fl(tt) = (rms(sum(secondary_sig_fl(:,1:...
secondaryNBR(ii)),2) + Master_sig'))^2;

end

reference_rmsAmp = 0.9 * (rms((1+secondaryNBR(ii)).*...
Master_sig))^2; % 90% coherence gain
P_90Coherence(jj, ii) = sum(Summ_rmsAmp >= ...
reference_rmsAmp) / length(Summ_rmsAmp);
P_90Coherence_fl(jj, ii) = sum(Summ_rmsAmp_fl >= ...
reference_rmsAmp) / length(Summ_rmsAmp_fl);

Coherence(jj, ii) = mean(Summ_rmsAmp ./ ...
(rms((1+secondaryNBR(ii)).* Master_sig))^2);
Coherence_fl(jj, ii) = mean(Summ_rmsAmp_fl ./ ...
(rms((1+secondaryNBR(ii)).* Master_sig))^2);

```

```

        end
    end
    loc = find(smooth(P_90Coherence_fl(:,1))>0.9 == 1, 1, 'last');
    if isempty(loc)
        loc = 0;
    end
    locationOf90(vv, uu) = loc;

end

end

locationOf90y = locationOf90;
locationOf90(locationOf90==100)=100.1;

figure(5)
hold off
semilogy(locationOf90(1,:), (OvarFactor),'LineWidth',2)
hold on

semilogy(locationOf90(2,:), (OvarFactor),'LineWidth',2)
semilogy(locationOf90(3,:), (OvarFactor),'LineWidth',2)
legend( ...
    ['$\sigma_R$ = ' num2str(RvarFactor(1)) '$\lambda$'], ['$\sigma_R$ = ' ...
    num2str(RvarFactor(2)) '$\lambda$'], ['$\sigma_R$ = ' num2str(...
    RvarFactor(3)) '$\lambda$'], 'Interpreter','latex', 'Location', 'northeast')
grid on
xlabel('Maximum \pm x and \pm y coordinates of secondary nodes (m)')
ylabel('$\sigma_{\theta} (\circ) \cdot \lambda$', 'Interpreter','latex')
ylim([min(OvarFactor) max(OvarFactor)])
xlim([min(max_xy) max(max_xy)])

```

APPENDIX B

MATLAB FREQUENCY SYNCHRONIZATION REQUIREMENTS FOR NODES WITH PLL

```
clear

trials = 1000;
c = 3e8;
carriers = [1e9, 5e9, 10e9];
% carriers = [0.001e9, 0.005e9, 0.02e9, 0.05e9, 0.08e9, 0.1e9 0.5e9, ...
              % 1e9, 3e9, 6e9, 9e9, 12e9, 15e9, 16e9, 17e9];
% carriers = [3750e9,2500e9, 1000e9, 500e9, 200e9,100e9, 60e9, 25e9,...
              % 15e9, 5e9, 1e9, 0.5e9];
wavelength_car = c ./ carriers;

updateRateVec = logspace(-2, 2, 40);

% Choose one of the following:
% Phase noise data for VCO A
PN_freq = [10, 100, 1e3, 10e3, 100e3, 50e6];
PN_dB_Hz = [-60, -90, -120, -145, -155, -155];
% Phase noise data from VCO B
PN_freq = [10, 100, 1e3, 10e3, 100e3, 50e6];
PN_dB_Hz = [-110, -140, -165, -185, -190, -190];
% Phase noise data ffor case 3: VCOs A and B
PN_freq = [10, 45, 1e3, 2250, 10e3, 100e3, 50e6];
PN_dB_Hz = [-110, -130, -130, -130, -145, -155, -155];

% Jitter calculation
A1 = 10^(((PN_dB_Hz(2) + PN_dB_Hz(1))/2 + 10 * log10(PN_freq(2)...
           - PN_freq(1)))/10);
A2 = 10^(((PN_dB_Hz(3) + PN_dB_Hz(1))/2 + 10 * log10(PN_freq(3)...
```

```

    - PN_freq(2))/10);
A3 = 10^(((PN_dB_Hz(4) + PN_dB_Hz(3))/2 + 10 * log10(PN_freq(4)...
    - PN_freq(3)))/10);
A4 = 10^(((PN_dB_Hz(5) + PN_dB_Hz(4))/2 + 10 * log10(PN_freq(5)...
    - PN_freq(4)))/10);
A5 = 10^(((PN_dB_Hz(6) + PN_dB_Hz(5))/2 + 10 * log10(PN_freq(6)...
    - PN_freq(5)))/10);

A = 10 * log10( A1 + A2 + A3 + A4 + A5);

RMS_phase_jitter_rad = sqrt(2*10^(A/10));
RMS_phase_jitter_sec = sqrt(2*10^(A/10))/(2 * pi * f0);

% Number of Nodes:
nodesNumberVect = [2, 10, 20, 100];
% nodesNumberVect = [2, 100];

%% without time errors

for p = 1 : trials
    for i = 1 : length(carriers)
        for j = 1 : length(updateRateVec)
            for k = 1 : length(nodesNumberVect)

                nodesNumber = nodesNumberVect(k);

                updateRate = updateRateVec(j);
                carrier = carriers(i);
                wavelength_car = c ./ carrier;

                updateTime = 1 ./ updateRate;

                % Sampling rate

```

```

Fs = carrier * 100;

% pulseWidth = 1/(100e6); % for a 100 MHz bandwidth
pulseWidth = 100/carrier;

t = updateTime - pulseWidth : 1/Fs : updateTime;
samples = length(t);

% Allan Deviation data (modified based on oscillator)
tau = 1;
AllanDev = 1.00e-10; % s
% TDEV data (modified based on oscillator)
tau_t = 0.5;
TDEV = 8e-11;

freqRMS_tau = AllanDev * f0;
freqOffsetSTD = freqRMS_tau * (updateTime / tau);%@ refresh rate
freqOffset = normrnd(0, freqOffsetSTD, [1, nodesNumber]);

signals = zeros(nodesNumber, samples);

for kk = 1 : nodesNumber

    % Random separation distance and phase error
    sepDist = 10*pi*randn;
    phaseError = normrnd(0, RMS_phase_jitter_rad, [1, samples]);

    % Phase offset due to frequency drift
    phi_0 = 2 * pi * (f0* updateTime + 0.5 * updateTime ...
        * freqOffset(kk)) - 2 * pi * (f0* updateTime ...
        + updateTime * freqOffset(kk));

    % For continuous waves:

```

```

% signals(k, 1:samples) = exp( (carrier/f0)...
    % .* (1j * 2*pi*f0*t + 1j .* phaseError));

% For pulsed waves:
signals(kk, 1:samples) = exp( (carrier/f0) .* (1j * 2*pi...
    *(f0 + freqOffset(kk)).*t + 1j .* (phi_0 + phaseError)));
end

% Signals summation
beamformedSignals = sum(signals,1);
idealBeamforming = nodesNumber * signals(1,:);

% Coherent gain
gainCoherence(i, j, k, p) = (rms(beamformedSignals)^2)...
    /(rms(idealBeamforming)^2);

end

end

end

end

% Average value for all the coherent gain data (depending on time and
% number of nodes)
finalGainCoherence = mean(gainCoherence(:,:,:),4);

%% With timing errors

for p = 1 : trials
    for i = 1 : length(carriers)
        for j = 1 : length(updateRateVec)

```

```

for k = 1 : length(nodesNumberVect)

nodesNumber = nodesNumberVect(k);

updateRate = updateRateVec(j);
carrier = carriers(i);
wavelength_car = c ./ carrier;

updateTime = 1 ./ updateRate;

Fs = carrier * 100;

pulseWidth = 1/(100e6); % for a 100 MHz bandwidth
pulseWidth = 100/carrier;

t = updateTime - pulseWidth : 1/Fs : updateTime;
samples = length(t);

% Allan Deviation data
tau = 1;
AllanDev = 1.00e-10; % s
% TDEV data
tau_t = 0.5;
TDEV = 8e-11;

freqRMS_tau = AllanDev * f0;
freqOffsetSTD = freqRMS_tau * (updateTime / tau);%@ refresh rate
freqOffset = normrnd(0, freqOffsetSTD, [1, nodesNumber]);

timeOffsetSTD_updateTime = TDEV * (updateTime / tau);
timeOffset=normrnd(0, timeOffsetSTD_updateTime, [1, nodesNumber]);

signals_TO = zeros(nodesNumber, samples);

```

```

signals_wTimeOffset = zeros(nodesNumber, samples);

% Modulation
b = [1 0 1 0 1 0 1 0 1 0];
bitTime = pulseWidth/10;

for kk = 1 : nodesNumber

    sepDist = 10*pi*randn;
    phaseError = normrnd(0,RMS_phase_jitter_rad,[1,samples]);

    timeError = normrnd(timeOffset(kk),RMS_phase_jitter_sec,...
        [1,samples]);
    timeErrorIntegrated = [];
    ttt = 1;
    while ttt <= length(t)
        if ttt <50
            timeErrorIntegrated(ttt) = 0;
        elseif ttt >length(t)-100
            timeErrorIntegrated(ttt) = 1;
        else
            if abs(rem((floor((t(ttt) + timeError(ttt)- t(1))...
                ./bitTime)),2)) ~= abs(rem((floor((t(ttt-1) +...
                    timeError(ttt-1)- t(1))./bitTime)),2))
                timeErrorIntegrated(ttt) = abs(rem((floor((t(ttt)...
                    + timeError(ttt)- t(1))./bitTime)),2));
                timeErrorIntegrated(ttt+1:ttt+50) = ones(1,50).*...
                abs(rem((floor((t(ttt) + timeError(ttt)- t(1))./bitTime)),2));
                ttt=ttt+50;
            else
                timeErrorIntegrated(ttt) = abs(rem(...
                (floor((t(ttt) + timeError(ttt)- t(1))./bitTime)),2));
            end
        end
    end
end

```

```

        end
    end
    ttt = ttt+1;
end

    phi_0 = 2 * pi * (f0* updateTime + 0.5 * updateTime * ...
freqOffset(kk)) - 2 * pi * (f0* updateTime + updateTime * freqOffset(kk));

    signals_TO(kk, 1:samples) = (2*timeErrorIntegrated -1).*...
        cos( (carrier/f0) .* (1 * 2*pi*(f0 + freqOffset(kk))...
        .*(t) + 1 .* phaseError + phi_0));

end

    beamformedSignals_TO = sum(signals_TO ,1);
    idealBeamforming_TO = nodesNumber * signals_TO(1,:);

    gainCoherence_TO(i, j, k, p) = (rms(beamformedSignals_TO)^2)...
        /(rms(idealBeamforming_TO)^2);

end

end

end

end

finalGainCoherence_TO = mean(gainCoherence_TO (:, :, :, :), 4);

```

APPENDIX C
MATLAB PEAK ESTIMATORS FOR RANGING

```
clear

F1 = [];
F2 = [];
actDistVec = [];
errors = [];
delay_ex_r = [];
delay_ex_t = [];
errors_sinc = [];
delay_ex_r_sinc = [];

InterSampPts = 20;
FreqPts = 100;

Fs = 10e6; % sampling rate
c = 3e8; % speed of light
% Tb = ceil(0.9*6000); % to control the pulse width
Tb = ceil(0.25*3992); % to control the pulse width
PW = (Tb-1)/Fs; % pulse width
t=0:1/Fs:2*(Tb-1)/Fs; % time vector (twice the PW)

f2 = 20e3; % second tone
f1_0 = linspace(f2*3, Fs/2 - f2*3, FreqPts); % list of first tone

for jj = 1 : FreqPts
    %% controlling the frequency (for two-tone or the bandwidth of LFM)

    f1 = Fs/(4); % this makes the bandwidth constant
```

```

f1 = f1_0(jj); % this makes the bandwidth variable

for k = 1 : InterSampPts
    % k % just for debugging purposes

%% phase constant for TX and RX signals
phi_init = 2*pi*rand;
phi_init = 0; % setting the TX phase to zero
phi_init_off = rand*360;
phi = phi_init + phi_init_off*(pi/180);
% phi = phi_init; % setting the RX phase equal to the TX phase

%% controlling the RX delay either by sleecting the distance or time
% %%% for distance:
% actDist = 1;
% tau = 2 * actDist / c;
% %%% for time:
% tau = (0/100)/Fs; % this makes the delay fixed
tau = ((k-1)/(InterSampPts-1))/Fs; % this makes the delay change...
    %every loop (where every 100 itteration we reach the second discr.)
actDist = (tau * c) / 2; % to calculate the distance from time

%% controlling the noise in the RX signals
noiseLeveldBW = 0; % currently I am not using the noise level
noise = wgn(1, length(t), noiseLeveldBW);
% noise = zeros(1, length(t)); % setting noise to zero for now

%% TX and RX waveforms: uncomment one of the three options
% %%% Sine pulse:
% TxTTsig = (heaviside(t - 0) - heaviside(t - PW - 0)) .* ...
%     (exp(1j * 2*pi*f1*(t - 0) + phi_init));
% RxTTsig = (heaviside(t - tau) - heaviside(t - PW - tau)) .* ...
%     (exp(1j * 2*pi*f1*(t - tau) + phi)) + noise;

```

```

% %%% Two-tone (TT) pulse:
TxTTsig = (heaviside(t - 0) - heaviside(t - PW - 0)) .* (exp(1j ...
* 2*pi*f1*(t - 0) + phi_init) + exp(1j * 2*pi*f2*(t - 0) + phi_init));
RxTTsig = (heaviside(t - tau) - heaviside(t - PW - tau)) .* (exp(1j ...
* 2*pi*f1*(t - tau) + phi) + exp(1j * 2*pi*f2*(t - tau) + phi)) + noise;

% %%% LFM pulse:
% TxTTsig = (heaviside(t - 0) - heaviside(t - PW - 0)) .* (exp(1j * ...
% 2*pi*(((f1 - f2)/PW)/2)*(t - 0).^2 + f2*(t - 0)) + phi_init));
% RxTTsig = (heaviside(t - tau) - heaviside(t - PW - tau)) .* (exp(1j ...
% 2*pi*(((f1 - f2)/PW)/2)*(t - tau).^2 + f2*(t - tau)) + phi) + noise;

% %%% LFM TT pulse:
% TxTTsig = (heaviside(t - 0) - heaviside(t - PW - 0)) .* (exp(1j * ...
% 2*pi*(((f1 - f2)/2)/PW)/2)*(t - 0).^2 + f2*(t - 0)) + phi_init)...
% + exp(1j * 2*pi*(((f1 - f2)/2)/PW)/2)*(t - 0).^2 + (f2 ...
% + (f1 - f2)/2)*(t - 0)) + phi_init));
% RxTTsig = (heaviside(t - tau) - heaviside(t - PW - tau)) .* (exp(1j ...
% 2*pi*(((f1 - f2)/PW)/2)*(t - tau).^2 + f2*(t - tau)) + phi) + ...
% exp(1j * 2*pi*(((f1 - f2)/2)/PW)/2)*(t - tau).^2 + (f2 + (f1 ...
% - f2)/2)*(t - tau)) + phi) + noise;

%% Matched filtering and interpolation

Ctt = abs(xcorr((RxTTsig),(TxTTsig)));
nRes_d = 0.001;
nCd = interp1(1:numel(Ctt),Ctt,1:nRes_d:numel(Ctt),'spline');
% %%% here I am cheating by selecting the main lobe (this is the case
% when we have a disambiguation pulse)
y = [-round(1/nRes_d): round(1/nRes_d)]';
[~,v]=max(nCd(ceil(numel(nCd)/2) + round(tau*Fs/nRes_d) + y));
match_nl_d = y(v) + ceil(numel(nCd)/2) + round(tau*Fs/nRes_d);

```

```

% [~, match_nl_d] = max(abs(nCd));
% peak location from interpolation:
actual_match_tt = (match_nl_d - ceil(numel(nCd)/2))*nRes_d;

%% NL-LS

L0 = []; L1 = []; L2 = [];
% peak = round(1+match_nl_d * nRes_d); % extracting peak from
% interpolation results (better use the next command, as if we had a
% disambiguation pulse

x = [-1: 1]';

[~,v]=max(Ctt(ceil(numel(Ctt)/2) + round(tau*Fs) + x));
peak = x(v) + ceil(numel(Ctt)/2) + round(tau*Fs);

L0(1) = Ctt(peak);
L1(1) = 0.1;
L2(1) = 2*abs(f2-f1)/Fs;

y = abs(Ctt(peak + x));

for m = 1:40
    arg = L2(m) .* (x - L1(m));

    func1 = sinc(arg);
    func2 = L0(m) .* (sinc(arg) - cos(pi .* arg)) ./ (x - L1(m));
    func3 = L0(m) .* (cos(pi .* arg) - sinc(arg)) ./ L2(m);
    J = [func1, func2, func3];

    func = L0(m) .* sinc(arg);

    Delta_y = y' - func;

```

```

Delta_L = inv(J' * J) * J' * Delta_y;

L0(m + 1) = L0(m) + Delta_L(1);
L1(m + 1) = min(0.6, max(-0.6, L1(m) + Delta_L(2)));
L2(m + 1) = L2(m) + Delta_L(3);
% L2(m + 1) = L2(m);

end

new_peak = peak + L1(m + 1);

actual_match_tt_sinc = new_peak - ceil(numel(Ctt)/2);

%% MF-LS

dt = round(tau*Fs)/Fs;

i_itt = 50; % how many MF to try
for i = 1:i_itt

    w2(i) = dt - 1/(2*abs(f2-f1)) + (i/i_itt)/(abs(f2-f1));
    t2=0 - w2(i) : 1/Fs : 2*(Tb-1)/Fs - w2(i); % time vector (2*PW)

    RxTTsig2 = (heaviside(t2) - heaviside(t2 - PW)) .* ...
        (exp(1j * 2*pi*f1*(t2) + phi) + exp(1j * 2*pi*f2*(t2) + phi));
    Ctt2 = abs(xcorr((RxTTsig2),(TxTTsig)));
%     [new_peak_val(i), new_peak_loc(i)] = max((Ctt2.*Ctt));
    [new_peak_val(i), new_peak_loc(i)] = min(sum((Ctt2./max(Ctt2)...
        - Ctt./max(Ctt)).^2));

% figure;plot(real(RxTTsig));hold on;plot(real(RxTTsig2));grid on

```

```

    % figure; plot(Ctt/max(Ctt)); hold on; plot(Ctt2/max(Ctt2)); grid on
end

% Methods for refining the peak estimation for MF-LS:
% Method 1: without any fitting
% [~, rrr] = min(new_peak_val);
% delay_ex_t_F(jj, k) = w2(rrr);

% Method 2: using interpolation
nRes_d2 = 0.001;
new_peak_val2 = interp1(1:numel(new_peak_val), new_peak_val, 1:nRes_d2...
    :numel(new_peak_val), 'spline');

[~, v2] = min(new_peak_val2);
match_n1_d2 = (v2 + 1/nRes_d2 - 1)*nRes_d2;
delay_ex_t_F(jj, k) = dt - 1/(2*abs(f2-f1)) + (match_n1_d2/i_itt)...
    /(abs(f2-f1)); % peak location from interpolation

% Method 3: using sinc fitting
L20 = []; L21 = []; L22 = [];
% peak = round(1+match_n1_d * nRes_d); % extracting peak from
% interpolation results (better use the next command, as if we had a
% disambiguation pulse)

x2 = [-1: 1]';
new_peak_val2 = - new_peak_val + 2*max(new_peak_val);
[peak_val, peak_loc] = max(new_peak_val2);

L20(1) = peak_val;
L21(1) = 0.1;
L22(1) = 2*abs(f2-f1)/Fs;

y = (new_peak_val2(peak_loc + x2));

```

```

for m = 1:40
    arg = L22(m) .* (x2 - L21(m));

    func1 = sinc(arg);
    func2 = L20(m) .* (sinc(arg) - cos(pi .* arg)) ./ (x2 - L21(m));
    func3 = L20(m) .* (cos(pi .* arg) - sinc(arg)) ./ L22(m);
    J = [func1, func2, func3];

    func = L20(m) .* sinc(arg);

    Delta_y = y' - func;

    Delta_L = inv(J' * J) * J' * Delta_y;

    L20(m + 1) = L20(m) + Delta_L(1);
    L21(m + 1) = min(1, max(-1, L21(m) + Delta_L(2)));
    L22(m + 1) = L22(m) + Delta_L(3);
    % L2(m + 1) = L2(m);

end

new_peak2 = peak_loc + L21(m + 1);

delay_ex_t_F2(jj, k) = dt - 1/(2*abs(f2-f1)) + (new_peak2/i_itt)...
    /(abs(f2-f1));

%% Logging the results
delay_ex_t(jj, k) = actual_match_tt / Fs;
delay_ex_r(jj, k) = delay_ex_t(jj, k) * c / 2;
delay_ex_r_sinc(jj, k) = actual_match_tt_sinc / Fs * c / 2;
delay_ex_r_F(jj, k) = delay_ex_t_F(jj, k) * c / 2;
delay_ex_r_F2(jj, k) = delay_ex_t_F2(jj, k) * c / 2;

```

```

F1(jj , k) = f1; F2(jj , k) = f2;
actDistVec(jj , k) = actDist;
errors(jj , k) = delay_ex_r(jj , k) - actDist;
errors_sinc(jj , k) = delay_ex_r_sinc(jj , k) - actDist;
errors_F(jj , k) = delay_ex_r_F(jj , k) - actDist;
errors_F2(jj , k) = delay_ex_r_F2(jj , k) - actDist;

end

end

%%% Final results

% %%% Peak Estimation Using Interpolation
figure(1)
mesh(F1-F2,actDistVec .* 2 ./ c, errors)
xlabel('Bandwidth (Hz)')
ylabel('Time Received (s)')
zlabel('Offset (m)')
title('TT - Peak Estimation Using Interpolation ')

% %%% Peak Estimation Using Sinc Method
figure(2)
mesh(F1-F2,actDistVec .* 2 ./ c, errors_sinc)
xlabel('Bandwidth (Hz)')
ylabel('Time Received (s)')
zlabel('Offset (m)')
title('TT - Peak Estimation Using Sinc Method')

% %%% Peak Estimation Using MF fitting Method
figure(3)
mesh(F1-F2,actDistVec .* 2 ./ c, errors_F)
zlim([-4,4])

```

```

xlabel('Bandwidth (Hz)')
ylabel('Time Received (s)')
zlabel('Offset (m)')
title('TT - Peak Estimation Using MF fitting Method')

% %%% Peak Estimation Using MF fitting Method 2
figure(4)
mesh(F1-F2, actDistVec .* 2 ./ c, errors_F2)
zlim([-4,4])
xlabel('Bandwidth (Hz)')
ylabel('Time Received (s)')
zlabel('Offset (m)')
title('TT - Peak Estimation Using MF fitting Method 2')

```

APPENDIX D

MATLAB SNR ESTIMATION USING EITHER EIGENVALUE DECOMPOSITION OR MATCHED FILTERING

```
% Initialization
A1 = 0.5;    A2 = 0.5;    c = 299792458;    % m/s

f1 = 10e3;
f2 = 1010e3;
fs = 10e6;

T = 0.0012;
D = 0.0038;
endOfSub = D + T;

noiseStep = 0.01;
% need snr instead of post int SNR
RdesdB = 0:70*noiseStep:70;    Sdes = 10.^(RdesdB./20)/2;

phi1 = rand(1);    phi2 = rand(1);
rate = 1/fs;

jj = 0;
ax = 0.1:0.3:20;
for ii = ax
    jj = jj + 1;

    %% TX RX signal generation
    delay = 0.5*D + rand(1)*D*0.05;

    % Signal Generation
    t_0 = 0 : rate : endOfSub;
```

```

t = t_0 - delay ;

% Signals with only real part
x1_1_0 = A1 * sin(2 * pi * f1 * t_0 - phi1);
x2_1_0 = A2 * sin(2 * pi * f2 * t_0 - phi2);
x1_1 = A1 * sin(2 * pi * f1 * t - phi1);
x2_1 = A2 * sin(2 * pi * f2 * t - phi2);

%% Signals with real and imag parts
% x1_0 = A1 * exp(- 1i * 2 * pi * f1 * t_0 - phi1);
% x2_0 = A2 * exp(- 1i * 2 * pi * f2 * t_0 - phi2);
% x1 = A1 * exp(- 1i * 2 * pi * f1 * t - phi1);
% x2 = A2 * exp(- 1i * 2 * pi * f2 * t - phi2);

x1 = x1_1; x2 = x2_1; x1_0 = x1_1_0; x2_0 = x2_1_0;

x1_0(floor(numel(x1_0)*T/endTime):end) = 0;    x2_0(floor(...
    numel(x1_0)*T/endTime):end) = 0;
[~, zeroDueToDelay] = min(abs(t));
x1(1:zeroDueToDelay) = 0;    x2(1:zeroDueToDelay) = 0;
x1(zeroDueToDelay+floor(numel(x1_0)*T/endTime):end) = 0;
x2(zeroDueToDelay+floor(numel(x1_0)*T/endTime):end) = 0;

noise = ii * (10.^(wgn(1, size(x1,2), 0)/10) - 10.^(...
    (wgn(1, size(x1,2), 0)/10)));
x = x1 + x2 + noise;

x_0 = x1_0 + x2_0;

% Actual SNR
snr_m(jj) = snr((x1+x2), noise);

%% Eigenvalue Decomposition

```

```

N = size(X,1);
L = size(X,2);

Rx = (1/N) * (X'*X);
v = (eig(Rx));
lambda = sort(v, 'descend ');

for M = 1:L
    Tm = prod(lambda(M+1:L).^(1/(L-M)));
    Pm = (1/(L-M))*sum(lambda(M+1:L));
    formula(M) = (-(L-M)*N*log(Tm/Pm)+0.5*M*(2*L - M)*log(N));
end

[~,locM] = min(formula);

% SNR with only one eigenvalue representing the signal power
K1 = locM;
Sn1 = mean(lambda(K1+1:end));
Ps1 = (1/L)*(sum(lambda(1:K1))-K1*Sn1^2);
snr_EVD1(jj) = 10*log(Ps1/(Sn1^2));

% SNR with multiple eigenvalues representing the signal power
K2 = find(lambda > (max(lambda)-min(lambda))*0.8 + min(lambda), 1, 'last ');
Sn2 = mean(lambda(K2+1:end));
Ps2 = (1/L)*(sum(lambda(1:K2))-K2*Sn2^2);
snr_EVD2(jj) = 10*log(Ps2/(Sn2^2));

%% SNR estimation using the Matched Filter Output
% Matched filter output
C = abs(xcorr(x, x_0));
[match_m, match_l] = max(C);

```

```

corrTop = max(abs(xcorr(x_0, x_0)));
rmsTop = rms(x1_0(1:floor(numel(x1_0)*T/endTime)-1)+...
    x2_0(1:floor(numel(x1_0)*T/endTime)-1))*(match_m/corrTop);

delay_samples = abs(ceil(numel(C)/2)-match_1);
pulseLength = floor(numel(x1_0)*T/endTime);

error_margin = pulseLength/20;
rms_of_noise = rms([x(1:max(2, delay_samples-error_margin)), ...
    x(min(delay_samples + 2 + pulseLength + error_margin, ...
    length(x)-1) : end)]);

snr_MF = 10*log10((rmsTop/rms_of_noise)^2);
end

```

APPENDIX E

MATLAB TIME OF ARRIVAL (TOA) ALGORITHM

```
clear

% From: Exact and Approximate Maximum Likelihood Localization Algorithms
% Yiu-Tong Chan, Herman Yau Chin Hang, and Pak-chung Ching

RXpos = [0.5588, 0, 0; 0, 0, 0; 0.5588, 0.254, 0; 0, 0.254, 0];
Dist = [0.3069; 0.3069; 0.3069; 0.3069];
DistVAR = [0.02;0.02;0.02;0.02].^2;
c = 3e8;

Q = diag(DistVAR); % covariance matrix
delta_i = Dist;

x_i = RXpos(:,1);
y_i = RXpos(:,2);
k_i = x_i.^2 + y_i.^2;

x_new = 0.2794;    y_new = 0.1270;
x_new = 0;    y_new = 0;

for i = 1:10
    x = x_new;
    y = y_new;

    r_i = sqrt( (x - x_i).^2 + (y - y_i).^2 );
    if i == 1
        g_i = x_i;
        h_i = y_i;
```

```

else
    g_i = (x - x_i) ./ (r_i .* (r_i + delta_i));
    h_i = (y - y_i) ./ (r_i .* (r_i + delta_i));
end
% s = x^2 + y^2;

LHS = 2 .* [sum(g_i .* x_i) , sum(g_i .* y_i); sum(h_i .* x_i) , ...
    sum(h_i .* y_i)];
% RHS = [sum(g_i .* (s + k_i - delta_i.^2)); sum(h_i .* ...
(s + k_i - delta_i.^2))];
% LSsolution = inv(LHS' * LHS) * LHS' * RHS; % \ means inv(LHS) * RHS
% x_new = LSsolution(1); y_new = LSsolution(2);

RHS_mod = [sum(g_i .* (0 + k_i - delta_i.^2)); sum(h_i .* (0 + k_i ...
    - delta_i.^2))];
LSsolution_mod = inv(LHS' * LHS) * LHS' * RHS_mod;

RHS_mod_s = [sum(g_i); sum(h_i)];
s_something = inv(LHS' * LHS) * LHS' * RHS_mod_s;

syms ss xx yy
eqns = [xx == LSsolution_mod(1) + ss * s_something(1), yy == ...
    LSsolution_mod(2) + ss * s_something(2), ss == xx^2 + yy^2];

S = solve(eqns, [ss xx yy]);
% x_new = double(S.xx);
% y_new = double(S.yy);
s_new = double(S.ss);

if isreal(s_new)

    if min(s_new > 0) && length(s_new) == 2

```

```

for j = 1:2
    s = s_new(j);
    RHS = [sum(g_i .* (s + k_i - delta_i.^2)); sum(h_i .* ...
        (s + k_i - delta_i.^2))];
    LSSolution = inv(LHS' * LHS) * LHS' * RHS;
    x_new = LSSolution(1);
    y_new = LSSolution(2);

    r_j = sqrt( (x_new - x_i).^2 + (y_new - y_i).^2 );

    JJ(j) = [Dist./c - r_j./c]' * inv(Q) * [Dist./c - r_j./c];
end
[u,v] = min(JJ);
s = s_new(v);
else
    s = max(s_new);
    if s < 0
        s = max(abs(s_new));
    end
end
else
    if length(s_new) == 2
        if isreal(s_new(1)) && s_new(1) > 0 && ~isreal(s_new(2))
            s = s_new(1)
        elseif isreal(s_new(2)) && s_new(2) > 0 && ~isreal(s_new(1))
            s = s_new(2)
        else
            s = max(real(s_new));
        end
    else
        s = max(real(s_new));
    end
end
end

```

```

        if s < 0
            s = max(abs(real(s_new)));
        end
    end
end

RHS = [sum(g_i .* (s + k_i - delta_i.^2)); sum(h_i .* (s + k_i - ...
    delta_i.^2))];
LSsolution = inv(LHS' * LHS) * LHS' * RHS;
x_new = LSsolution(1);
y_new = LSsolution(2);

r_j = sqrt( (x_new - x_i).^2 + (y_new - y_i).^2 );
J(i) = [Dist./c - r_j./c]' * inv(Q) * [Dist./c - r_j./c];

xxx(i) = x_new;
yyy(i) = y_new;
end
[xxx;yyy;J]

[uu,vv] = min(J);
x_selected = xxx(vv)
y_selected = yyy(vv)

```

APPENDIX F

MATLAB CRAMER-RAO LOWER BOUND FOR TIME OF ARRIVAL (TOA)

```
% close all; clc

clear

df = 4e6; % tone separation = ...MHz
f_0 = 0e6; % center frequency = ...MHz
Fs = 2 * df;
Fs = 20e6;

f_1 = f_0 - df/2;
f_2 = f_0 + df/2;
B = Fs/2; % noise bandwidth assumed equal to df
T = 1e-4; % pulse width = 1 ms
SNR_1 = 10^(42/10); % 10^(20/10) == 30 dB
SNR_2 = 10^(40.2/10); % 10^(20/10) == 30 dB
SNR_3 = 10^(41.8/10); % 10^(20/10) == 30 dB
SNR_4 = 10^(41/10); % 10^(20/10) == 30 dB

% Time delay estimation CRLB for two-tone waveform with the above settings
sigma_t1 = sqrt( 1/(((pi * df)^2+(2 * pi * f_0)^2)*(2*B*T*SNR_1)));
sigma_t2 = sqrt( 1/(((pi * df)^2+(2 * pi * f_0)^2)*(2*B*T*SNR_2)));
sigma_t3 = sqrt( 1/(((pi * df)^2+(2 * pi * f_0)^2)*(2*B*T*SNR_3)));
sigma_t4 = sqrt( 1/(((pi * df)^2+(2 * pi * f_0)^2)*(2*B*T*SNR_4)));

c = 3e8;
% *c/2 to make it in m and include two way factor:
sigma_i = [sigma_t1*c/2, sigma_t2*c/2, sigma_t3*c/2, sigma_t4*c/2];
sigma_i = [0.00608, 0.0061, 0.00458];
```

```

% Multiple case scenarios:
% 4 receivers and 1 transmitter locations
p_i = [+0.3, +0.3;...
       +0.3, -0.3; ...
       -0.3, +0.3; ...
       -0.3, -0.3]';

% 3 receivers and 1 transmitter locations
p_i = [ 0.1727,  0.1918;...
       -0.2038, -0.0572; ...
       -0.3683,  0.1918]';

xxLim =[-5 5]; yyLim = xxLim;
% xyRes = .1;
xyRes = (xxLim(2) - xxLim(1))/200;

ii = 0; xx = 0; yy = 0;
for y = xxLim(1):xyRes:xxLim(2)
    yy = yy + 1;
    xx = 0;
    for x = xxLim(1):xyRes:xxLim(2)
        xx = xx + 1;
        ii = ii + 1;
        p = [x; y];
        locations(1,1:2,ii) = p;

        F_11 = FIM_parts(p_i, p, sigma_i,1,1);
        F_22 = FIM_parts(p_i, p, sigma_i,2,2);
        F_12 = FIM_parts(p_i, p, sigma_i,1,2);
        F_21 = F_12;

        FIM(:, :, ii) = [F_11, F_12;...
                        F_21, F_22];
    end
end

```

```

invFIM(:, :, ii) = inv(FIM(:, :, ii));

Xvar(yy, xx) = (invFIM(1, 1, ii));
Yvar(yy, xx) = (invFIM(2, 2, ii));

Ovar(yy, xx) = (-y ./ (x.^2 + y.^2)).^2 .* Xvar(yy, xx) + ...
(x ./ (x.^2 + y.^2)).^2 .* Yvar(yy, xx) + ...
2.*(-y ./ (x.^2 + y.^2)) .* (x ./ (x.^2 + y.^2)) .* (invFIM(1, 2, ii));

Rvar(yy, xx) = (x ./ sqrt(x.^2 + y.^2)).^2 .* Xvar(yy, xx) + ...
(y ./ sqrt(x.^2 + y.^2)).^2 .* Yvar(yy, xx) + 2.*(x ./ ...
sqrt(x.^2 + y.^2)) .* (y ./ sqrt(x.^2 + y.^2)) .* (invFIM(1, 2, ii));

% O = atan(y/x);
O = atan2d(y, x);
R = sqrt(x^2 + y^2);

Xvar2(yy, xx) = Rvar(yy, xx) * cos(O)^2 + Ovar(yy, xx) * R^2 * sin(O)^2;
Yvar2(yy, xx) = Rvar(yy, xx) * sin(O)^2 + Ovar(yy, xx) * R^2 * cos(O)^2;
XYcov2(yy, xx) = (Rvar(yy, xx) - Ovar(yy, xx) * R^2) * sin(O) * cos(O);

end

end

% in case we have singular points:
% Xvar(isinf(Xvar)|isnan(Xvar)) = 10;
% Yvar(isinf(Yvar)|isnan(Yvar)) = 10;
% Xvar2(isinf(Xvar2)|isnan(Xvar2)) = 10;
% Yvar2(isinf(Yvar2)|isnan(Yvar2)) = 10;
% Ovar(isinf(Ovar)|isnan(Ovar)) = 10;

```

```

%% Plotting theta and R
[X,Y] = meshgrid(xxLim(1):xyRes:xxLim(2));
h = figure;
A = axes;
mesh(X,Y, sqrt((Ovar)).*180./pi)
set(A, 'ZScale', 'log')
xlabel('x (m)')
ylabel('y (m)')
title('Standard deviation for \theta (degree)');

h = figure;
A = axes;
mesh(X,Y, sqrt(abs(Rvar)))
set(A, 'ZScale', 'log')
xlabel('x (m)')
ylabel('y (m)')
title('Standard deviation for distance (m)');

%% Figures
figure
clims = [0.00 0.1];
imagesc(xxLim,yyLim, sqrt(abs(Xvar)), clims)
% imagesc(xxLim,yyLim, sqrt(abs(Xvar)))
xlabel('x (m)')
ylabel('y (m)')
h = colorbar;
set(get(h, 'label'), 'string', 'Standard deviation of x (m)');

figure
% clims = [0.00001 0.01];
imagesc(xxLim,yyLim, sqrt(abs(Yvar)), clims)
xlabel('x (m)')

```

```

ylabel('y (m)')
h = colorbar;
set(get(h,'label'),'string','Standard deviation of y (m)');

```

```

figure
clims = [0 0.15];
imagesc(xxLim,yyLim,sqrt(abs(Xvar2)+abs(Yvar2)),clims)
xlabel('x (m)')
ylabel('y (m)')
h = colorbar;
set(get(h,'label'),'string','\surd(CRLB_x + CRLB_y) (m)');

```

```

% [X,Y] = meshgrid(xxLim(1):xyRes:xxLim(2));
h = figure;
A = axes;
mesh(X,Y,sqrt(abs(Xvar)))
set(A,'ZScale','log')
xlabel('x (m)')
ylabel('y (m)')

```

```

h = figure;
A = axes;
mesh(X,Y,sqrt(abs(Yvar)))
set(A,'ZScale','log')
xlabel('x (m)')
ylabel('y (m)')

```

```

h = figure;
A = axes;
mesh(X,Y,sqrt(abs(Xvar)+abs(Yvar)))
set(A,'ZScale','log')
xlabel('x (m)')

```

```

ylabel('y (m)')
title('\surd(std_x^2 + std_y^2) (m)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function F = FIM_parts(p_i, p, sigma_i, xy1, xy2)

N = size(p_i,2); %number of nodes
for i = 1:N
    d(i) = norm(p-p_i(:,i));
    a_11(i) = (1/(sigma_i(i)^2)) * (p(xy1)-p_i(xy1,i)) * (p(xy2)-...
        p_i(xy2,i))/(d(i)^2);
end

F = sum(a_11);

end

```

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Serge R Mghabghab and Jeffrey A Nanzer. Localization in distributed wireless systems based on high-accuracy microwave ranging. In *2021 IEEE-APS Topical Conference on Antennas and Propagation in Wireless Communications (APWC)*, pages 122–125. IEEE, 2021.
- [2] Patrick Bidigare, Miguel Oyarzyn, David Raeman, Dan Chang, Dave Cousins, Rich O’Donnell, Charlie Obranovich, and D Richard Brown. Implementation and demonstration of receiver-coordinated distributed transmit beamforming across an ad-hoc radio network. In *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 222–226. IEEE, 2012.
- [3] Sean M. Ellison Jeffrey A. Nanzer, Serge R. Mghabghab and Anton Schlegel. Distributed phased arrays: Challenges and recent advances. *IEEE Trans. Microw. Theory Techn.*, 2021.
- [4] Serge Mghabghab and Jeffrey A Nanzer. Ranging requirements for open-loop coherent distributed arrays with wireless frequency synchronization. In *2020 IEEE USNC-CNC-URSI North American Radio Science Meeting (Joint with AP-S Symposium)*, pages 69–70. IEEE, 2020.
- [5] Serge R Mghabghab and Jeffrey A Nanzer. Impact of localization error on open-loop distributed beamforming arrays. In *2021 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)*, pages 1–3. IEEE, 2021.
- [6] Serge R Mghabghab and Jeffrey A Nanzer. Impact of vco and pll phase noise on distributed beamforming arrays with periodic synchronization. *IEEE Access*, 9:56578–56588, 2021.
- [7] Sean M Ellison, Serge R Mghabghab, and Jeffrey A Nanzer. Scalable high-accuracy ranging and wireless frequency synchronization for open-loop distributed phased arrays. In *2020 IEEE 63rd International Midwest Symposium on Circuits and Syst.*, pages 41–44, August 2020.
- [8] Serge Mghabghab and Jeffrey A Nanzer. A self-mixing receiver for wireless frequency synchronization in coherent distributed arrays. In *2020 IEEE/MTT-S International Microwave Symposium (IMS)*, pages 1137–1140. IEEE, 2020.
- [9] Serge Mghabghab, Hassna Ouassal, and Jeffrey A Nanzer. Wireless frequency synchronization for coherent distributed antenna arrays. In *2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting*, pages 1575–1576. IEEE, 2019.
- [10] Serge R Mghabghab and Jeffrey A Nanzer. Adaptive internode ranging for coherent distributed antenna arrays. *IEEE Transactions on Aerospace and Electronic Systems*, 56(6):4689–4697, 2020.

- [11] Serge Mghabghab and Jeffrey A Nanzer. Microwave ranging via least squares estimation of spectrally sparse signals in software defined radio. *IEEE Microwave and Wireless Components Letters*, 2021.
- [12] Serge R Mghabghab and Jeffrey A Nanzer. High accuracy adaptive microwave ranging using snr-based perception for coherent distributed antenna arrays. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(12):5540–5549, 2020.
- [13] Serge Mghabghab, Anton Schlegel, Robert Gress, and Jeffrey A Nanzer. Long-range wireless frequency synchronization for distributed phased arrays. In *2020 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*, pages 1253–1254. IEEE, 2020.
- [14] Serge R Mghabghab and Jeffrey A Nanzer. Open-loop distributed beamforming using wireless frequency synchronization. *IEEE Trans. Microw. Theory Techn.*, 69(1):896–905, September 2020.
- [15] Serge Mghabghab, Sean M. Ellison, and Jeffrey A Nanzer. Distributed beamforming using wireless phase and frequency synchronization. *IEEE Microwave and Wireless Components Letters*, 2021.
- [16] Don Parker and David C Zimmermann. Phased arrays-part 1: theory and architectures. *IEEE transactions on microwave theory and techniques*, 50(3):678–687, 2002.
- [17] Don Parker and David C Zimmermann. Phased arrays-part ii: implementations, applications, and future trends. *IEEE Transactions on microwave theory and techniques*, 50(3):688–698, 2002.
- [18] Caleb Fulton, Mark Yeary, Daniel Thompson, John Lake, and Adam Mitchell. Digital phased arrays: Challenges and opportunities. *Proceedings of the IEEE*, 104(3):487–503, 2016.
- [19] Salvador H Talisa, Kenneth W O’Haver, Thomas M Comberiate, Matthew D Sharp, and Oscar F Somerlock. Benefits of digital phased array radars. *Proceedings of the IEEE*, 104(3):530–543, 2016.
- [20] P Benthem and GW Kant. Embrace: Results from an aperture array for radio astronomy. In *2012 6th European Conference on Antennas and Propagation (EUCAP)*, pages 629–633. IEEE, 2012.
- [21] AI Zaghloui, O Kilic, LQ Sun, and M Lieberman. Distributed aperture design for low profile earth station antennas. In *IEEE Antennas and Propagation Society International Symposium. 1999 Digest. Held in conjunction with: USNC/URSI National Radio Science Meeting (Cat. No. 99CH37010)*, volume 3, pages 2088–2091. IEEE, 1999.
- [22] Marco Rossi, Alexander M Haimovich, and Yonina C Eldar. Spatial compressive sensing for mimo radar. *IEEE Transactions on Signal Processing*, 62(2):419–430, 2013.

- [23] Sairam Goguri, Dennis Ogbe, Soura Dasgupta, Raghuraman Mudumbai, D Richard Brown, David J Love, and Upamanyu Madhow. Optimal precoder design for distributed transmit beamforming over frequency-selective channels. *IEEE Transactions on Wireless Communications*, 17(11):7759–7773, 2018.
- [24] MK Arti and Manav R Bhatnagar. Beamforming and combining in hybrid satellite-terrestrial cooperative systems. *IEEE Communications Letters*, 18(3):483–486, 2014.
- [25] Junming Diao, Maziar Hedayati, and Yunxuan Ethan Wang. Experimental demonstration of distributed beamforming on two flying mini-drones. In *2019 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, pages 1–2. IEEE, 2019.
- [26] Neil JA Egarguin, David R Jackson, Daniel Onofrei, Julien Leclerc, and Aaron Becker. Adaptive beamforming using scattering from a drone swarm. In *2020 IEEE Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS)*, pages 1–6. IEEE, 2020.
- [27] Shangang Fan, Jiangbo Liu, Mengqian Tian, Hao Huang, Fei Dai, Lie Yang, and Guan Gui. Robust adaptive beamforming signal techniques for drone surveillance. In *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 582–586. IEEE, 2018.
- [28] Yoshimi Fujii, Tetsuya Iye, Kensuke Tsuda, and Akihiro Tanibayashi. 28ghz cooperative digital beamforming for 5g advanced system on an sdr platform. In *2021 IEEE Radio and Wireless Symposium (RWS)*, pages 80–82. IEEE, 2021.
- [29] Daisuke Kurita, Kiichi Tateishi, Atsushi Harada, Yoshihisa Kishiyama, Shoji Itoh, Hideshi Murai, Arne Simonsson, and Peter Okvist. Indoor and outdoor experiments on 5g radio access using distributed mimo and beamforming in 15 ghz frequency band. In *2016 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2016.
- [30] Pratik Chatterjee and Jeffrey A Nanzer. Effects of time alignment errors in coherent distributed radar. In *2018 IEEE Radar Conference (RadarConf18)*, pages 0727–0731. IEEE, 2018.
- [31] Pilei Yin, Xiaopeng Yang, Tao Zeng, and Xiaona Hu. Robust time synchronization method based on step frequency signal for wideband distributed coherent aperture radar. In *2013 IEEE International Symposium on Phased Array Systems and Technology*, pages 383–388. IEEE, 2013.
- [32] C Shen and H Yu. Time-delay alignment technique for a randomly distributed sensor array. *IET communications*, 5(8):1068–1072, 2011.
- [33] Alexander Tashlinskiy and Mikhail Tsarev. An algorithm for time shift estimation of radio pulses received by spatially distributed sensors. In *2015 International Siberian Conference on Control and Communications (SIBCON)*, pages 1–4. IEEE, 2015.

- [34] Thomas Musch, Michael Gerding, and Burkhard Schiek. A phase-locked-loop concept for the generation of two rf-signals with a small frequency offset. *IEEE transactions on instrumentation and measurement*, 54(2):709–712, 2005.
- [35] Wei-Hao Chiu, Yu-Hsiang Huang, and Tsung-Hsien Lin. A dynamic phase error compensation technique for fast-locking phase-locked loops. *IEEE Journal of Solid-State Circuits*, 45(6):1137–1149, 2010.
- [36] Raghuraman Mudumbai, Ben Wild, Upamanyu Madhow, and Kannan Ramchandran. Distributed beamforming using 1 bit feedback: from concept to realization. In *Proceedings of the 44th Allerton conference on commun., control and comput.*, volume 8, pages 1020–1027. Citeseer, 2006.
- [37] Ben Peiffer, Raghu Mudumbai, Sairam Goguri, Anton Kruger, and Soura Dasgupta. Experimental demonstration of retrodirective beamforming from a fully wireless distributed array. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 442–447. IEEE, 2016.
- [38] Raghuraman Mudumbai, Gwen Barriac, and Upamanyu Madhow. On the feasibility of distributed beamforming in wireless networks. *IEEE Transactions on Wireless communications*, 6(5):1754–1763, 2007.
- [39] Robert D Preuss and D Richard Brown III. Two-way synchronization for coordinated multicell retrodirective downlink beamforming. *IEEE Transactions on Signal Processing*, 59(11):5415–5427, 2011.
- [40] Omid Abari, Hariharan Rahul, Dina Katabi, and Mondira Pant. Airshare: Distributed coherent transmission made seamless. In *IEEE Conference on Comput. Commun.*, pages 1742–1750, April 2015.
- [41] Richard K Pooler, Joshua S Sunderlin, R Henry Tillman, and Robert L Schmid. A precise rf time transfer method for coherent distributed system applications. In *IEEE USNC-URSI Radio Sci. Meeting (Joint with AP-S Symposium)*, pages 5–6, July 2018.
- [42] F. Sivrikaya and B. Yener. Time synchronization in sensor networks: a survey. *IEEE Netw.*, 18(4):45–50, July 2004.
- [43] Saurabh Ganeriwal, Ram Kumar, and Mani B Srivastava. Timing-sync protocol for sensor networks. In *1st international conference on Embedded netw. sensor syst.*, pages 138–149, November 2003.
- [44] Mihail L Sichitiu and Chanchai Veerarittiphan. Simple, accurate time synchronization for wireless sensor networks. In *IEEE Wireless Commun. and Netw.*, volume 2, pages 1266–1273, May 2003.
- [45] M. Li, S. Gvozdenovic, A. Ryan, R. David, D. R. Brown, and A. G. Klein. A real-time implementation of precise timestamp-free network synchronization. In *49th Asilomar Conference on Signals, Syst. and Comput.*, pages 1214–1218, November 2015.

- [46] D Richard Brown and Andrew G Klein. Precise timestamp-free network synchronization. In *47th Annual Conference on Inf. Sci. and Syst.*, pages 1–6, March 2013.
- [47] Mitchell WS Overdick, Joseph E Canfield, Andrew G Klein, and D Richard Brown. A software-defined radio implementation of timestamp-free network synchronization. In *IEEE International Conference on Acoustics, Speech and Signal Proc.*, pages 1193–1197, March 2017.
- [48] J. A. Nanzer, R. L. Schmid, T. M. Comberiate, and J. E. Hodkin. Open-loop coherent distributed arrays. *IEEE Trans. Microw. Theory Techn.*, 65(5):1662–1672, January 2017.
- [49] Michele Morelli, C-C Jay Kuo, and Man-On Pun. Synchronization techniques for orthogonal frequency division multiple access (ofdma): A tutorial review. *Proc. IEEE*, 95(7):1394–1427, August 2007.
- [50] P. Bidigare, M. Oyarzyn, D. Raeman, D. Chang, D. Cousins, R. O’Donnell, C. Obranovich, and D. R. Brown. Implementation and demonstration of receiver-coordinated distributed transmit beamforming across an ad-hoc radio network. In *Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers*, pages 222–226, November 2012.
- [51] I. C. Sezgin, M. Dahlgren, T. Eriksson, M. Coldrey, C. Larsson, J. Gustavsson, and C. Fager. A low-complexity distributed-mimo testbed based on high-speed sigma–delta-over-fiber. *IEEE Trans. Microw. Theory Techn.*, 67(7):2861–2872, April 2019.
- [52] Munkyo Seo, M. Rodwell, and U. Madhow. A feedback-based distributed phased array technique and its application to 60-ghz wireless sensor network. In *IEEE MTT-S International Microwave Symposium Digest*, pages 683–686, June 2008.
- [53] Mabel Pontón and Almudena Suárez. Stability analysis of wireless coupled-oscillator circuits. In *IEEE MTT-S International Microwave Symposium*, pages 83–86, June 2017.
- [54] Xuebei Yang, Xuyang Lu, and Aydin Babakhani. Picosecond wireless synchronization using an optically locked voltage controlled oscillator (ol-vco). In *IEEE MTT-S International Microwave Symposium*, pages 1–4, June 2014.
- [55] Kun-Yuan Tu and Chia-Shu Liao. Application of anfis for frequency syntonization using gps carrier-phase measurements. In *IEEE International Freq. Control Symposium Joint with the 21st European Freq. and Time Forum*, pages 933–936, May 2007.
- [56] R. Mudumbai, G. Barriac, and U. Madhow. On the feasibility of distributed beamforming in wireless networks. *IEEE Transactions on Wireless Commun.*, 6(5):1754–1763, May 2007.
- [57] D. R. Brown, G. B. Prince, and J. A. McNeill. A method for carrier frequency and phase synchronization of two autonomous cooperative transmitters. In *IEEE 6th Workshop on Signal Process. Advances in Wireless Commun.*, pages 260–264, June 2005.
- [58] Hassna Ouassal, Torre Rocco, Ming Yan, and Jeffrey A Nanzer. Decentralized frequency synchronization in distributed antenna arrays with quantized frequency states and directed communications. *IEEE Trans. Antennas Propag.*, 68(7):5280–5288, March 2020.

- [59] D. R. Brown III and H. V. Poor. Time-slotted round-trip carrier synchronization for distributed beamforming. *IEEE Trans. Signal Process.*, 56(11):5630–5643, June 2008.
- [60] R. D. Preuss and D. R. Brown, III. Two-way synchronization for coordinated multicell retrodirective downlink beamforming. *IEEE Trans. Signal Process.*, 59(11):5415–5427, July 2011.
- [61] C Pon. Retrodirective array using the heterodyne technique. *IEEE Transactions on Antennas and Propagation*, 12(2):176–180, 1964.
- [62] Ioannis Krikidis. Retrodirective large antenna energy beamforming in backscatter multi-user networks. *IEEE Wireless Communications Letters*, 7(4):678–681, 2018.
- [63] Robert D Preuss and D Richard Brown. Retrodirective distributed transmit beamforming with two-way source synchronization. In *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2010.
- [64] Sean M Ellison and Jeffrey A Nanzer. Open-loop distributed beamforming using scalable high accuracy localization. *arXiv preprint arXiv:2008.07748*, 2020.
- [65] Hassna Ouassal, Ming Yan, and Jeffrey A Nanzer. Decentralized frequency alignment for collaborative beamforming in distributed phased arrays. *IEEE Transactions on Wireless Communications*, 2021.
- [66] Stuart Bennett. Development of the pid controller. *IEEE Control Systems Magazine*, 13(6):58–62, 1993.
- [67] John G Ziegler, Nathaniel B Nichols, et al. Optimum settings for automatic controllers. *trans. ASME*, 64(11), 1942.
- [68] Davide Dardari, Andrea Conti, Ulric Ferner, Andrea Giorgetti, and Moe Z Win. Ranging with ultrawide bandwidth signals in multipath environments. *Proceedings of the IEEE*, 97(2):404–426, 2009.
- [69] Reinhard Feger, Christoph Wagner, Stefan Schuster, Stefan Scheiblhofer, Herbert Jager, and Andreas Stelzer. A 77-ghz fmcw mimo radar based on an sige single-chip transceiver. *IEEE Transactions on Microwave theory and Techniques*, 57(5):1020–1035, 2009.
- [70] Cemin Zhang, Michael J Kuhn, Brandon C Merkl, Aly E Fathy, and Mohamed R Mahfouz. Real-time noncoherent uwb positioning radar with millimeter range accuracy: Theory and experiment. *IEEE Transactions on Microwave Theory and Techniques*, 58(1):9–20, 2009.
- [71] Andreas Stelzer, Christian G Diskus, Kurt Lubke, and Hartwig W Thim. A microwave position sensor with submillimeter accuracy. *IEEE Transactions on Microwave Theory and Techniques*, 47(12):2621–2624, 1999.
- [72] C Meier, A Terzis, and S Lindenmeier. A robust 3d high precision radio location system. In *2007 IEEE/MTT-S International Microwave Symposium*, pages 397–400. IEEE, 2007.

- [73] Adrian N Bishop, Barış Fidan, Brian DO Anderson, Kutluyıl Doğançay, and Pubudu N Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, 46(3):479–492, 2010.
- [74] Stavros Vakalis and Jeffrey A Nanzer. Millimeter-wave angle estimation of multiple targets using space-time modulation and interferometric antenna arrays. *arXiv preprint arXiv:2008.00356*, 2020.
- [75] Jeffrey A Nanzer and Matthew D Sharp. On the estimation of angle rate in radar. *IEEE Trans. Antennas Propag.*, 65(3):1339–1348, December 2016.
- [76] Sheng Xu and Kutluyıl Doğançay. Optimal sensor deployment for 3d aoa target localization. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2544–2548. IEEE, 2015.
- [77] Benoit Denis, J-B Pierrot, and Chadi Abou-Rjeily. Joint distributed synchronization and positioning in uwb ad hoc networks using toa. *IEEE transactions on microwave theory and techniques*, 54(4):1896–1911, 2006.
- [78] Stefan Galler, Waldemar Gerok, Jens Schroeder, Kyandoghene Kyamakya, and Thomas Kaiser. Combined aoa/toa uwb localization. In *2007 International Symposium on Communications and Information Technologies*, pages 1049–1053. IEEE, 2007.
- [79] Mohamed Laaraiedh, Stéphane Avrillon, and Bernard Uguen. A maximum likelihood toa based estimator for localization in heterogeneous networks. *Int’l J. of Communications, Network and System Sciences*, 3(1):38–42, 2010.
- [80] Xinya Li, Zhiqun Daniel Deng, Lynn T Rauchenstein, and Thomas J Carlson. Contributed review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements. *Review of Scientific Instruments*, 87(4):041502, 2016.
- [81] Naresh Vankayalapati, Steven Kay, and Quan Ding. Tdoa based direct positioning maximum likelihood estimator and the cramer-rao bound. *IEEE transactions on aerospace and electronic systems*, 50(3):1616–1635, 2014.
- [82] Regina Kaune, Julian Hörst, and Wolfgang Koch. Accuracy analysis for tdoa localization in sensor networks. In *14th international conference on information fusion*, pages 1–8. IEEE, 2011.
- [83] Regina Kaune. Accuracy studies for tdoa and toa localization. In *2012 15th International Conference on Information Fusion*, pages 408–415. IEEE, 2012.
- [84] Richard K Martin, Chunpeng Yan, H Howard Fan, and Christopher Rondeau. Algorithms and bounds for distributed tdoa-based positioning using ofdm signals. *IEEE transactions on signal processing*, 59(3):1255–1268, 2010.
- [85] Arie Yeredor and Eyal Angel. Joint tdoa and fdoa estimation: A conditional bound and its use for optimally weighted localization. *IEEE Transactions on Signal Processing*, 59(4):1612–1623, 2010.

- [86] Dong-Gyu Kim, Geun-Ho Park, Hyoung-Nam Kim, Jin-Oh Park, Young-Mi Park, and Wook-Hyeon Shin. Computationally efficient tdoa/fdoa estimation for unknown communication signals in electronic warfare systems. *IEEE Transactions on Aerospace and Electronic Systems*, 54(1):77–89, 2017.
- [87] KC Ho, Xiaoning Lu, and La-or Kovavisaruch. Source localization using tdoa and fdoa measurements in the presence of receiver location errors: Analysis and solution. *IEEE Transactions on Signal Processing*, 55(2):684–696, 2007.
- [88] JM Fornies-Marquina, J Letosa, M Garcia-Gracia, and JM Artacho. Error propagation for the transformation of time domain into frequency domain. *IEEE Transactions on Magnetics*, 33(2):1456–1459, 1997.
- [89] John E. Freund and Ronald E. Walpole. *Mathematical Statistics 4th Ed.* Prentice Hall, 1987.
- [90] Adem Aktas and Mohammed Ismail. *CMOS PLLs and VCOs for 4G wireless.* Springer Science & Business Media, 2004.
- [91] David A Howe. *Frequency domain stability measurements: a tutorial introduction*, volume 13. US Department of Commerce, National Bureau of Standards, Institute for Basic . . . , 1976.
- [92] B Razavi and RF Microelectronics. Prentice-hall: Englewood cliffs. NJ, USA, 1998.
- [93] D. B. Leeson. A simple model of feedback oscillator noise spectrum. *Proceedings of the IEEE*, 54(2):329–330, 1966.
- [94] Konstantin A Kouznetsov and Robert G Meyer. Phase noise in lc oscillators. *IEEE Journal of Solid-State Circuits*, 35(8):1244–1248, 2000.
- [95] Thomas H Lee and Ali Hajimiri. Oscillator phase noise: A tutorial. *IEEE journal of solid-state circuits*, 35(3):326–336, 2000.
- [96] Guan-Chyun Hsieh and James C Hung. Phase-locked loop techniques. a survey. *IEEE Transactions on industrial electronics*, 43(6):609–615, 1996.
- [97] Cicero S Vaucher. *Architectures for RF frequency synthesizers*, volume 693. Springer Science & Business Media, 2006.
- [98] Shobha N Pawar and Pradeep B Mane. Wide band pll frequency synthesizer: A survey. In *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pages 1–6. IEEE, 2017.
- [99] Reza Binaei and Mohammad Gholami. Using d flip-flop with reset terminal to design pfd in qca nanotechnology. *International Journal of Electronics*, 107(12):1940–1962, 2020.
- [100] Dean Banerjee. *PLL performance, simulation and design.* Dog Ear Publishing, 2006.
- [101] Peter J Ashenden, Gregory D Peterson, and Darrell A Teegarden. *The system designer's guide to VHDL-AMS: analog, mixed-signal, and mixed-technology modeling.* Elsevier, 2002.

- [102] David W Allan. Statistics of atomic frequency standards. *Proceedings of the IEEE*, 54(2):221–230, 1966.
- [103] James A Barnes, Andrew R Chi, Leonard S Cutler, Daniel J Healey, David B Leeson, Thomas E McGunigal, James A Mullen, Warren L Smith, Richard L Sydnor, Robert FC Vessot, et al. Characterization of frequency stability. *IEEE transactions on instrumentation and measurement*, (2):105–120, 1971.
- [104] Stefano Bregni. *Synchronization of digital telecommunications networks*, volume 27. Wiley Online Library, 2002.
- [105] William J Riley. Handbook of frequency stability analysis. 2008.
- [106] David Clark. *Wireless Communications & Networking. The Morgan Kaufmann Series in Networking*. Elsevier Science & Technology, 2007.
- [107] Connor-Winfield-Corporation. TB / TVB Model Series, 2018.
- [108] Wenzel-Associates-Inc. Wenzel VHF Citrine 501-31631 crystal oscillator, 2018.
- [109] Z. Liang, G. Zhang, X. Dong, and Y. Huo. Design and analysis of passband transmitted reference pulse cluster uwb systems in the presence of phase noise. *IEEE Access*, 6:14954–14965, 2018.
- [110] Yupeng Fu, Lianming Li, Dongming Wang, and Xuan Wang. A- 193.6 dbc/hz fom t 28.6-to-36.2 ghz dual-core cmos vco for 5g applications. *IEEE Access*, 8:62191–62196, 2020.
- [111] Stephen A Maas. Microwave mixers. *Norwood*, 1986.
- [112] B Roy Frieden. *Science from Fisher information: a unification*. Cambridge University Press, 2004.
- [113] Harry L Van Trees. *Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory*. John Wiley & Sons, 2004.
- [114] Sean M Ellison, Serge Mghabghab, John J Doroshewitz, and Jeffrey A Nanzer. Combined wireless ranging and frequency transfer for internode coordination in open-loop coherent distributed antenna arrays. *IEEE Trans. Microw. Theory Techn.*, 68(1):277–287, October 2019.
- [115] J. E. Hodkin, K. S. Zilevu, M. D. Sharp, T. M. Comberiate, S. M. Hendrickson, M. J. Fitch, and J. A. Nanzer. Microwave and millimeter-wave ranging for coherent distributed rf systems. In *2015 IEEE Aerosp. Conference*, pages 1–7, March 2015.
- [116] Samuel Prager, Mark S Haynes, and Mahta Moghaddam. Wireless subnanosecond rf synchronization for distributed ultrawideband software-defined radar networks. *IEEE Transactions on Microwave Theory and Techniques*, 68(11):4787–4804, 2020.
- [117] Richard A Caruana, Roger B Searle, Thomas Heller, and Saul I Shupack. Fast algorithm for the resolution of spectra. *Analytical chemistry*, 58(6):1162–1167, 1986.

- [118] Hongwei Guo. A simple algorithm for fitting a gaussian function [dsp tips and tricks]. *IEEE Signal Processing Magazine*, 28(5):134–137, 2011.
- [119] Ian Sharp, Kegen Yu, and Y Jay Guo. Peak and leading edge detection for time-of-arrival estimation in band-limited positioning systems. *IET communications*, 3(10):1616–1627, 2009.
- [120] Sean M Ellison and Jeffrey A Nanzer. High-accuracy multinode ranging for coherent distributed antenna arrays. *IEEE Trans. Aerosp. Electron. Syst.*, 56(5):4056–4066, April 2020.
- [121] Botao Ma, Haowen Chen, Bin Sun, and Huaitie Xiao. A joint scheme of antenna selection and power allocation for localization in mimo radar sensor networks. *IEEE communications letters*, 18(12):2225–2228, 2014.
- [122] Zhu Han and KJ Ray Liu. Power minimization under constant throughput constraint in wireless networks with beamforming. In *Proceedings IEEE 56th Vehicular Technology Conference*, volume 1, pages 611–615. IEEE, 2002.
- [123] Pratik Chatterjee and Jeffrey A Nanzer. A study of coherent gain degradation due to node vibrations in open loop coherent distributed arrays. In *2017 USNC-URSI Radio Science Meeting (Joint with AP-S Symposium)*, pages 115–116. IEEE, 2017.
- [124] Karaputugala Madushan Thilina, Mohammad Moghadari, and Ekram Hossain. Generalized spectral footprint minimization for ofdma-based cognitive radio networks. In *2014 IEEE International Conference on Communications (ICC)*, pages 1657–1662. IEEE, 2014.
- [125] Karaputugala G Madushan Thilina, Ekram Hossain, and Mohammad Moghadari. Cellular ofdma cognitive radio networks: Generalized spectral footprint minimization. *IEEE Transactions on Vehicular Technology*, 64(7):3190–3204, 2014.
- [126] Deborah Cohen, Kumar Vijay Mishra, and Yonina C Eldar. Spectrum sharing radar: Coexistence via xampling. *IEEE Transactions on Aerospace and Electronic Systems*, 54(3):1279–1296, 2017.
- [127] Hai Deng and Braham Himed. Interference mitigation processing for spectrum-sharing between radar and wireless communications systems. *IEEE Transactions on Aerospace and Electronic Systems*, 49(3):1911–1919, 2013.
- [128] Anthony F Martone, Kenneth I Ranney, Kelly Sherbondy, Kyle A Gallagher, and Shannon D Blunt. Spectrum allocation for noncooperative radar coexistence. *IEEE Transactions on Aerospace and Electronic Systems*, 54(1):90–105, 2017.
- [129] Tianyao Huang, Yimin Liu, Huadong Meng, and Xiqin Wang. Cognitive random stepped frequency radar with sparse recovery. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2):858–870, 2014.
- [130] Simon Haykin. Cognitive radar: a way of the future. *IEEE signal processing magazine*, 23(1):30–40, 2006.

- [131] Joseph R Guerci. Cognitive radar: A knowledge-aided fully adaptive approach. In *2010 IEEE Radar Conference*, pages 1365–1370. IEEE, 2010.
- [132] D DeLong and E Hofstetter. On the design of optimum radar waveforms for clutter rejection. *IEEE Transactions on Information Theory*, 13(3):454–463, 1967.
- [133] D DeLong and E Hofstetter. The design of clutter-resistant radar waveforms with limited dynamic range. *IEEE Transactions on Information Theory*, 15(3):376–385, 1969.
- [134] Antonio De Maio and Alfonso Farina. Cognitive radar signal processing. In *Proc. Workshop Math. Issues Inf. Sci.*, pages 6–18, 2014.
- [135] Moez Ben Kilani, Yogesh Nijsure, Ghyslain Gagnon, Georges Kaddoum, and François Gagnon. Cognitive waveform and receiver selection mechanism for multistatic radar. *IET Radar, Sonar & Navigation*, 10(2):417–425, 2016.
- [136] Gannan Yuan, Xingli Gan, and Wei Zhou. Design a high-precision controller of attitude stabilization in wave-measured radar. In *2007 International Conference on Mechatronics and Automation*, pages 3305–3309. IEEE, 2007.
- [137] Mustafa Yagimli, Ugur Simsir, and Hakan Tozan. Trajectory tracking of a satellite communicated missile with fuzzy and pid control. In *Proceedings of 5th International Conference on Recent Advances in Space Technologies-RAST2011*, pages 318–323. IEEE, 2011.
- [138] Murad Yaghi and Mehmet Önder Efe. Fractional order pid control of a radar guided missile under disturbances. In *2018 9th International Conference on Information and Communication Systems (ICICS)*, pages 238–242. IEEE, 2018.
- [139] Kiam Heong Ang, Gregory Chong, and Yun Li. Pid control system analysis, design, and technology. *IEEE transactions on control systems technology*, 13(4):559–576, 2005.
- [140] Arne Svensson. An introduction to adaptive qam modulation schemes for known and predicted channels. *Proceedings of the IEEE*, 95(12):2322–2336, 2007.
- [141] Biren Shah and Sami Hinedi. The split symbol moments snr estimator in narrow-band channels. *IEEE transactions on aerospace and electronic systems*, 26(5):737–747, 1990.
- [142] Robert B Kerr. On signal and noise level estimation in a coherent pcm channel. *IEEE Transactions on Aerospace and Electronic Systems*, (4):450–454, 1966.
- [143] CE Gilchrist. Signal-to-noise monitoring. *JPL Space Programs Summary*, 4(37-27):169–184, 1966.
- [144] Rolf Matzner. An snr estimation algorithm for complex baseband signals using higher-order statistics. *Facta Universitatis (Nis)*, 1993, 6:41–52, 1993.
- [145] AL Brandao, Luis B Lopes, and Desmond C McLemon. In-service monitoring of multipath delay and cochannel interference for indoor mobile communication systems. In *Proceedings of ICC/SUPERCOMM'94-1994 International Conference on Communications*, pages 1458–1462. IEEE, 1994.

- [146] David R Pauluzzi and Norman C Beaulieu. A comparison of snr estimation techniques for the awgn channel. *IEEE Transactions on communications*, 48(10):1681–1691, 2000.
- [147] Mohamed Hamid, Niclas Björsell, and Slimane Ben Slimane. Sample covariance matrix eigenvalues based blind snr estimation. In *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pages 718–722. IEEE, 2014.
- [148] N Shafiza Mohd Tamim and Farid Ghani. Techniques for optimization in time delay estimation from cross correlation function. *Int J Eng Technol*, 10(2):69–75, 2010.
- [149] A. R. Jiménez Ruiz and F. Seco Granja. Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis. *IEEE Transactions on Instrumentation and Measurement*, 66(8):2106–2117, 2017.
- [150] Yiu-Tong Chan, H Yau Chin Hang, and Pak-chung Ching. Exact and approximate maximum likelihood localization algorithms. *IEEE Transactions on Vehicular Technology*, 55(1):10–16, 2006.