



ARL-TR-9184 • APR 2021



A Hybrid Code Division and Time Division Multiple Access (CDMA/TDMA) Communications Network with Localization via Two-Way Ranging and Direction of Arrival

by Michael L Don

Approved for public release: distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



A Hybrid Code Division and Time Division Multiple Access (CDMA/TDMA) Communications Network with Localization via Two-Way Ranging and Direction of Arrival

Michael L Don

***Weapons and Materials Research Directorate,
DEVCOM Army Research Laboratory***

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) April 2021		2. REPORT TYPE Technical Report		3. DATES COVERED (From - To) January–August 2020	
4. TITLE AND SUBTITLE A Hybrid Code Division and Time Division Multiple Access (CDMA/TDMA) Communications Network with Localization via Two-Way Ranging and Direction of Arrival				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Michael L Don				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) DEVCOM Army Research Laboratory ATTN: FCDD-RLW-LF Aberdeen Proving Ground, MD 21005-5066				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-9184	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release: distribution unlimited.					
13. SUPPLEMENTARY NOTES ORCID ID: Michael L Don, 0000-0002-8021-9066					
14. ABSTRACT This report develops a hybrid code division and time division multiple access (CDMA/TDMA) network model for localization via two-way ranging and direction of arrival (DoA). CDMA allows multiple agents to transmit simultaneously on the same frequency, increasing the measurement rate. However, the noise generated by simultaneous transmissions can negatively impact the probability of dropped messages as well as the ranging error variance. These competing factors are combined into a normalized ranging variance, providing a single metric to evaluate performance. Previous DoA research using two frequencies is expanded to include multiple frequencies. A statistical DoA model is presented that optimally combines DoA measurements. These models will be a valuable tool for research and development of high dynamic vehicles in GPS-denied environments.					
15. SUBJECT TERMS wireless networks, localization, radio frequency ranging, multiple access, two-way ranging					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 66	19a. NAME OF RESPONSIBLE PERSON Michael L Don
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 410-306-0775

Contents

List of Figures	iv
List of Tables	v
1. Introduction	1
2. A Hybrid CDMA/TDMA Network	2
3. Ranging Variance	4
4. Probability of Dropped Messages	6
5. Inverse Models	7
6. Normalized Variance	8
7. Example with Unequal Power Levels	10
8. Direction of Arrival	11
8.1 Direction of Arrival with Two Frequencies	11
8.2 Direction of Arrival with More than Two Frequencies	17
8.3 Statistical Model	21
9. Conclusion	23
10. References	24
Appendix. MATLAB Scripts	27
List of Symbols, Abbreviations, and Acronyms	58
Distribution List	59

List of Figures

Fig. 1	Total swarm update rate vs.	3
Fig. 2	Ranging variance vs. distance for given values of K	5
Fig. 3	Probability of a dropped range or broadcast vs. distance for given values of K	6
Fig. 4	Demonstration of inverse models. The distance is found given p_b or p_r	7
Fig. 5	Error vs. the number of iterations for the algorithm to find distance given p_r	8
Fig. 6	Normalized variance $\hat{\sigma}_r^2$ vs. distance for various values of K	9
Fig. 7	K with the minimum normalized variance $\hat{\sigma}_r^2$ (top), and the difference between the $\hat{\sigma}_r^2$ and the minimum $\hat{\sigma}_r^2$ for the given values of K	9
Fig. 8	Comparison of the probability of a dropped range (top) and ranging variance (bottom) using CDMA with equal and unequal signal power levels	10
Fig. 9	DoA system with two antennas.....	12
Fig. 10	Example of phase difference measurements using two frequencies...	14
Fig. 11	Example phase line plot with phase difference measurements plotted as phase points	15
Fig. 12	Unwrapping phase measurements for the first wavelength (top) and second wavelength (bottom)	15
Fig. 13	Recovered angle vs. the true angle (top) and the recovered angle error (bottom) for noisy phase measurements	16
Fig. 14	Minimum phase line spacing vs. λ_2 (top), and the corresponding maximum integer ambiguity values (bottom).....	16
Fig. 15	Example of phase difference measurements using three frequencies.	18
Fig. 16	Example phase line plot with phase difference measurements plotted as phase points using three frequencies	19
Fig. 17	Unwrapping phase measurements for three frequencies	19
Fig. 18	Recovered angle vs. the true angle (top), and the recovered angle error (bottom) for three frequencies	20
Fig. 19	Minimum phase line spacing vs. λ_3 (top), and the corresponding maximum integer ambiguity values (bottom).....	20
Fig. 20	The SNR vs. distance for the three wavelengths used in the preceding simulations (top), and variance of the angle measurement for each σ_i^2 and σ_w^2 for $\theta = 30^\circ$ (bottom).....	21

Fig. 21	Average RMSE of θ using each wavelength independently, the average value of θ from the three wavelengths using equal weights, and the inverse-variance weighted average	23
---------	---	----

List of Tables

Table 1	Example network schedule for Agent 1	3
Table 2	Simulation parameters	4

1. Introduction

Collaborative swarms exhibit advantages over a comparable group of solitary agents. One advantage is the ability for swarm agents to localize relative to the group using spatial relationships between many agents to achieve accurate relative position information.¹ This is particularly important in GPS-denied environments where there are limited positioning options.² Many applications exist for relative positioning such as collision avoidance,³ formation flying,⁴ and patterned weapon delivery.⁵ In addition, relative localization can be transformed into absolute localization even if the absolute positions of only a few agents are known. Previous research developed a simple swarm network model using two-way ranging (TWR) and time division multiple access (TDMA) for swarm relative localization.⁶⁻⁸ RF direction of arrival (DoA) was also investigated as a technology for swarm localization.⁹ Unlike many common applications, high dynamic platforms require a fast measurement rate for accurate guidance, navigation, and control. In this report, a network scheme that combines code division multiple access (CDMA) with TDMA is developed to increase the measurement rate. In addition, previous DoA research is expanded to use multiple frequencies. These models will be a valuable tool for the research and development of high dynamic swarms.

This report is organized as follows. In Section 2, a hybrid CDMA/TDMA network is proposed. The measurement update rate is evaluated as a function of the number of agents transmitting simultaneously. In Section 3, a statistical model of the ranging error is presented. In Section 4, a separate statistical model is developed to characterize the probability of dropped range and communication messages. In Section 5, inverse models are derived. An inverse range error model is used by state estimation algorithms to estimate ranging variance, while the inverse of the dropped message model can help estimate range even when messages are dropped. In Section 6, a normalized ranging variance is derived that incorporates ranging variance, dropped messages, and the measurement update rate in order to evaluate network performance. In Section 7, an example of CDMA using unequal signal power is presented, highlighting the importance of gain control in CDMA networks. Last, in Section 8, previous research performing DoA with two frequencies is reviewed and expanded to include more than two frequencies. Measurements from multiple frequencies are combined intelligently to minimize error, and a statistical DoA model is derived.

2. A Hybrid CDMA/TDMA Network

In TWR, Agent 1 sends a ranging message of b_r bits to Agent 2 at a rate of r bps. After a ranging processing delay of δ_1 seconds, Agent 2 sends a message back to Agent 1, which calculates the total message time of flight t_f . The distance between agents is $d = c(t_f - \delta_1)/2$, where c is the speed of light. Using direct sequence CDMA with a code of length C , K agents can transmit simultaneously. Additionally, there is a TDMA schedule where each agent takes turns ranging to the other agents. Thus, only the return ranging messages involve simultaneous transmissions that require CDMA. Assuming there is another ranging processing delay of δ_2 s after the return message, the total ranging time for an agent to range to K other agents is

$$t_r = t_1 + t_2, \quad (1)$$

where $t_1 = rb_r + \delta_1$ is the time of the outgoing non-CDMA message to be transmitted and processed, and $t_2 = rb_rC + \delta_2$ is the time for the incoming CDMA message to be transmitted and processed. Here the non-CDMA bit rate r and the CDMA chip rate are assumed to be the same. Given N agents, each agent performs $n_r = \lceil (N - 1)/K \rceil$ ranging operations, with $\lceil \cdot \rceil$ indicating the ceiling function.

After finishing its range measurements, each agent broadcasts all of its intra-swarm information, including timing and alternative localization data. There are $M = N(N - 1)$ possible range measurements. Given a word width of w bits, the total broadcast message length is approximated as $b_b = 6wM$. Assuming each broadcast message is divided into n_b transmissions, and a processing delay of δ_b for each transmission, the total broadcast time for each agent is $t_b = n_b t_t$, where $t_t = b_t/r + \delta_b$ is the time of one broadcast transmission and $b_t = b_b/n_b$. This gives a total swarm period of

$$t_s = N(t_r n_r + t_b) \quad (2)$$

for all the agents to complete one round of their ranging and broadcast operations. To simplify simulation, all processing delays are chosen so that t_1 , t_2 , and t_t are multiples of a constant time step. DoA measurements are addressed later and do not require their own dedicated messages.

Table 1 shows an example schedule for the first agent with $N = 8$, $K = 3$, and $n_b = 4$. Agent 1 transmits an outgoing ranging message to the other agents. Since only a maximum of three agents can respond simultaneously, it takes $n_r = 3$ TWR operations for all seven other agents to respond. After Agent 1 has finished ranging to all of the other agents, it begins its broadcast messages. The total message of b_b

bits is divided into n_b parts, and sent in n_b separate broadcast messages. After Agent 1 completes its transmissions, each of the other agents take their turn performing the same operations. In this case, the total swarm period will be $t_s = N(n_r(t_1 + t_2) + n_b t_t) = 8(3(t_1 + t_2) + 4t_t)$. Assuming $t_1 = t_2 = t_t$ is a constant time step, this gives a total swarm period of 80 time steps.

Table 1 Example network schedule for Agent 1

Step	Schedule	Comments
1	Agent 1 transmits range message	Outgoing non-CDMA range message
2	Agents 2, ..., 4 transmits reply range message	K incoming CDMA range messages
3	Agent 1 transmits range message	Outgoing non-CDMA range message
4	Agents 5, ..., 7 transmits reply range message	K incoming CDMA range messages
5	Agent 1 transmits range message	Outgoing non-CDMA range message
6	Agents 8 transmits reply range message	Remaining incoming range messages
7	Agent 1 transmits $\frac{1}{4}$ of broadcast message	First of n_b broadcasts
8	Agent 1 transmits $\frac{1}{4}$ of broadcast message	Second of n_b broadcasts
9	Agent 1 transmits $\frac{1}{4}$ of broadcast message	Third of n_b broadcasts
10	Agent 1 transmits $\frac{1}{4}$ of broadcast message	Fourth of n_b broadcasts

Figure 1 shows the total swarm update rate $r_s = 1/t_s$ for a range of K values. The simulation parameters used are shown in Table 2. This table also includes additional parameters used later in this report. Discontinuities in the curve are due to the ceiling function in the n_r calculation. The MATLAB scripts to generate all of the figures in this report are included in the Appendix.

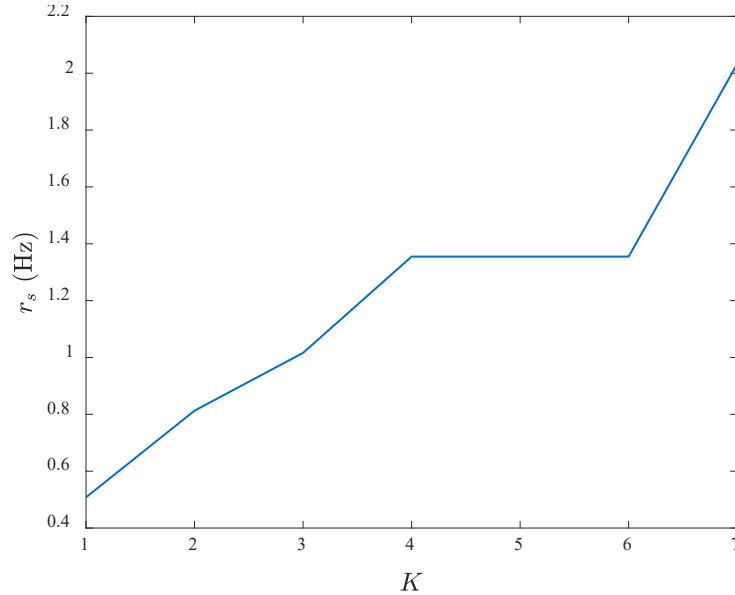


Fig. 1 Total swarm update rate vs. K

Table 2 Simulation parameters

Parameter	Value	Description
N	8	Number of agents
C	32	CDMA code length
r	500 Kbps	Bit rate
w	16	Bits per word
n_b	4	Number of transmissions per broadcast
δ_1	15.04 ms	Outgoing processing delay
δ_2	4.624 ms	Incoming processing delay
f	6 GHz	RF
P_t	-17 dBW	Transmit power
T	3000 K	System noise temperature
b_r	168	Bits per ranging message
b_t	1344	Bits per broadcast message

3. Ranging Variance

The variance of one-way range measurements in meters squared is bounded by the Cramér–Rao lower bound¹⁰

$$\sigma^2 \geq \frac{c^2}{(2\pi B)^2 b_r E_b / N_0} \left(1 + \frac{1}{\frac{b_r E_b}{N_0}} \right), \quad (3)$$

where B is the system bandwidth and E_b/N_0 is the bit-energy-to-noise-density ratio. It is assumed $B = 2r$ throughout this report. For the outgoing non-CDMA ranging transmissions, the variance σ_1^2 is calculated using E_b/N_0 defined in terms of the signal-to-noise ratio (SNR)

$$\frac{E_b}{N_0} = \frac{r}{BSNR}. \quad (4)$$

For the incoming CDMA ranging transmissions, the variance σ_2^2 is calculated using E_b/N_0 with an additional CDMA noise term¹¹

$$\frac{E_b}{N_0} = \frac{1}{\frac{r}{BSNR} + \frac{2}{3C} \sum_{i=1}^{K-1} \frac{P_i}{P_0}}, \quad (5)$$

where P_i/P_0 is ratio of the power at the receiver of signal i to the power of the signal of interest P_0 , and assuming that the code has basic randomness properties. If it is assumed that the received power from the K CDMA signals are equal, this becomes

$$\frac{E_b}{N_0} = \frac{1}{\frac{r}{B\text{SNR}} + \frac{2(K-1)}{3C}}. \quad (6)$$

Although this assumption is generally not true for arbitrary networks, transmit power gain control can be used equalize power levels.

Using the free space path loss model¹²

$$\text{FSPL} = \left(\frac{4\pi df}{c} \right)^2, \quad (7)$$

and the noise power at the receiver

$$P_n = kTB, \quad (8)$$

the SNR is

$$\text{SNR} = \frac{P_t}{P_n \text{FSPL}}, \quad (9)$$

with Boltzmann constant k , total noise temperature of the receiver T , transmit power P_t , carrier frequency f Hz, and distance d meters. The total ranging variance is

$$\sigma_r^2 = \sigma_1^2 + \sigma_2^2. \quad (10)$$

Figure 2 shows σ_r^2 versus distance for a range of K values assuming equal power levels.

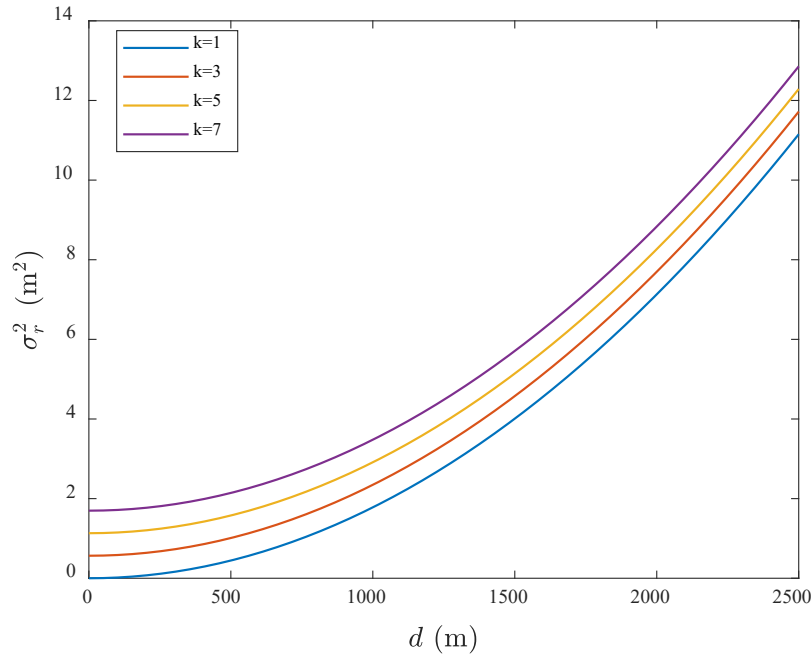


Fig. 2 Ranging variance vs. distance for given values of K

4. Probability of Dropped Messages

Assuming a binary phase shift key modulation, the probability of a bit error is¹¹

$$p = Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \quad (11)$$

Denote p_1 and p_2 to be the outgoing and incoming bit error probabilities, respectively, using the appropriate value of E_b/N_0 . Assuming that the ranging message is verified by both agents and dropped if there is one or more bit errors, the total probability of a dropped ranging measurement is

$$p_r = 1 - (1 - p_1)^{b_r}(1 - p_2)^{b_r}. \quad (12)$$

The same equations can be used to calculate the bit error probability for a non-CDMA broadcast transmission

$$p_b = 1 - (1 - p_1)^{b_t}. \quad (13)$$

Figure 3 shows p_r for a range of K values, and p_b , which is independent of K . This simulation assumes equal power levels.

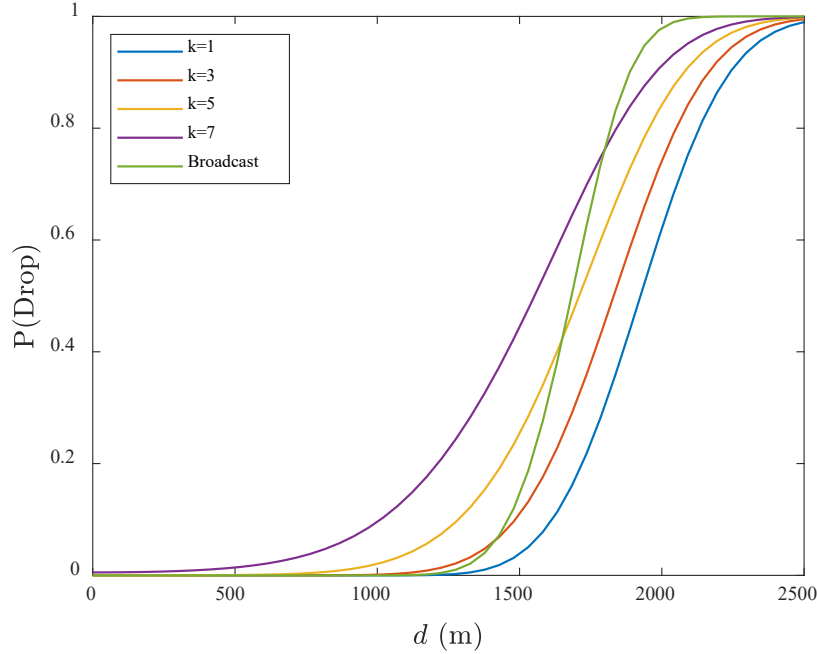


Fig. 3 Probability of a dropped range or broadcast vs. distance for given values of K

5. Inverse Models

State estimation algorithms use an inverse range error model to estimate ranging variance, while the inverse of the dropped message model can help estimate ranges even when the messages are dropped. Distance can be determined given p_b as follows:

$$p_t = 1 - (1 - p_b)^{\left(\frac{1}{b_t}\right)} \quad (14)$$

$$\frac{E_b}{N_0} = Q^{-1}(2p_t)^2 \quad (15)$$

$$\text{SNR} = \frac{r}{B} \left(\frac{E_b}{N_0} \right) \quad (16)$$

$$\text{FSPL} = \frac{P_t}{P_n \text{SNR}} \quad (17)$$

$$d = \frac{c\sqrt{\text{FSPL}}}{4\pi f} \quad (18)$$

It is difficult to find a closed-form expression for distance given p_r , but it can be easily solved numerically. The MATLAB code in the Appendix implements a binary search algorithm for a given p_r . Figure 4 plots the distances at which $p_b = 0.5$ and $p_r = 0.5$, determined through the inverse models. Figure 5 shows the average error versus the number of iterations curve for 1000 trials of the binary search algorithm used to determine the distance of a given p_r . For unknown reasons there is a consistent dip in error on the sixth iteration. This requires additional study.

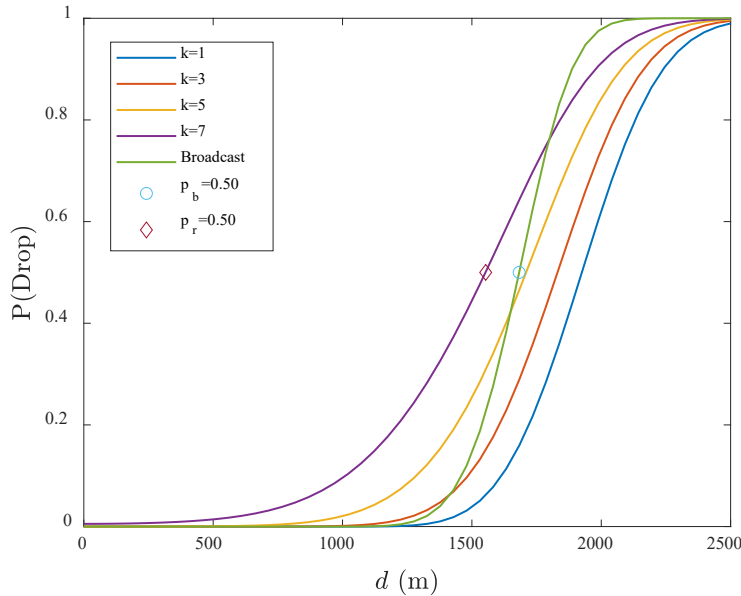


Fig. 4 Demonstration of inverse models. The distance is found given p_b or p_r .

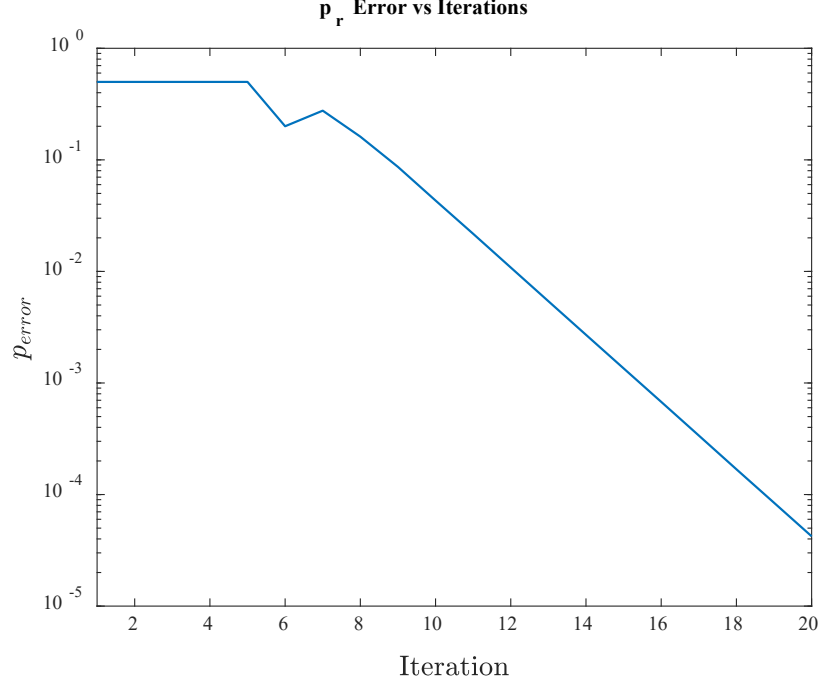


Fig. 5 Error vs. the number of iterations for the algorithm to find distance given p_r

6. Normalized Variance

Comparing the performance of different values of K is not straightforward. Higher values of K lead to faster measurement rates, but the additional noise degrades performance. However, faster rates produce more measurements, which can be averaged to decrease error variance, and produce more broadcasts, which increases the rate of successful transmissions. To create a basic comparison of different values of K , a normalized variance, $\hat{\sigma}_r^2$ can be computed as

$$\hat{\sigma}_r^2 = \frac{\sigma_r^2}{t_n r_s (1-p_r)(1-p_b)}. \quad (19)$$

The denominator is the expected number of successful range broadcasts to a single agent for a given normalization time t_n . $\hat{\sigma}_r^2$ then represents the variance of the average of the expected number of measurements over t_n . For large distances, where it is likely no ranging measurements will be received, $\hat{\sigma}_r^2$ will be very large. For small distances, where multiple measurements can be averaged, $\hat{\sigma}_r^2$ will be small. Figure 6 compares $\hat{\sigma}_r^2$ versus distance, where p_b is calculated using the same distance as p_r and $t_n = 1$ s. Observe that for different distance intervals, different values of K have the best performance. This is highlighted in Fig. 7, where the top plot shows the K with the minimum $\hat{\sigma}_r^2$ versus distance, and the bottom plot shows the difference in $\hat{\sigma}_r^2$ between the minimum K and the other selected values of K .

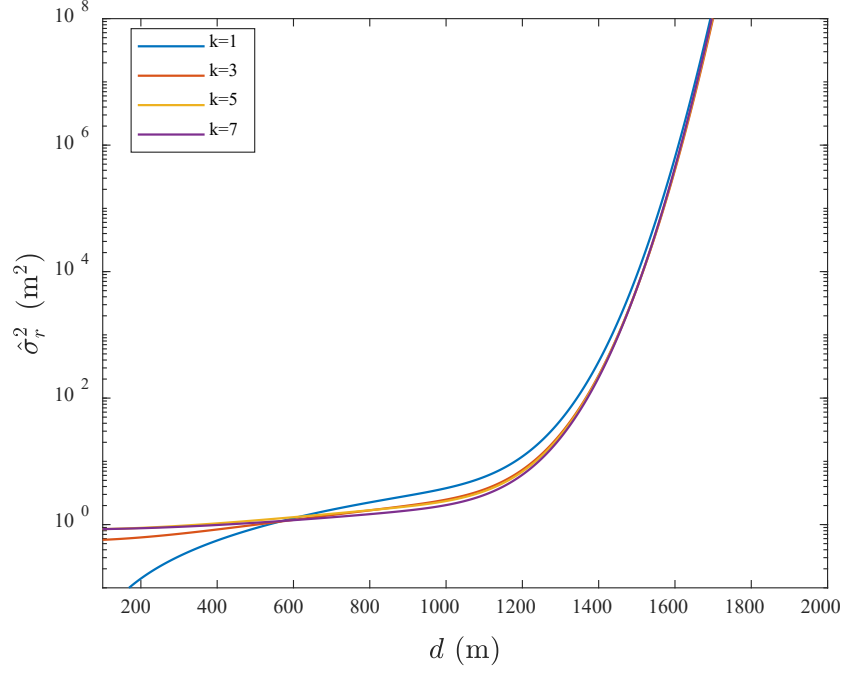


Fig. 6 Normalized variance $\hat{\sigma}_r^2$ vs. distance for various values of K

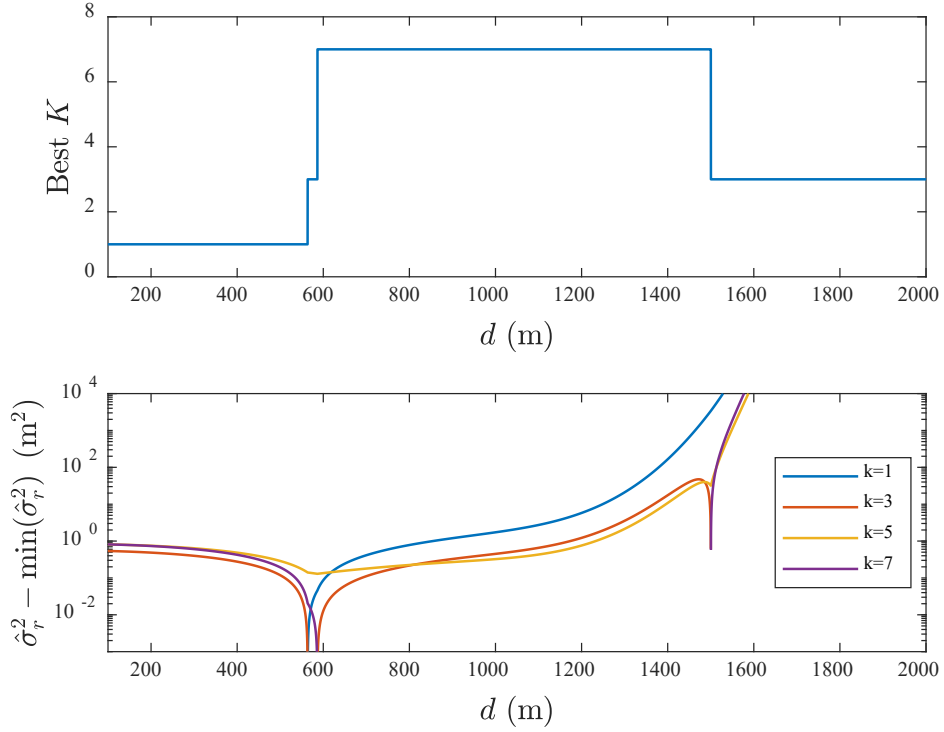


Fig. 7 K with the minimum normalized variance $\hat{\sigma}_r^2$ (top), and the difference between the $\hat{\sigma}_r^2$ and the minimum $\hat{\sigma}_r^2$ for the given values of K

7. Example with Unequal Power Levels

So far, simulation results showed CDMA with equal power levels. Figure 8 shows an example comparison of the probability of a dropped ranging measurement, p_r , and the ranging variance, σ_r^2 , for equal and unequal power levels. Equal power levels use Eq. 6 to calculate E_b/N_0 , while unequal levels use Eq. 5. The agents are arranged in a cube, with the cube edges growing from 0 to 2000 m, giving a maximum distance between agents of 3464 m. Results are shown for the transmissions between the furthest spaced agents with $K = 7$. The unequal power levels are completely determined by the spacing between agents: no transmitter gain control has been applied. When the two communicating agents are next to each other, the other transmitters are close by as well, creating interference that results in a p_r close to 0.5 from the outset. As the vertices of the cube grow, the power of the interference signals decreases, but the power of the agents communicating decreases as well, resulting in a gradual increase in p_r with distance. The unequal power case shows a significant degradation in performance compared with the case of equal power levels, demonstrating the importance on implementing a gain control strategy for CDMA networks.

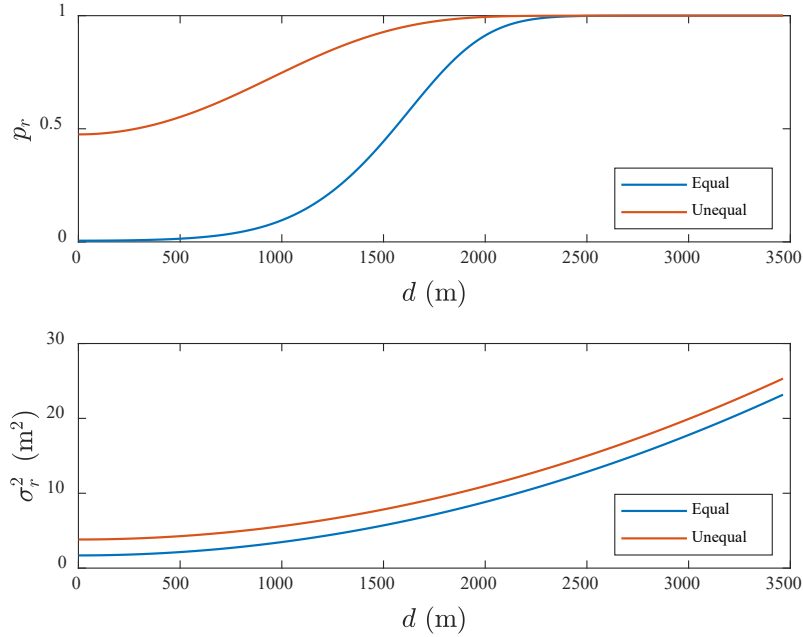


Fig. 8 Comparison of the probability of a dropped range (top) and ranging variance (bottom) using CDMA with equal and unequal signal power levels

8. Direction of Arrival

8.1 Direction of Arrival with Two Frequencies

Swarm relative localization can be achieved by intra-swarm ranges alone, but pointing vectors must also be known to implement formation control. In addition, angle information can be combined with ranging measurements to increase localization accuracy.¹³ For small platforms, large antenna arrays are not practical. Therefore, the discussion here is limited to antenna arrays of two elements. Two-dimensional DoA can be performed using two arrays, one oriented in azimuth and the other in elevation.¹⁴ Furthermore, rather than using additional antennas to provide more DoA information, multiple frequencies can be employed instead. Traditional radio technology operated at predefined, narrow frequency bands. With the proliferation of software-defined radio (SDR), platforms capable of operating at a wide range of frequencies have become easily available. Leveraging prior experience with SDRs,^{15–17} research begun using two frequencies for DoA.⁹ This research is reviewed here and expanded to include multiple frequencies, an optimal approach to average multiple measurements, and a DoA statistical model. Non-CDMA transmissions in the existing communications schedule outlined in Section 2 can be used for DoA measurements.¹⁸

Figure 9 shows a basic DoA system. A_1 and A_2 indicate the position of two antennas spaced a distance d apart. The red lines indicate the RF signal's direction of propagation from the source to the antennas. If the distance from the signal source to the antennas is much greater than d , these lines can be assumed to be parallel. The angle of arrival, θ , is shown as the angle off center of the incoming RF signal that intersects A_2 . Given a RF signal with wavelength λ , assuming $d \leq \lambda/2$ and $-90^\circ \leq \theta \leq 90^\circ$, the phase difference between the received signals at A_1 and A_2 is given by

$$\Delta\phi = 2\pi \left(\frac{d \sin \theta}{\lambda} \right), \quad (20)$$

where $-\pi \leq \Delta\phi \leq \pi$. Here and in the rest of the paper, phases are in radians while angles are in degrees. Given a measured phase difference between A_1 and A_2 , the angle of arrive is then

$$\theta = \sin^{-1} \left(\frac{\lambda \left(\frac{\Delta\phi}{2\pi} \right)}{d} \right). \quad (21)$$

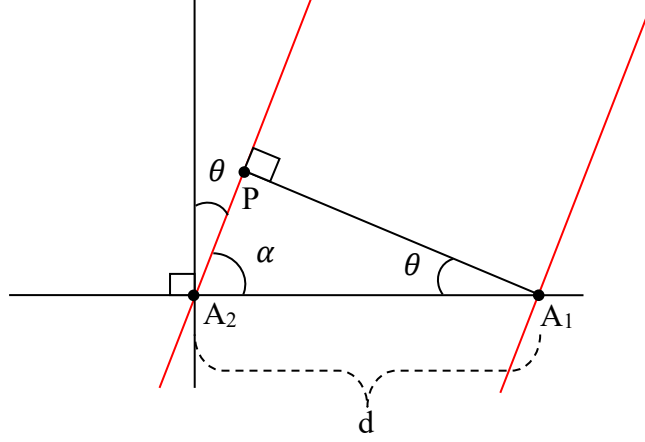


Fig. 9 DoA system with two antennas

From Eq. 20, if $d > \lambda/2$, $\Delta\phi$ can exceed the range of $-\pi \leq \Delta\phi \leq \pi$. This causes an ambiguity problem because values of $\Delta\phi$ and $\Delta\phi \pm 2\pi$ cannot be distinguished from each other at the antennas, leading to multiple solutions for θ . Many systems solve this problem by restricting $d < \lambda/2$, but since the variance of the angle error is proportional to $(1/d)^2$, it is desirable to make d as large as possible.¹⁹ This consideration, as well as external factors that can limit the choice of λ , motivate developing strategies for systems where $d > \lambda/2$. One strategy to solve the ambiguity problem is to add an integer I to Eq. 21 to account for the phase roll-overs.

$$\theta = \sin^{-1} \left(\frac{\lambda \left(\frac{\Delta\phi}{2\pi} + I \right)}{d} \right). \quad (22)$$

For a given range of $-\theta_{max} \leq \theta \leq \theta_{max}$, the range of I will be $-I_{max} \leq I \leq I_{max}$, with I_{max} given as

$$I_{max} = \left\lfloor \frac{d \sin \theta_{max} + \lambda/2}{\lambda} \right\rfloor, \quad (23)$$

with $\lfloor x \rfloor$ indicating the floor of x . A typical method to determine I is to use three antennas, with distances d_{12} and d_{13} denoting the spacing between the first and second and first and third antennas, respectively, leading to two phase measurements, $\Delta\phi_{12}$ and $\Delta\phi_{13}$.²⁰ This approach is necessary in many DoA applications where the direction-finding system has no control over the signal being detected. However, in this application, where each agent can be controlled, an

alternative way of resolving the ambiguity presents itself. Instead of using two different antenna spacings, two different transmit frequencies can be used, denoted λ_1 and λ_2 .⁹ Rearranging Eq. 22 for each pair of antennas gives

$$\Delta\phi_1 - 2\pi I_1 = 2\pi \left(\frac{d \sin \theta}{\lambda_1} \right) \quad (24)$$

and

$$\Delta\phi_2 - 2\pi I_2 = 2\pi \left(\frac{d \sin \theta}{\lambda_2} \right). \quad (25)$$

Solving for $\Delta\phi_2$ in terms of $\Delta\phi_1$ gives

$$\Delta\phi_2 = \frac{\lambda_1}{\lambda_2} \Delta\phi_1 - \frac{\lambda_1}{\lambda_2} 2\pi I_1 + 2\pi I_2. \quad (26)$$

Equation 26 indicates a linear relationship between the phase measurements $\Delta\phi_1$ and $\Delta\phi_2$ for each possible combination of I_1 and I_2 , which are referred to as phase lines. For a given measurement $(\Delta\phi_1, \Delta\phi_2)$ point, I_1 and I_2 can be determined by identifying the phase line closest to the measurement in 2-D $\Delta\phi_1 \Delta\phi_2$ space. The distance δ from point $(\Delta\phi_1, \Delta\phi_2)$ to the phase lines in Eq. 26 is

$$\delta = \frac{\left| \Delta\phi_2 - \frac{\lambda_1}{\lambda_2} \Delta\phi_1 + \frac{\lambda_1}{\lambda_2} 2\pi I_1 - 2\pi I_2 \right|}{\sqrt{\left(\frac{\lambda_1}{\lambda_2} \right)^2 + 1}}. \quad (27)$$

Once the closest phase line is determined for a given phase point, the corresponding values of I_1 and I_2 can be inserted into Eqs. 24 and 25 to solve for θ .

The spacing between two phase lines, ρ , with integer phase corrections I_{11} and I_{12} for the first line, and I_{21} and I_{22} for the second line, is given by

$$\rho = \frac{\left| \frac{\lambda_1}{\lambda_2} 2\pi(I_{21} - I_{11}) + 2\pi(I_{12} - I_{22}) \right|}{\sqrt{\left(\frac{\lambda_1}{\lambda_2} \right)^2 + 1}}. \quad (28)$$

To make the system more robust to noise, λ_1 and λ_2 can be chosen to maximize the spacing between phase lines.²¹

An example scenario showing values of $\Delta\phi_1$ and $\Delta\phi_2$ for $-50^\circ \leq \theta \leq 50^\circ$ is shown in Fig. 10 for $d = 26.6$ cm, $\lambda_1 = 12.5$ cm, and $\lambda_2 = 15.625$ cm. Gaussian noise with $\sigma^2 = 0.03$ rad² was added directly to the $\Delta\phi$ measurements. In this case, $-2 < I_1 < 2$ and $-1 < I_2 < 1$ for $-50 \leq \theta \leq 50$, resulting in the phase lines shown in Fig. 11. One-hundred phase measurements from Fig. 10 are plotted as $(\Delta\phi_1, \Delta\phi_2)$ phase points that occur near their respective phase lines. The phase line closest to each phase point determines its value of I_1 and I_2 , which can then be inserted into Eqs. 24 and 25 to unwrap the phase measurements and determine an unambiguous value for θ . Figure 12 shows this process by displaying the original phase measurements, the correction factor of $-2\pi I$ from Eqs. 24 and 25, and the unwrapped phases for both λ_1 and λ_2 found by adding the correction factor to the phase measurements. Figure 13 shows the recovered angle versus the true angle and the recovered angle error derived from the phase measurements. As is discussed later in the report, the angle variance is proportional to λ^2 ; therefore, the λ_2 measurements are more noisy than λ_1 . The top plot of Fig. 14 uses Eq. 28 to show the minimum phase line spacing versus λ_2 for $\lambda_1 = 12.5$ cm. Multiples of λ_1 are clearly poor choices for λ_2 , creating no space between phase lines. Local maxima are preferable choices, such as $\lambda_2 = 18.75$ cm, although external factors may limit available options.

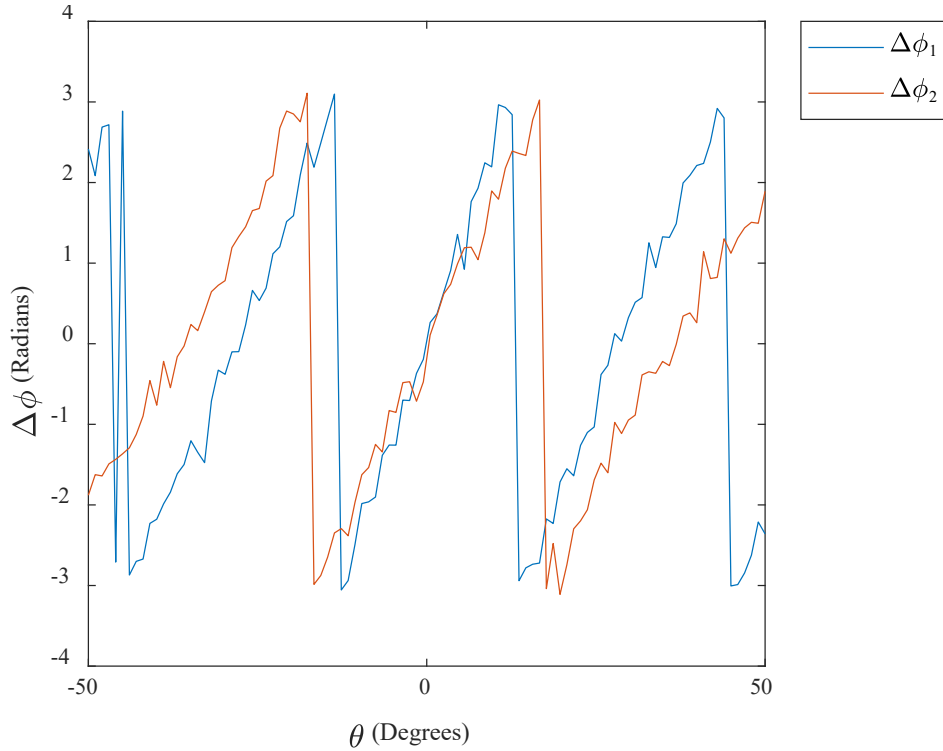


Fig. 10 Example of phase difference measurements using two frequencies

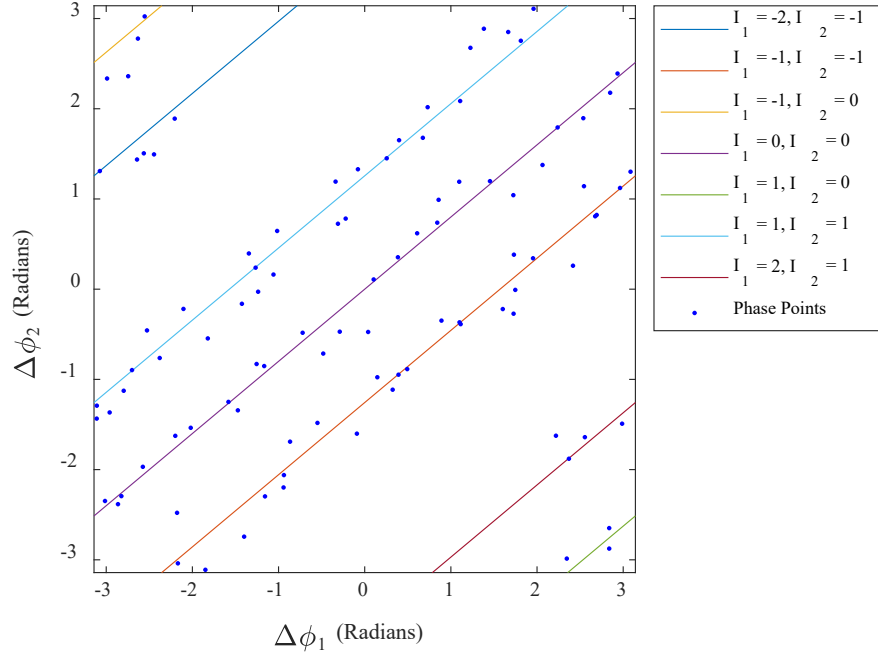


Fig. 11 Example phase line plot with phase difference measurements plotted as phase points

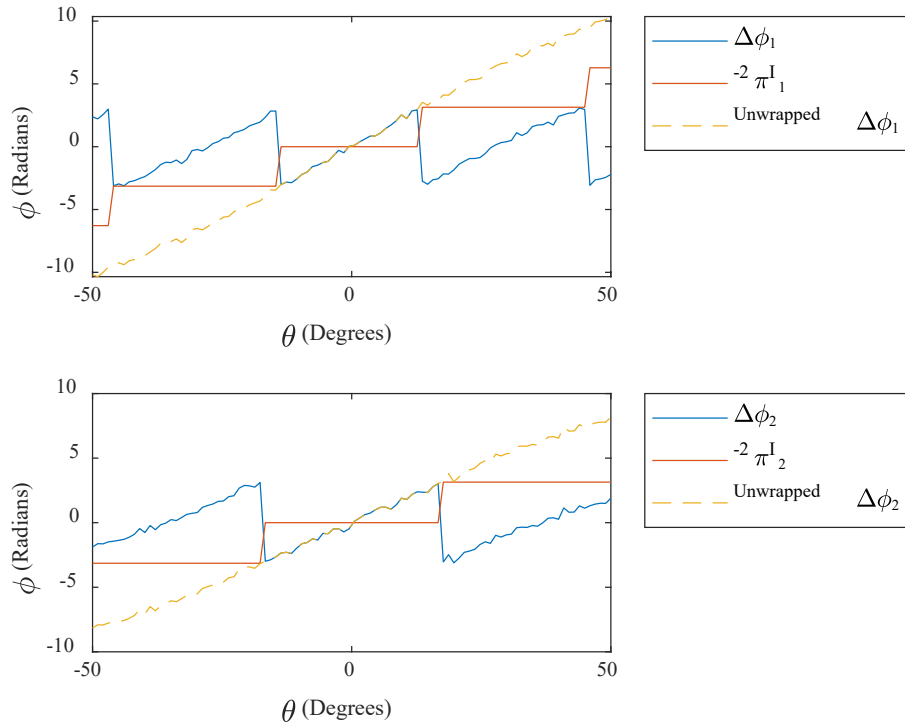


Fig. 12 Unwrapping phase measurements for the first wavelength (top) and second wavelength (bottom)

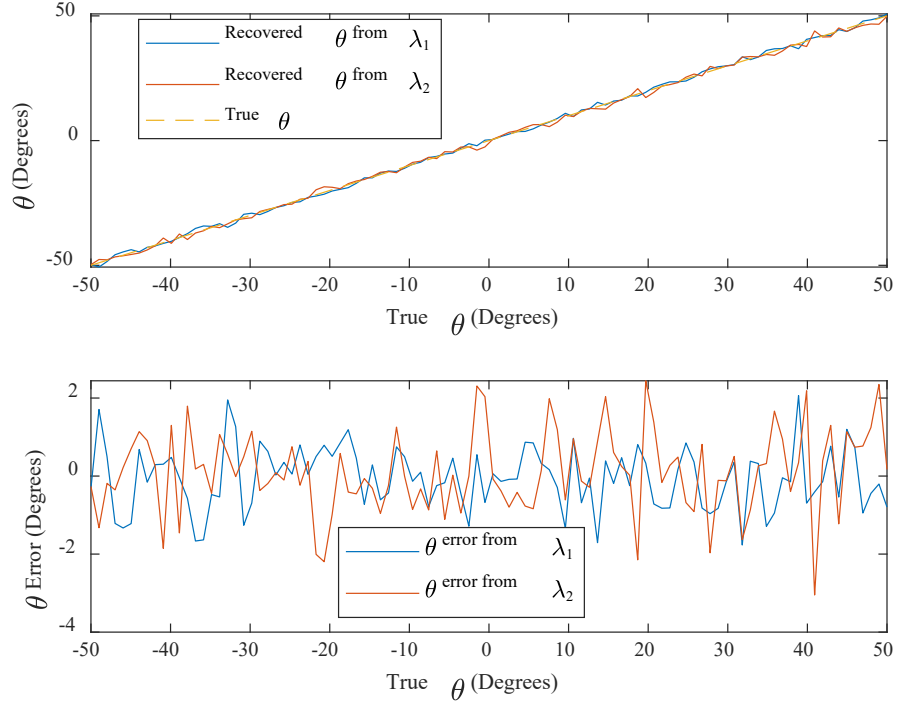


Fig. 13 Recovered angle vs. the true angle (top) and the recovered angle error (bottom) for noisy phase measurements

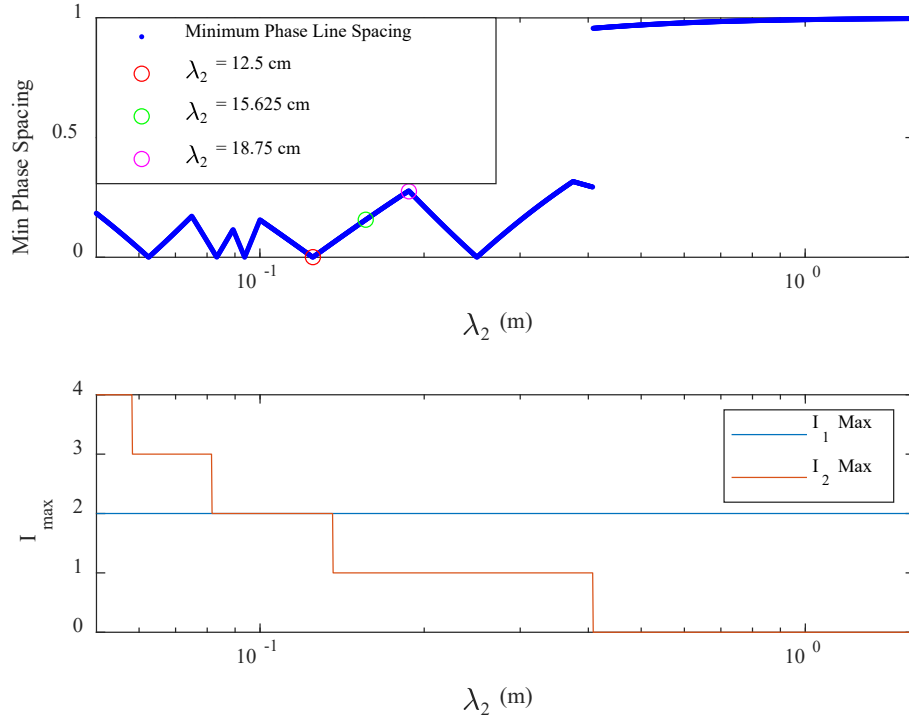


Fig. 14 Minimum phase line spacing vs. λ_2 (top), and the corresponding maximum integer ambiguity values (bottom)

8.2 Direction of Arrival with More than Two Frequencies

Using the same concept developed for two frequencies, this method can be extended to more than two frequencies. Let the vector $\boldsymbol{\phi} \in \mathbb{R}^{N_f}$ denote a phase point of dimension N_f for each of the N_f different frequencies. Phase points on the phase lines are given by

$$\boldsymbol{\phi} = \frac{1}{c} \mathbf{f} 2\pi d \sin \theta + 2\pi \mathbf{I}, \quad (29)$$

where c is the speed of light, $\mathbf{f}$ is a vector of the N_f frequencies, and $\mathbf{I}$ is a vector of N_f integer ambiguity values. Phase lines are defined by a phase point and a unit pointing vector $\mathbf{n}$. Two arbitrary phase points on the phase lines are used to define $\mathbf{n}$. $\boldsymbol{\phi}_1$ is evaluated at $\theta = 0^\circ$ and $\boldsymbol{\phi}_2$ is evaluated at $\theta = 90^\circ$, giving

$$\boldsymbol{\phi}_1 = 2\pi \mathbf{I} \quad (30)$$

and

$$\boldsymbol{\phi}_2 = \frac{1}{c} \mathbf{f} 2\pi d + 2\pi \mathbf{I}. \quad (31)$$

$\mathbf{n}$ is then

$$\mathbf{n} = \frac{\boldsymbol{\phi}_2 - \boldsymbol{\phi}_1}{\|\boldsymbol{\phi}_2 - \boldsymbol{\phi}_1\|}. \quad (32)$$

The distance from a measured point $\boldsymbol{\phi}_m$ to the phase line is²²

$$\delta = \|(\boldsymbol{\phi}_1 - \boldsymbol{\phi}_m) - ((\boldsymbol{\phi}_1 - \boldsymbol{\phi}_m) \cdot \mathbf{n})\mathbf{n}\|. \quad (33)$$

The line closest to $\boldsymbol{\phi}_m$ determines $\mathbf{I}$, which can be used to solve for θ . In this formulation, a separate estimate of θ can be calculated for each frequency, giving vector $\boldsymbol{\theta} \in \mathbb{R}^{N_f}$ calculated as

$$\boldsymbol{\theta} = \sin^{-1} \left(\frac{\text{diag}(\boldsymbol{\lambda}) \left(\frac{\boldsymbol{\phi}}{2\pi} + \mathbf{I} \right)}{d} \right). \quad (34)$$

An example scenario using three frequencies is shown in Fig. 15. $\Delta\phi_1$, $\Delta\phi_2$, and $\Delta\phi_3$ for $-50^\circ \leq \theta \leq 50^\circ$ are shown for $d = 26.6$ cm, $\lambda_1 = 12.5$ cm, $\lambda_2 = 15.625$ cm, and $\lambda_3 = 22.206$ cm. Gaussian noise with $\sigma^2 = 0.03$ rad² was added directly to the $\Delta\phi$ measurements. In this case, $-2 < I_1 < 2$, $-1 < I_2 < 1$,

and $-1 < I_3 < 1$ for $-50 \leq \theta \leq 50$, resulting in the phase lines shown in Fig. 16. One-hundred phase measurements from Fig. 15 are plotted as $(\Delta\phi_1, \Delta\phi_2, \Delta\phi_3)$ phase points, which occur near their respective phase lines. The phase line closest to each phase point determines its value of I , which is then inserted into Eq. 34 to unwrap the phase measurements and determine an unambiguous value for θ . Figure 17 shows this process by displaying the original phase measurements, the correction factor of $-2\pi I$ from Eq. 31, and the unwrapped phases for λ_1 , λ_2 , and λ_3 are found by adding the correction factor to the phase measurements. Figure 18 shows the recovered angle versus the true angle and the recovered angle error derived from the phase measurements. Equation 33 can be adapted to determine the spacing between phase lines by replacing ϕ_m with a point on another line. Using this approach, Fig. 19 shows the minimum phase line spacing versus λ_3 for $\lambda_1 = 12.5$ cm and $\lambda_2 = 15.625$ cm. Local maxima are preferable choices, such as $\lambda_2 = 22.206$ cm or $\lambda_2 = 9.66$ cm.

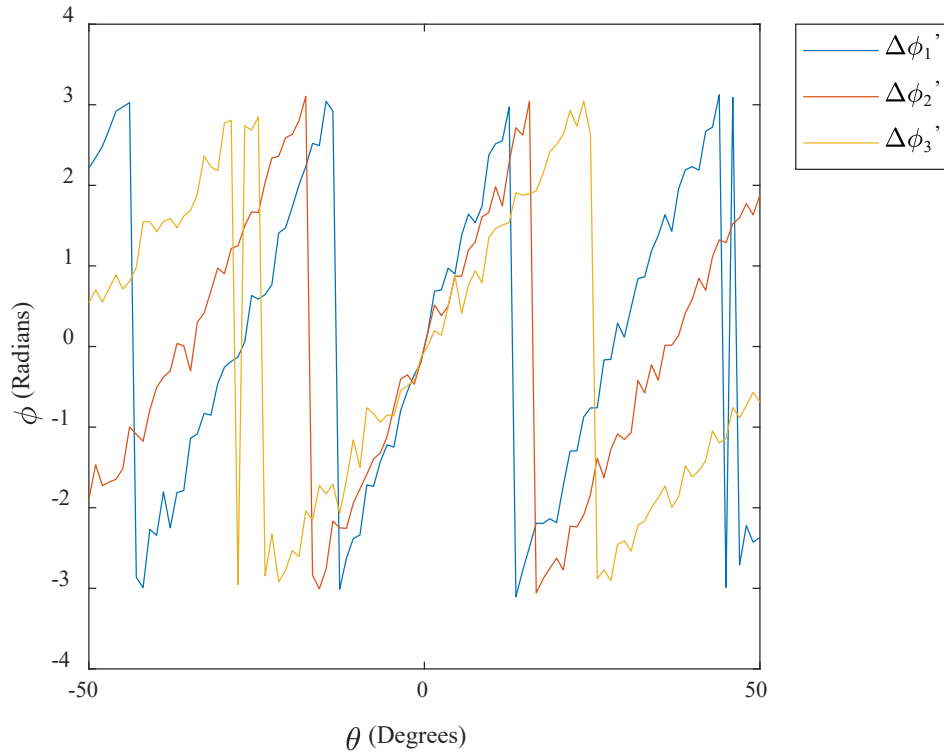


Fig. 15 Example of phase difference measurements using three frequencies

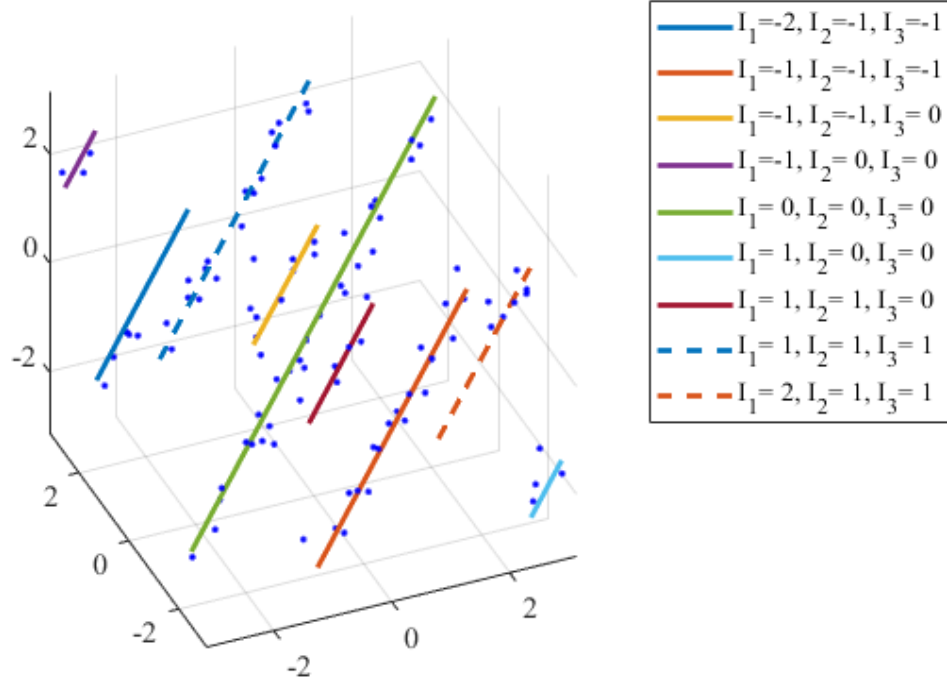


Fig. 16 Example phase line plot with phase difference measurements plotted as phase points using three frequencies

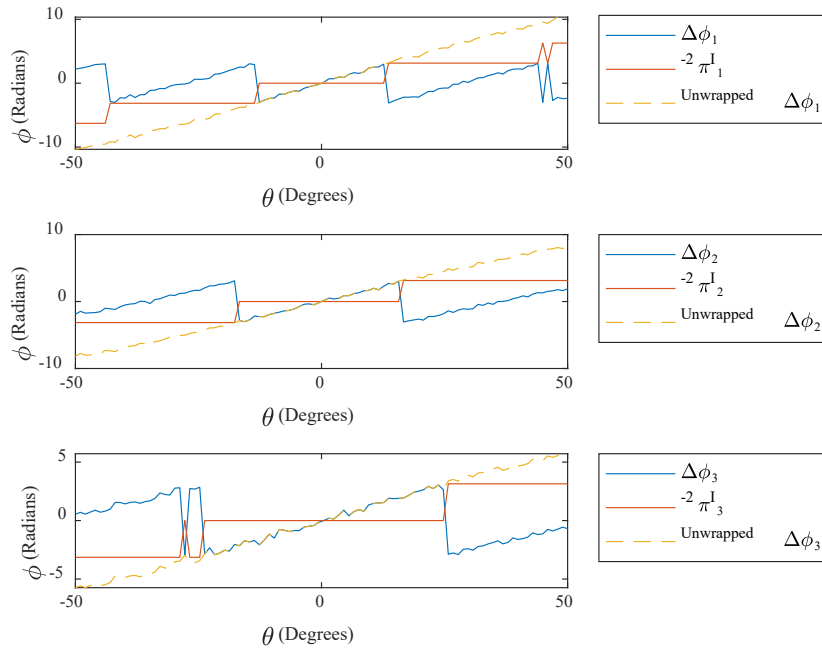


Fig. 17 Unwrapping phase measurements for three frequencies

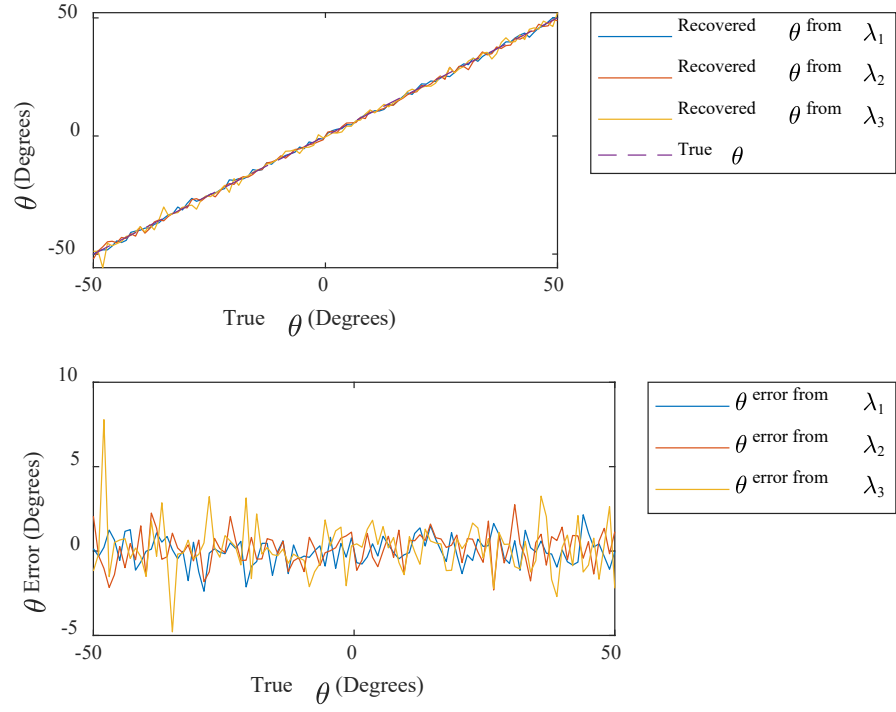


Fig. 18 Recovered angle vs. the true angle (top), and the recovered angle error (bottom) for three frequencies

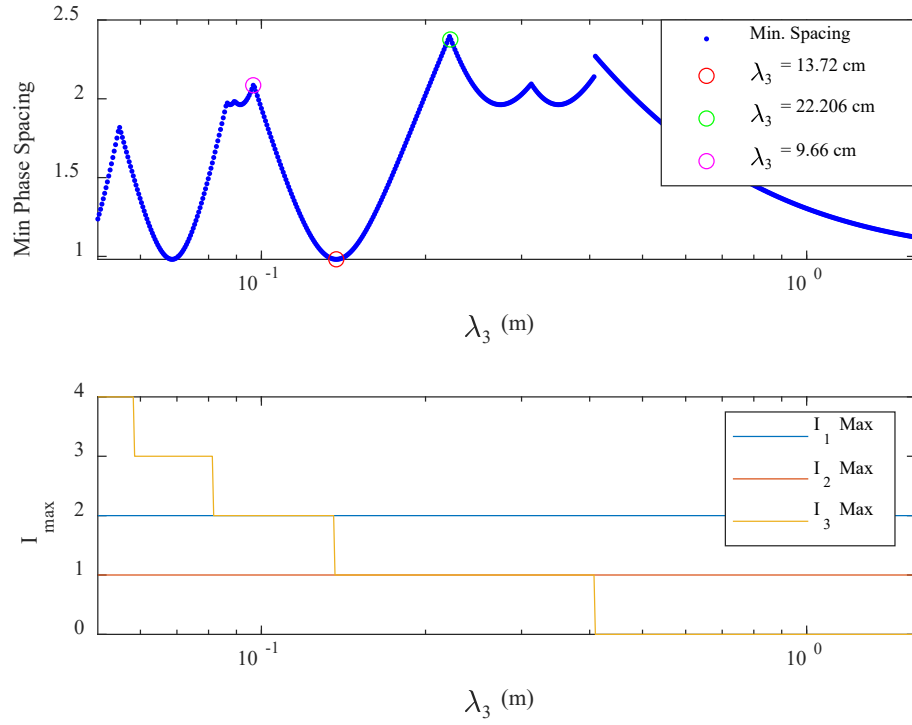


Fig. 19 Minimum phase line spacing vs. λ_3 (top), and the corresponding maximum integer ambiguity values (bottom)

8.3 Statistical Model

DoA measurements with two antennas are bounded by the Cramér–Rao lower bound¹⁹

$$\sigma^2 = \left(\frac{\lambda}{2\pi d \cos \theta} \right)^2 \frac{1}{N_\phi \text{SNR}}, \quad (35)$$

where N_ϕ is the number of phase difference measurements. Using the SNR defined in Eqs. 9 and 35, the variance for a given θ can be calculated as

$$\sigma^2 = \frac{4P_n d^2}{P_t N_\phi (d_a \cos \theta)^2}, \quad (36)$$

where the antenna spacing is denoted as d_a to differentiate from the distance between agents d . Using these simple models, the λ^2 in the SNR cancels with the λ^2 in the Cramér–Rao lower bound, making σ^2 independent of λ . Figure 20 shows an example using the parameters in Table 2, except the frequencies are the same as those used in Fig. 15. In addition, $\theta = 30^\circ$, $N_\phi = 1$, and d_a is as defined previously. The top plot shows SNR versus distance, while the bottom plot shows σ^2 for each λ and the σ^2 of the averaged measurement. As shown in Eq. 36, σ^2 is the same for each λ , making $\sigma_{Ave}^2 = \sigma^2 / N_f$.

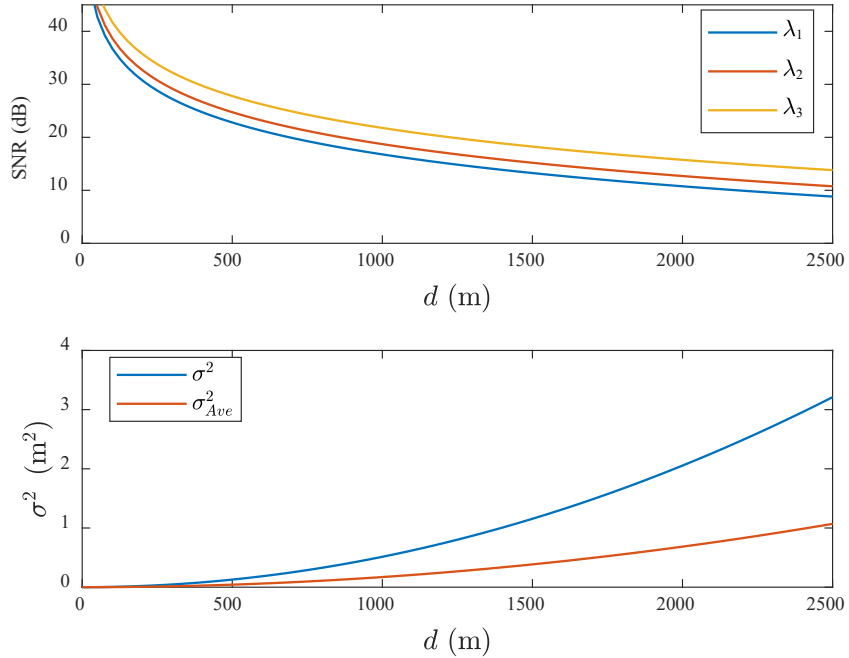


Fig. 20 The SNR vs. distance for the three wavelengths used in the preceding simulations (top), and variance of the angle measurement for each σ_i^2 and σ_w^2 for $\theta = 30^\circ$ (bottom)

Although in the simple preceding models the final σ^2 in Eq. 36 is independent of λ , alternative modeling may produce σ^2 that is dependent on λ . For example, some path loss (PL) models use a PL exponent γ to estimate attenuation due to the RF environment.

$$\text{PL} = \left(\frac{4\pi d}{\lambda} \right)^\gamma. \quad (37)$$

Using $\gamma = 3$ with Eqs. 9 and 35 gives a variance that depends on λ of

$$\sigma^2 = \frac{8\pi P_n d^3}{\lambda P_t N_\phi (d_a \cos \theta)^2}. \quad (38)$$

The optimal way to combine measurements with different variances is through inverse-variance weighting using weights²³

$$w_i = \frac{1/\sigma_i^2}{\sum_{k=1}^{N_f} 1/\sigma_k^2}, \quad (39)$$

where σ_i^2 is the variance assoiated with λ_i . Given that the only change between measurements is λ , all of the other factors cancel, giving

$$w_i = \frac{\lambda_i}{\sum_{k=1}^{N_f} \lambda_k}. \quad (40)$$

The resulting variance of the weighted average σ_w^2 will be

$$\sigma_w^2 = \frac{1}{\sum_{i=1}^{N_f} \frac{1}{\sigma_i^2}}. \quad (40)$$

Figure 21 shows the average root mean squared error (RMSE) of 10,000 runs of an example simulation using the parameters in Table 2, the indicated wavelengths, $N_\phi = 1$, $d = 750$, and $d_a = 26.6$ cm. An evenly weighted average is actually worse than the best single measurement, while the inverse-variance-weighted average provides the optimal estimate.

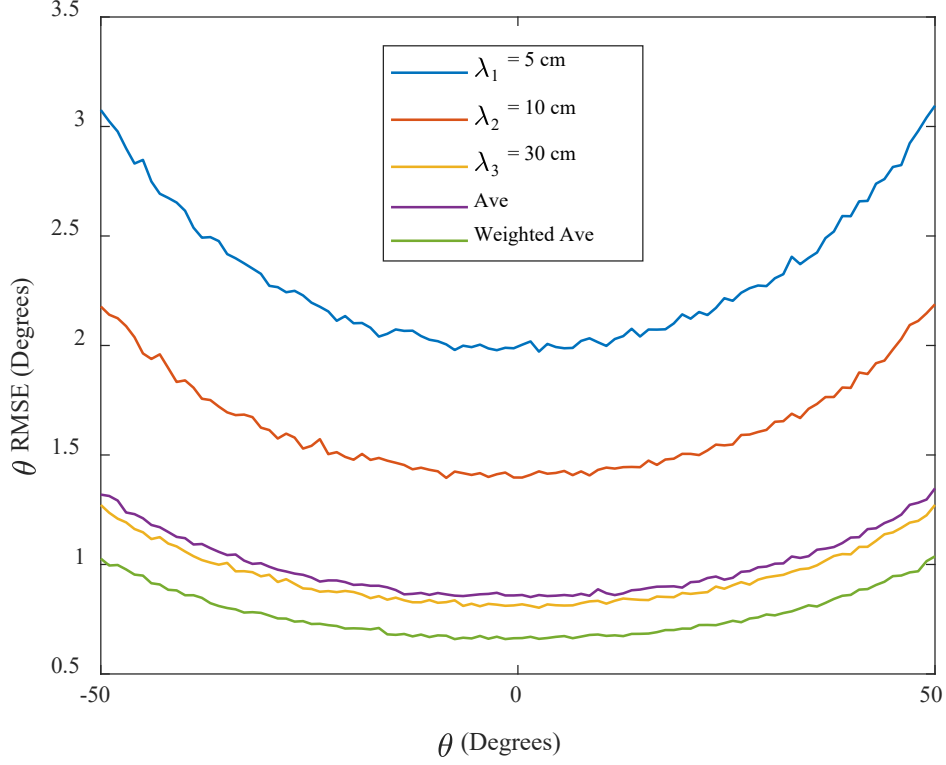


Fig. 21 Average RMSE of θ using each wavelength independently, the average value of θ from the three wavelengths using equal weights, and the inverse-variance weighted average

Although these statistical models determine the effect of signal noise on the angle error, previous research has shown that phase bias due to antenna effects can be the principle source of DoA error.⁹ This error can be removed through system calibration. Calibration acceleration techniques such as multiprobe measurement systems²⁴ and compressive sensing^{25,26} can speed measurement, making it a practical option for DoA system production.

9. Conclusion

This report has developed a hybrid CDMA/TDMA network model with localization via TWR and DoA. CDMA allows multiple agents to transmit simultaneously, increasing the measurement rate. However, the noise generated by simultaneous transmissions can negatively impact the probability of dropped messages as well as the ranging error variance. These competing factors have been combined into a normalized ranging variance, providing a single metric to evaluate performance. Previous DoA research using two frequencies has been expanded to include multiple frequencies. A statistical DoA model was presented that optimally combines DoA measurements. These models will be a valuable tool for the research and development of high dynamic vehicles in GPS-denied environments.

10. References

1. Kurazume R, Nagata S, Hirose S. Cooperative positioning with multiple robots. Proceedings of IEEE International Conference on Robotics and Automation; 1994; San Diego, CA.
2. Bachrach A, Prentice S, He R, Roy N. RANGE – robust autonomous navigation in GPS-denied environments. Journal of Field Robotics. 2011;1:28.
3. Parker R, Valae S. Vehicular node localization using received-signal-strength indicator. IEEE Transactions on Vehicular Technology. 2007;56.
4. Ryan A, Zennaro M, Howell A, Sengupta R, Hedrick JK. An overview of emerging results in cooperative UAV control. Proceedings of the 43rd IEEE Conference on Decision and Control; 2004. doi: 10.1109/CDC.2004.1428700.
5. Fresconi F, Fermen-Coker M. Delivery of modular lethality via a parent-child concept. Proceedings of the AIAA Atmospheric Flight Mechanics Conference; 2015 Feb. p. 2708.
6. Don, ML. The feasibility of radio direction finding for swarm localization. Army Research Laboratory (US); 2017 Aug. Report No. ARL-TR-8114.
7. Allik, BL, Hamaoui M, Don M, Miller C. Kalman filter aiding MDS for projectile localization. Proceedings of the AIAA Scitech Forum; 2019. p. 1159.
8. Grabner MJ, Don ML, Everson DP. Constrained geometry relative swarm localization. CCDC Army Research Laboratory (US); 2019 Dec. Report No.: ARL-TR-8877.
9. Don ML. The feasibility of radio direction finding for swarm localization. Army Research Laboratory (US); 2017 Sep. Report No.: ARL-TR-8114.
10. Van Trees HL. Detection, estimation, and modulation theory. John Wiley & Sons; 2004.
11. Dixon RC. Spread spectrum systems: with commercial applications. Vol. 994. Wiley; 1994.
12. Sklar B. Digital communications. 2nd ed. Prentice-Hall; 2001.
13. Hamaoui M. Non-iterative MDS method for collaborative network localization with sparse range and pointing measurements. IEEE Transactions on Signal Processing. 2018;67(3):568–578.

14. Doi Y, Moriya H, Ichige K, Arai H, Hayashi T, Matsuno H, Nakano M. Low-cost antenna array via antenna switching for high resolution 2-D DOA estimation. Proceedings of the InSignal Processing Systems (SiPS) IEEE Workshop; 2013 Oct. p. 83–88.
15. Don M. A low-cost software-defined telemetry receiver. Proceedings of the International Telemetry Conference, International Foundation for Telemetry; 2015.
16. Don M, Ilg M. Advances in a low-cost software-defined telemetry system. Proceedings of the International Telemetry Conference, International Foundation for Telemetry; 2017.
17. Grabner MJ, Don ML, Zajicek JM, Ilg MD, Hall R, Hallameyer JM. A networked software-defined radio telemetry receiver. DEVCOM Army Research Laboratory (US); 2020 June. Report No.: ARL-TR-8987.
18. Don M, Wessel D. Software-defined radio angle of arrival measurement of modulated signals. CCDC Army Research Laboratory (US); 2018 Dec. Report No.: ARL-TR-8601.
19. Penna F, Cabric D. Bounds and tradeoffs for cooperative DoA-only localization of primary users. Proceedings of the IEEE Global Telecommunications Conference; 2011 Dec. p. 1–5.
20. Jacobs E, Ralston EW. Ambiguity resolution in interferometry. IEEE Transactions on Aerospace and Electronic Systems. 1981 Nov;(6):766–780.
21. Guerin D. Passive direction finding. Worcester Polytechnic Institute; 2012 Oct 10.
22. Ballantine JP, Jerbert AR. Distance from a line, or plane, to a point. The American Mathematical Monthly. 1952;59(4):242–243.
23. Hartung J, Knapp G, Sinha BK. Statistical meta-analysis with applications. John Wiley & Sons; 2008.
24. Duchesne L, Garreau P, Robic N, Gandois A, Iversen PO, Barone G. Compact multi-probe antenna test station for rapid testing of antennas and wireless terminals. Proceedings of IEEE Antenna Measurements and SAR; 2004. p. 101–105.
25. Don ML, Arce GR. Antenna radiation pattern compressive sensing. Proceedings of the IEEE Military Communications Conference (MILCOM); 2018.

26. Don ML, Arce GR. Antenna pattern measurement with compressive phase retrieval. Proceedings of the IEEE Radio and Wireless Symposium (RWS); 2020.

Appendix. MATLAB Scripts

MATLAB script for Fig. 1:

```
clear %initialize MATLAB workspace
N=8; %number of agents
C=32; %CDMA code length
r=500e3; %data rate (bps)
w=16; %word width (bits)
M=N*(N-1); %total number of range measurements in swarm
b_b=6*w*M; %assumed total bits in full broadcast message
% Probably want to set n_b so that the broadcast size is not too large
% so that p_b is not too high, want p_r and p_b to be about the same
n_b=4; %number of transmissions for each broadcast
delta_b=0.005; %broadcast delay (s)
b_r=b_b/C; %b_r can be arbitrary, here made b_r*C the same as b_b
% For simplicity, the base timestep is one broadcast transmission t_t
% delta_1 and delta_2 are set so that t_1 and t_2 are multiples of a timestep
% Ranging time should be large enough so that t_r significantly affects r_s
% and isn't dominated by t_b
Rtrans_steps=2; %time steps per ranging transmission
b_t=b_b/n_b;
t_t=b_t/r+delta_b; %time of one broadcast transmission
t_b=b_b/r+n_b*delta_b; %time of full broadcast message
delta_1=Rtrans_steps*t_t-b_r/r; %delay to process outgoing ranging message
delta_2=Rtrans_steps*t_t-b_r*C/r; %delay to process incoming ranging message
if delta_1<=0 || delta_2<=0
    fprintf('Delays must be greater than zero.')
end
t_1=b_r/r+delta_1; %outgoing ranging time
t_2=b_r*C/r+delta_2; %incoming ranging time
t_r=t_1+t_2; %total ranging time

r_s=zeros(1,N-1); %preallocate total swarm update rate
K_range=1:N-1; %range of CDMA K values
for K=K_range
    n_r=ceil((N-1)/K); %number of ranging operations per agent
    t_s=N*(t_r*n_r+t_b); %total swarm update time
    r_s(K)=1/t_s; %total swarm update rate (Hz)
end
figure
plot(K_range,r_s,'LineWidth',1)
xlabel('$K$', "Interpreter", "latex")
ylabel('$r_s$ (Hz)', "Interpreter", "latex")
title('Total Swarm Update Rate')
```

MATLAB script for Fig. 2:

```
% Power calc
dmax=2500;
points=50;
d=linspace(0,dmax,points); %distance vector for calculations
k=1.381e-23; %boltzmann constant
f=6e9; %Hz
B=2*r; %bandwidth
c=3e8; %speed of light
lambda=c/f; %wavelength
T=3000; %noise temp
Pt=-17; %dBW transmit power
Pt=10.^(Pt/10); %transmit power W
FSPL=(4*pi*d/lambda).^2; %free space path loss
Pn=k*T*B; %noise power
SNR=Pt./(FSPL*Pn); %signal to noise ratio
EtoN_noise=SNR*B/r; %energy per data bit to noise density, thermal noise
figure
K_range=[1 3 5 7];
i=1;
legend_text=cell(length(K_range),1); %preallocate
for K=K_range
    EtoN_CDMA = 3*C/(2*(K-1)); %EtoN for CDMA assuming all equal power
    EtoN_total = 1./ ( 1./EtoN_noise + 1./EtoN_CDMA ); %total EtoN
    sigma_1=(1+1./(EtoN_noise*b_r))*c^2./((2*pi*B)^2.*EtoN_noise*b_r); %var1
    sigma_2=(1+1./(EtoN_total*b_r))*c^2./((2*pi*B)^2.*EtoN_total*b_r); %var2
    sigma_r=sigma_1+sigma_2; %total ranging variance
    plot(d,sigma_r,'LineWidth',1)
    hold on
    legend_text{i}=sprintf('k=%d',K);
    i=i+1;
end
legend(legend_text,'Location','Best')
xlabel('$d$ (m)','Interpreter','latex')
ylabel('$\sigma^2_r$', 'Interpreter','latex')
title('Ranging Variance')
```

MATLAB script for Fig. 3.

```
figure
Q=@(s)erfc(s/sqrt(2))/2; %Q fuction
for K=K_range
    EtoN_CDMA = 3*C/(2*(K-1)); %EtoN for CDMA assuming all equal power
    EtoN_total = 1./( 1./EtoN_noise + 1./EtoN_CDMA ); %total EtoN
    p_total = Q( ( 2*EtoN_total ).^0.5 ); %prob of error for 1 CDMA bit
    p_noise = Q( ( 2*EtoN_noise ).^0.5 ); %prob of error for 1 non-CDMA bit
    p_r = 1-((1-p_total).^(b_r)).*(1-p_noise).^(b_r)); %for range dropped
    plot(d,p_r,'LineWidth',1)
    hold on
end
p_b = 1-(1-p_noise).^(b_t); %probability broadcast transmission drop
plot(d,p_b,'LineWidth',1)
legend_text{i}='Broadcast';
legend(legend_text,'Location','Best')
xlabel('$d$ (m)','Interpreter','latex')
ylabel('P(Drop)','Interpreter','latex')
title('Probability of Dropped Messages')
```

MATLAB script for Figs. 4 and 5:

```
RFconst.C = C; %number of CDMA code bits per data bit
RFconst.w=w; %word width
RFconst.r=r; %data rate
RFconst.f=f; %freq
RFconst.Pt=10*log10(Pt); %dBW trasmit power
RFconst.T=T; %total noise temp at receiver
RFconst.perfect_gain_control=0; %assume equal power?
RFconst.b_r=b_r; %bits per broadcast transmission
RFconst.b_t=b_t; %bits per ranging message
%p_b
p_b=0.5;
p_noise=1-(1-p_b)^(1/b_t);
EtoN_noise=erfcinv(p_noise*2)^2;
SNR=EtoN_noise*r/B;
FSPL=Pt./(SNR*Pn);
d=sqrt( FSPL )*lambda/(4*pi);
set(0,'DefaultLegendAutoUpdate','off') %set legend so won't auto update
plot(d,p_b,'o') %plot p_b point on p_b plot from above
legend_text{length(legend_text)+1}=sprintf('p_b=%.2f',p_b);

%p_r
p_r_target=0.5;
```

```

dmin=0;
dmx=1e5; %initialize maximum d to a high value
K=7;
EtoN_CDMA = 3*C/(2*(K-1));
max_error=0.001;
p_error=inf;
i=1;
max_interactions = 30;
p_errorv=zeros(1,max_interactions);
p_errorv(i)=inf;
while p_errorv(i) > max_error && i<max_interactions
    d=(dmx-dmin)/2+dmin; %test half way point between dmin and dmx
    FSPL=(4*pi*d/lambda).^2; %free space path loss
    SNR=Pt./(FSPL*Pn); % SNR
    EtoN_noise=SNR*B/r; %Eb/N0, for just thermal noise
    EtoN_total = 1./( 1./EtoN_noise + 1./EtoN_CDMA ); %total EtoN
    p_noise = Q( ( 2*EtoN_noise ).^0.5 ); %for 1 non-CDMA bit
    p_total = Q( ( 2*EtoN_total ).^0.5 ); %for 1 CDMA bit
    p_r = 1-((1-p_total).^(b_r)).*(1-p_noise).^(b_r)); %for range
    if p_r > p_r_target %if probability of drop is too large, lower dmx
        dmx=d;
    else %if probability of drop is too small, raise dmim
        dmin=d;
    end
    i=i+1;
    p_errorv(i)=abs(p_r_target-p_r); %error
end
max_interactions=i-1;
plot(d,p_r_target,'d') %plot p_r point on p_r plot from above
legend_text{length(legend_text)+1}=sprintf('p_r=%.2f',p_r);
legend(legend_text, 'Location','best')

figure
semilogy(1:max_interactions,p_errorv(2:max_interactions+1),'LineWidth',1)
xlabel('Iteration','Interpreter','latex')
ylabel('$p_{error}$','Interpreter','latex')
title('p_r Error vs Iterations')
xlim([1 max_interactions])

```

MATLAB script for Figs. 6 and 7:

```

d=linspace(0,dmax,points);
FSPL=(4*pi*d/lambda).^2;
Pn=k*T*B;

```

```

SNR=Pt./(FSPL*Pn);
EtoN_noise=SNR*B/r; %Eb/N0, for just thermal noise

i=1;
figure
t_n=1;
legend_text=cell(length(K_range),1); %preallocate
sigma_norm=zeros(length(K_range),length(d));
for K=K_range

    EtoN_CDMA = 3*C/(2*(K-1)); %EtoN for CDMA assuming all equal power
    EtoN_total = 1./( 1./EtoN_noise + 1./EtoN_CDMA ); %total EtoN
    sigma_1=(1+1./(EtoN_noise*b_r))*c^2./((2*pi*B)^2.*EtoN_noise*b_r); %var1
    sigma_2=(1+1./(EtoN_total*b_r))*c^2./((2*pi*B)^2.*EtoN_total*b_r); %var2
    sigma_r=sigma_1+sigma_2;
    p_total = Q( ( 2*EtoN_total ).^0.5 ); %prob of error for 1 CDMA bit
    p_noise = Q( ( EtoN_noise ).^0.5 ); %prob of error for 1 non-CDMA bit
    p_r = 1-((1-p_total).^(b_r).*(1-p_noise).^(b_r)); %for range dropped
    p_b = 1-(1-p_noise).^(b_t); %probability broadcast transmission drop

    t_s=N*(t_r*ceil((N-1)/K)+t_b); %total swarm update time
    r_s=1/t_s; %total swarm update rate (Hz)

    sigma_norm(i,:)=sigma_r./((1-p_r).*(1-p_b).*r_s*t_n);
    semilogy(d,sigma_norm(i,:), 'LineWidth',1)
    hold on
    legend_text{i}=sprintf('k=%d',K);
    i=i+1;
end
xlim([200 2000])
ylim([10^-0.5 10^8])
title('Normalized Variance')
ylabel('$\hat{\sigma}^2_r$', "Interpreter", "latex")
xlabel('$d$ (m)', "Interpreter", "latex")

%find best K for a given d
figure
varmin=zeros(1,length(d));
varmini=zeros(1,length(d));
for i=1:length(d)
    [varmin(i),varmini(i)]=min(sigma_norm(:,i));
end
varmini2=varmini;
for i=1:length(K_range)
    varmini2(varmini==i)=K_range(i);

```

```

end
plot(d,varmini2,'LineWidth',1)
xlabel('$d$ (m)', "Interpreter", "latex")
ylabel('Best $K$', "Interpreter", "latex")
ylim([0 N])
xlim([200 2000])
title('Best K')

```

MATLAB script for Fig. 8.

```

scale_max=1000;
scale=linspace(0,scale_max,points); %distance vector for calculations
K=7;

%create cube of points
str=dec2bin(0:N-1);
a=zeros(N,3);
for i=1:N
    for j=1:3
        a(i,j)=str2double(str(i,j));
    end
end
a=a*2-repmat([1 1 1],N,1); %center at zero

%grow from 0 to dmax
A=repmat(a,1,1,points);
for i=1:points
    A(:, :, i)=A(:, :, i)*scale(i);
end

%find adj matrix
adj=zeros(N,N,points);
for i=1:points
    %adj(:, :, i) = squareform(pdist(A(:, :, i), 'euclidean')); %license problem
    for r=1:N
        for c=1:N
            adj(r,c,i)=sqrt(sum((A(r, :, i)-A(c, :, i)).^2));
        end
    end
end

%assume agent 1 is ranging to K other agents
%find p_r from agent 1 to agent K+1
%change RFconst into vector

```

```

fields = fieldnames(RFconst);
RFconstv= zeros(length(fields),1);
for i=1:length(fields)
    RFconstv(i)=RFconst.(fields{i});
end
p_1m=zeros(N,N,points); %preallocate p_r matrix
p_2m=zeros(N,N,points); %preallocate p_r matrix
sched1=zeros(N,N); %incoming range schedule
sched1(1,2:K+1)=1; %row = from, col = to; from 1 to others
sched2=zeros(N,N); %outgoing range schedule
sched2(2:K+1,1)=1;
sigma_1m=zeros(N,N,points);
sigma_2m=zeros(N,N,points);

% equal power levels
RFconstv(7)=1;
for i=1:points
    [~,~,p_1m(:,:,i),sigma_1m(:,:,i),~] = calc_ranges_function(adj(:,:,i),
sched1, RFconstv);
    [~,~,p_2m(:,:,i),sigma_2m(:,:,i),~] = calc_ranges_function(adj(:,:,i),
sched2, RFconstv);
end
figure
subplot(2,1,1)
p_r_equal=1-(1-squeeze(p_2m(K+1,1,:))).*(1-squeeze(p_1m(1,K+1,:)));
d=squeeze(adj(1,K+1,:));
plot(d,p_r_equal,'LineWidth',1)
hold on

subplot(2,1,2)
sigma_r_equal=squeeze(sigma_2m(K+1,1,:))+squeeze(sigma_1m(1,K+1,:));
plot(d,sigma_r_equal,'LineWidth',1)
hold on

% unequal power levels
RFconstv(7)=0;
for i=1:points
    if i==25
        ggg=3;
    end
    [~,~,p_1m(:,:,i),sigma_1m(:,:,i),~] = calc_ranges_function(adj(:,:,i),
sched1, RFconstv);
    [~,~,p_2m(:,:,i),sigma_2m(:,:,i),~] = calc_ranges_function(adj(:,:,i),
sched2, RFconstv);
end

```

```

subplot(2,1,1)
p_r_unequal=1-(1-squeeze(p_2m(K+1,1,:)).*(1-squeeze(p_1m(1,K+1,:))));
d=squeeze(adj(1,K+1,:));
plot(d,p_r_unequal,'LineWidth',1)
title('Comparison of Equal and Unequal Received Power')
xlabel('$d$ (m)','Interpreter','latex')
ylabel('$p_r$', 'Interpreter','latex')
legend('Equal','Unequal','Location','southeast')

subplot(2,1,2)
sigma_r_unequal=squeeze(sigma_2m(K+1,1,:))+squeeze(sigma_1m(1,K+1,:));
plot(d,sigma_r_unequal,'LineWidth',1)
title('Comparison of Equal and Unequal Received Power')
xlabel('$d$ (m)','Interpreter','latex')
ylabel('$\sigma^2_r$', 'Interpreter','latex')
legend('Equal','Unequal','Location','southeast')

function [wAdj,drops,Pdrop_out,var_out,dMat] = calc_ranges_function(dMat,
sched, RFconstv)

% dMat - real distances
% sched - schedule
% wAdj - successful range measurements, if other way successful last time step

persistent var_last cnt %var of sucessful tx last time step (one-way)

%from is row, col is to
%example:
% sched=[ 0 0 1 1; %this is 1 -> 3 and 1 -> 4
%          0 0 1 0; %this is 2 -> 3
%          0 0 0 0;
%          0 0 0 0];
%
% var_last=[ 0 0 0 0; %this is 3 -> 1 and 4 -> 1
%            0 0 0 0; %this is 3 -> 2
%            2 1 0 0;
%            3 0 0 0];
% var_last(var_last==0)=nan;
%
% dMat= [ 0    1.01  1    1.02;
%         1.01  0    1.04  1.03;
%         1    1.05  0    1.05;
%         1.02  1.03  1.04  0    ]*23000;

[n,~]=size(sched);

```

```

%init
if isempty(var_last)
    var_last=nan(n,n);
end

var_last2=nan(n,n); %tmp for var_last
var_now=nan(n,n);
wAdj=nan(n,n); %range measurements
                %if not dropped, and not last time going other way,
                %then fill in with range measurement, var = var1 + var2

tx = double(sum(sched,2)>0); %list of transmitters
tx(tx==0)=nan;
rx = sum(sched,1)'>0; %list of recievers
drops=nan(n,n);
var_out=nan(n,n);
Pdrop_out=nan(n,n);
%for each rx, get ProbDrop and Rvar with respect to each tx
for i=1:n
    if rx(i) %if doing any rx
        d=tx.*dMat(:,i); %dist from rx to all tx's
        %ProbDrop, Rvar with respect to each tx to this rx
        [~,ProbDrop,Rvar]=cdma_function(d,RFconstv);
        RxSuccess=(rand(n,1)>ProbDrop).*sched(:,i); %RxSuccess if doing rx
        drops(:,i)=(RxSuccess==0) & (sched(:,i)>0); %debug output
        var_out(:,i)=Rvar; %debug output
        Pdrop_out(:,i)=ProbDrop; %debug output
        RxSuccess(RxSuccess==0)=nan;
        var_now(:,i)=RxSuccess.*Rvar; %current one-way vars, from each tx
        %if previous range message was successful,
        % add the calculated variances together to get full range variance
        %if previous range was unsuccessful
        % then nan in var_last will cause this to be nan too
        wAdj(:,i)=((var_now(:,i)+var_last(i,:)).^0.5).*randn(n,1)+dMat(:,i);
%ranges

        %keep track of successful one-way ranging
        % for any 2-way ranging in next step
        %successful one-way ranging for any 2-way ranging in next step:
        var_last2(:,i)=var_now(:,i);
    else
        var_last2(:,i)=nan(n,1); %if didn't do a rx, then not successful
    end
end
end

```

```

var_last=var_last2; %update var_last
end

```

MATLAB script for Figs. 10–14:

```

% simulate two angle measurement sweeps at two freq, and resolve ambiguity
% simple sim - no sdr data

clear
close all

set(0,'defaultAxesFontName','Times')
set(0,'defaultTextFontName','Times')
set(0,'defaultTextFontSize',12)
set(0,'defaultAxesFontSize',12)

addpath './geom toolbox\geom2d-2017.01.04\geom2d\geom2d'
addpath './geom toolbox\geom2d-2017.01.04\geom2d\polygons2d'
addpath './geom toolbox\geom2d-2017.01.04\geom2d\polynomialCurves2d'
addpath './geom toolbox\geom2d-2017.01.04\geom2d\utils'
rng(0);
findBestF2=1; %find the beast second frequency

fhighv=[2.4e9 1.92e9]; %freq of incoming signal
lambdav=3e8./fhighv; %wavelength of incoming signal
d1=2*(3e8./2.2555e9);
dv=[d1 d1]; %distance between antennas
nmeasurements=2;
thetamax=50;
n=floor((dv*sin(thetamax*pi/180)+(lambdav/2))./lambdav);
noise_var=0.03; %0.03 works, 0.04 doesn't
%noise_var=0.0;
ntheta=100;
thetav=linspace(-thetamax*pi/180,thetamax*pi/180,ntheta);
rthetav=zeros(nmeasurements,ntheta);
phase_diffv=zeros(nmeasurements,ntheta);
nsamples=1;
fs=20*max(fhighv);
t=0:1/fs:(nsamples-1)*1/fs;

figure(1)
hold on
for measurementi=1:nmeasurements
    for i=1:ntheta
        fhigh=fhighv(measurementi);
        d=dv(measurementi);
    end
end

```

```

        lambda=3e8./fhigh;
        phase_diff=2*pi*d*sin(thetav(i))/lambda; %initial phase of signal at
antenna 2
        noise1=noise_var^0.5*randn(1,length(t));
        noise2=noise_var^0.5*randn(1,length(t));
        %plot(phase_diff)
        %if d <= lambda/2 and -pi<AoA< pi, then the max phase diff with be +-
pi
        %if not this, it's 2*pi off, need to fix
        %problem, if real angle is close to pi/2, and alot of noise, can get a
        %false case of 2*pi off, and mess things up!!!
        p1=phase_diff+noise1;
        phase_diff=(sign(p1)*pi+p1)-floor((sign(p1)*pi+p1)/(2*pi))*2*pi-pi;
%wrap phase
        phase_diffv(measurementi,i)=mean(phase_diff);
    end
    plot(thetav*180/pi,phase_diffv(measurementi,:))
end

xlabel('\theta (Degrees)')
ylabel('\Delta\phi (Radians)')
legend('\Delta\phi_1', '\Delta\phi_2', 'Location', 'BestOutside')
xlim([-thetamax thetamax])
box on

figure(44)
for measurementi=1:1
    for i=1:ntheta
        fhigh=fhighv(measurementi);
        d=dv(measurementi);
        lambda=3e8./fhigh;
        phase_diff=2*pi*d*sin(thetav(i))/lambda; %initial phase of signal at
antenna 2
        noise1=noise_var^0.5*randn(1,length(t));
        noise2=noise_var^0.5*randn(1,length(t));
        %plot(phase_diff)
        %if d <= lambda/2 and -pi<AoA< pi, then the max phase diff with be +-
pi
        %if not this, it's 2*pi off, need to fix
        %problem, if real angle is close to pi/2, and alot of noise, can get a
        %false case of 2*pi off, and mess things up!!!
        p1=phase_diff+noise1;
        phase_diff=(sign(p1)*pi+p1)-floor((sign(p1)*pi+p1)/(2*pi))*2*pi-pi;
%wrap phase
        phase_diffv(measurementi,i)=mean(phase_diff);
    end
end

```

```

end
plot(thetav*180/pi,phase_diffv(measurementi,:))
end

xlabel('\theta (Degrees)')
ylabel('\Delta\phi (Radians)')
xlim([-thetamax thetamax])
box on

%% resolve amb
index1=1;
index2=2;

r=dv(index2)*lambdav(index1)/(dv(index1)*lambdav(index2)); %ratio of distance
between 13 and 12 (or between wavelengths)
phi12=-100:100:100;
dista=[];
figure(2)
ta='legend(';
linem=[];
im=[];
for i12 = -1*n(index1):n(index1)
    for i13= -1*n(index2):n(index2)
        phi13=r*phi12-r*i12*2*pi+i13*2*pi;
        dist=abs(phi13-r*phi12+i13-r*i13);

        %test if line is in 1x1 box
        poly = [-pi -pi;pi -pi;pi pi;-pi pi];
        line = [phi12(1) phi13(1) 1 r];
        p=intersectLinePolygon(line, poly);

        if ~isempty(p)
            plot(phi12,phi13)
            hold on
            ta=[ta sprintf(''I_{1} = %d, I_{2} = %d'',',i12,i13)];
            linem=[linem; line];
            im=[im; [i12 i13]];
        end
    end
end
end
axis([-pi pi -pi pi])
plot(phase_diffv(index1,:),phase_diffv(index2,:), 'b.')
ta=ta(1:end-1);
ta=[ta '','Phase Points','','Location','','BestOutside'];'];
eval(ta)

```

```

xlabel('\Delta\phi_{1} (Radians)')
ylabel('\Delta\phi_{2} (Radians)')

%find line closest to each point
nlines=length(linem(:,1));
pim=zeros(length(phase_diffv(1,:)),2);
for i=1:length(phase_diffv(1,:))
    distmin=inf;
    for k=1:nlines
        dist=distancePointLine([phase_diffv(index1,i) phase_diffv(index2,i)],
linem(k,:));
        if dist<distmin
            pim(i,:)=im(k,:);
            distmin=dist;
        end
    end
end

%% correct phase roll over
figure(3)
subplot(2,1,1)
hold off
plot(thetav*180/pi,phase_diffv(index1,:))
hold on
plot(thetav*180/pi,pim(:,1)*pi);
phase_diff_correctv(index1,:)=phase_diffv(index1,:)-pim(:,1)*pi*2;
plot(thetav*180/pi,phase_diff_correctv(index1,:), '--');
ylabel('\phi (Radians)')
xlabel('\theta (Degrees)')
legend('\Delta\phi_1', '-2\pi I_1', 'Unwrapped
\Delta\phi_1', 'Location', 'BestOutside');
xlim([-thetamax thetamax])
subplot(2,1,2)
hold off
plot(thetav*180/pi,phase_diffv(index2,:))
hold on
plot(thetav*180/pi,pim(:,2)*pi);
phase_diff_correctv(index2,:)=phase_diffv(index2,:)-pim(:,2)*pi*2;
plot(thetav*180/pi,phase_diff_correctv(index2,:), '--');
ylabel('\phi (Radians)')
xlabel('\theta (Degrees)')
legend('\Delta\phi_2', '-2\pi I_2', 'Unwrapped
\Delta\phi_2', 'Location', 'BestOutside');
xlim([-thetamax thetamax])

```

```

%% recover theta
figure(4)
subplot(2,1,1)
rangle=asin(phase_diff_correctv(:,:).*repmat(lambdav',1,ntheta)./(2*pi.*repmat
(dv',1,ntheta)))*180/pi;
plot(thetav*180/pi,rangle(index1,:))
hold on
plot(thetav*180/pi,rangle(index2,:))
plot(thetav*180/pi,thetav*180/pi,'--')
xlabel('True \theta (Degrees)')
ylabel('\theta (Degrees)')
legend('Recovered \theta from \lambda_1','Recovered \theta from
\lambda_2','True \theta','Location','Best')
xlim([-thetamax thetamax])
subplot(2,1,2)
error=[thetav*180/pi;thetav*180/pi]-rangle;
plot(thetav*180/pi,(error(index1,:)))
hold on
plot(thetav*180/pi,(error(index2,:)))
xlabel('True \theta (Degrees)')
ylabel('\theta Error (Degrees)')
legend('\theta error from \lambda_1','\theta error from
\lambda_2','Location','North')

%% find the best second frequency
if findBestF2
    s12=2.4e9;
    lambda13a=10.^(-1.3:0.001:0.2); %0.275 at 1.6 GHz
    s13a=3e8./lambda13a;
    dminv=zeros(1,length(s13a));
    nv=zeros(length(s13a),2);
    for i=1:length(s13a)
        s13=s13a(i);
        lambdav=3e8./[s12 s13];
        r=s12/s13;
        mind=inf;
        n=floor((dv*sin(thetamax*pi/180)+(lambdav/2))./lambdav);
        for i12 = -n(1):n(1)
            for i13= -n(2):n(2)
                for i122=-n(1):n(1)
                    for i132=-n(2):n(2)
                        if ~((i12==i122) && (i13==i132))
                            b1=r*i12-i13;
                            b2=r*i122-i132;
                            m=r;

```

```

        d=abs((b1-b2)/(1+m^2)^0.5);
        if d<mind
            mind=d;
        end
    end
end
end
end
end
dminv(i)=mind;
nv(i,:)=n;
end

figure(7)
subplot(2,1,1)
semilogx(lambda13a,dminv,'b.')
ylabel('Min Phase Spacing')
xlabel('\lambda_2 (m)')
xlim([min(lambda13a) max(lambda13a)]);
x1=3e8/2.4e9;
y1=dminv(find(lambda13a>x1,1));
hold on
plot(x1,y1,'ro')
x1=3e8/1.92e9;
y1=dminv(find(lambda13a>x1,1));
plot(x1,y1,'go')
x1=0.1875;
y1=dminv(find(lambda13a>x1,1));
plot(x1,y1,'mo')
legend('Minimum Phase Line Spacing','\lambda_2 = 12.5 cm','\lambda_2 = 15.625 cm','\lambda_2 = 18.75 cm','Location','Best')

subplot(2,1,2)
semilogx(lambda13a,nv(:,1)');
hold on
semilogx(lambda13a,nv(:,2)');
xlim([min(lambda13a) max(lambda13a)]);
xlabel('\lambda_2 (m)')
ylabel('I_{max}')
legend('I_1 Max','I_2 Max')
end

```

MATLAB script for Figs. 15-19 (uses the geom2d toolbox from MathWork's File Exchange)

```
% simulate two angle measurement sweeps at two freq, and resolve ambiguity
```

```

% simple sim - no sdr simulation
addpath './geom toolbox\geom2d-2017.01.04\geom2d\geom2d'
addpath './geom toolbox\geom2d-2017.01.04\geom2d\polygons2d'
addpath './geom toolbox\geom2d-2017.01.04\geom2d\polynomialCurves2d'
addpath './geom toolbox\geom2d-2017.01.04\geom2d\utils'

nsamples=1; %number of samples for phase measurement at one angle
nloops=500;
MSEv=zeros(1,nloops);
fhighv=[2.4e9 1.92e9 1.351e9];
lambdav=3e8./fhighv;
Nf=length(fhighv);
ntheta=100;
errorrm=zeros(Nf,ntheta,nloops);
ave_errorrm=zeros(ntheta,nloops);
Wave_errorrm=zeros(ntheta,nloops);
dphase_diff=zeros(Nf,ntheta);
noise_var=0.03;
d1=2*(3e8./2.255e9);
dv=ones(1,Nf)*d1; %distance between antennas
thetamax=50;
n=floor((dv*sin(thetamax*pi/180)+(lambdav/2))./lambdav); %maximum I for each
freq.
thetav=linspace(-thetamax*pi/180,thetamax*pi/180,ntheta);
rng(2);

findBestF2=0;
findBestF3=0;
findBestF4=0;
do_plot=0; %turn off basic plotting to use nloops>1
do_plot3=0; %turn off 3D plotting to use nloops>1
do_plots_first_loop=1;

for loopi=1:nloops
    if loopi==1
        do_plot=1; %turn off basic plotting
        do_plot3=1; %turn off 3D plottin
    else
        do_plot=0; %turn off basic plotting
        do_plot3=0; %turn off 3D plottin
    end

    if mod(loopi,20)==0
        fprintf('%d ',loopi)
    end
end

```

```

rthetav=zeros(Nf,ntheta);
phase_diffv=zeros(Nf,ntheta);

fs=20*max(fhighv); %sampling frequency
t=0:1/fs:(nsamples-1)*1/fs; %time vector

if do_plot
    figure(1)
    hold on
end
for measurementi=1:Nf
    for i=1:ntheta
        fhigh=fhighv(measurementi);
        d=dv(measurementi);
        lambda=3e8./fhigh;
        phase_diff=2*pi*d*sin(thetav(i))/lambda; %initial phase of signal
at antenna 2
        noise1=noise_var^0.5*randn(1,length(t));
        noise2=noise_var^0.5*randn(1,length(t));
        %if d <= lambda/2 and -pi<AoA< pi, then the max phase diff with be
+-pi
        %if not this, it's 2*pi off, need to fix
        %problem, if real angle is close to pi/2, and alot of noise, can
get a
        %false case of 2*pi off, and mess things up!!!
        p1=phase_diff+noise1;
        phase_diff=(sign(p1)*pi+p1)-floor((sign(p1)*pi+p1)/(2*pi))*2*pi-
pi; %wrap phase
        phase_diffv(measurementi,i)=mean(phase_diff);
    end
    if do_plot
        plot(thetav*180/pi,phase_diffv(measurementi,:))
    end
end

if do_plot
    xlabel('\theta (Degrees)')
    ylabel('\phi (Radians)')
    for i=1:Nf
        ltext{i}=sprintf('\Delta\phi_%d',i);
    end

    legend(ltext,'Location','BestOutside')
    xlim([-thetamax thetamax])

```

```

        box on
    end

    %want to make all lines
    %need every permutation of i1 i2 i3
    %each col of I is for that freq.
    Ni=prod(n*2+1);
    I=zeros(Ni,Nf);
    for i=1:Nf
        if i<Nf
            rep1=prod(n(i+1:end)*2+1); %number of times to repeat each number
        else
            rep1=1;
        end
        if i>1
            rep2=prod(n(1:i-1)*2+1); %number of times to repeat each series
        else
            rep2=1;
        end
        tmp=-n(i):n(i);
        tmp= repmat(tmp,rep1,rep2);
        I(:,i)=tmp(:);
    end

    %plot each line, need two points
    %specify line - need one point and unit pointing vector
    if do_plot3
        figure(2)
    end
    IinBox=zeros(Ni,1);
    for i=1:Ni
        theta=repmat(linspace(-pi,pi,ntheta),Nf,1);

        Pdelta_phi=2*pi*(d*sin(theta)./repmat(lambdav',1,ntheta)+repmat(I(i,:)',1,ntheta));

        if do_plot3
            plot3(Pdelta_phi(1,:),Pdelta_phi(2,:),Pdelta_phi(3,:));
            hold on
        end
        tmp=Pdelta_phi(1,:) < pi & Pdelta_phi(1,:) > -pi;
        tmp=tmp & Pdelta_phi(2,:) < pi & Pdelta_phi(2,:) > -pi;
        if Nf>2
            tmp=tmp & Pdelta_phi(3,:) < pi & Pdelta_phi(3,:) > -pi;
        end
        if sum(tmp)

```

```

        IinBox(i)=1;
    end
end
if do_plot3
    plot3(phase_diffv(1,:),phase_diffv(2,:),phase_diffv(3,:), 'b. ');
    axis([-pi pi -pi pi -pi pi])
    grid on
end

Ni=sum(IinBox);
I=I(IinBox==1,:);

if do_plot3
    figure(3);
end
for i=1:Ni
    theta= repmat(linspace(-pi,pi,ntheta),Nf,1);

    Pdelta_phi=2*pi*(d*sin(theta)./repmat(lambdav',1,ntheta)+repmat(I(i,:) ',1,ntheta));

    if do_plot3
        if i>7
            plot3(Pdelta_phi(1,:),Pdelta_phi(2,:),Pdelta_phi(3,:), '--', 'LineWidth', 2);
        else
            plot3(Pdelta_phi(1,:),Pdelta_phi(2,:),Pdelta_phi(3,:), 'LineWidth', 2);
        end
        hold on
        ltext{i}=sprintf('I_1=%2d, I_2=%2d, I_3=%2d', I(i,1), I(i,2), I(i,3));
    end
end
if do_plot3
    plot3(phase_diffv(1,:),phase_diffv(2,:),phase_diffv(3,:), 'b. ');
    axis([-pi pi -pi pi -pi pi])
    grid on
    legend(ltext)
end
%for each point, see which line its closest to
%doesn't work if right on line
pim=zeros(ntheta,Nf);
for k=1:ntheta
    dmin=inf;
    for i=1:Ni

```

```

        theta=0;
        Pdelta_phi1=2*pi*(d*sin(theta)./lambdav'+I(i,:)); %point 1 of
line
        theta=pi/2;
        Pdelta_phi2=2*pi*(d*sin(theta)./lambdav'+I(i,:)); %point 2 of
line
        Vunit=(Pdelta_phi2-Pdelta_phi1);
        Vunit=Vunit/norm(Vunit);
        %plot3(phase_diffv(1,k),phase_diffv(2,k),phase_diffv(3,k),'r*')
        d=norm((Pdelta_phi1-phase_diffv(:,k))-(dot((Pdelta_phi1-
phase_diffv(:,k)),Vunit)*Vunit));
        if d<dmin
            dmin=d;
            pim(k,:)=I(i,:);
        end
    end
end

% correct phase roll over
phase_diff_correctv=phase_diffv-pim'*pi*2;

if do_plot
    figure(4)
    for i=1:Nf
        subplot(Nf,1,i)
        hold off
        plot(thetav*180/pi,phase_diffv(i,:))
        hold on
        plot(thetav*180/pi,pim(:,i)'+pi);
        plot(thetav*180/pi,phase_diff_correctv(i,:), '--');
        ylabel('\phi (Radians)')
        xlabel('\theta (Degrees)')
        clear ltext
        ltext{1}=sprintf('\Delta\phi_%d',i);
        ltext{2}=sprintf('-2\pi I_%d',i);
        ltext{3}=sprintf('Unwrapped \Delta\phi_%d',i);
        legend(ltext, 'Location', 'BestOutside');
        xlim([-thetamax thetamax])
    end

end

range=asin(phase_diff_correctv(:,:).*repmat(lambdav',1,ntheta)./(2*pi.*repmat(
dv',1,ntheta)))*180/pi;
error=repmat(thetav*180/pi,Nf,1)-range;

```

```

MSE=sum(error'.^2)/ntheta;

rangle_ave=sum(rangle)/Nf;
error_ave=thetav*180/pi-rangle_ave;
MSE_ave=sum(error_ave.^2)/ntheta;
MSEv(loopi)=MSE_ave;
errorrm(:, :, loopi)=error;

ave_errorm(:, loopi)=thetav*180/pi-sum(rangle)/Nf;

for i=1:Nf
    %dphase_diff(i,:)=2*pi*d*cos(thetav)/lambdav(i);

    dphase_diff(i,:)=1/lambdav(i);
end
weights=dphase_diff.^2;
weights=weights./repmat(sum(weights),Nf,1);

Wave_errorm(:, loopi)=thetav*180/pi-sum(rangle.*weights);

if do_plot
    figure(5)
    subplot(2,1,1)
    clear ltext
    for i=1:Nf
        plot(thetav*180/pi, rangle(i,:))
        hold on
        ltext{i}=sprintf('Recovered \\theta from \\lambda_%d', i);
    end
    plot(thetav*180/pi, thetav*180/pi, '--')
    xlabel('True \\theta (Degrees)')
    ylabel('\\theta (Degrees)')
    ltext{i+1}='True \\theta';
    legend(ltext, 'Location', 'northeastoutside')
    xlim([-thetamax thetamax])

    subplot(2,1,2)
    clear ltext
    for i=1:Nf
        plot(thetav*180/pi, (error(i,:)))
        hold on
        ltext{i}=sprintf('\\theta error from \\lambda_%d', i);
    end
    plot(thetav*180/pi, (error_ave))
    ltext{i+1}='Ave \\theta error';

```



```

        dminv(i)=mind;
        nv(i,:)=n;
    end
    figure(77)
    subplot(2,1,1)
    semilogx(3e8./lambda13a,dminv,'b.')
    ylabel('Min Phase Spacing')
    xlabel('\lambda_2 (m)')
    hold on

    figure(7)
    subplot(2,1,1)
    semilogx(lambda13a,dminv,'b.')
    ylabel('Min Phase Spacing')
    xlabel('\lambda_2 (m)')
    xlim([min(lambda13a) max(lambda13a)]);
    x1=3e8/2.4e9;
    y1=dminv(find(lambda13a>x1,1));
    hold on
    plot(x1,y1,'ro')
    x1=3e8/1.92e9;
    y1=dminv(find(lambda13a>x1,1));
    plot(x1,y1,'go')
    x1=3e8/2.6e9;
    y1=dminv(find(lambda13a>x1,1));
    plot(x1,y1,'mo')
    legend('Minimum Phase Line Spacing','\lambda_2 = 12.5 cm','\lambda_2 = 15.625 cm','\lambda_2 = 11.538 cm','Location','Best')

    subplot(2,1,2)
    semilogx(lambda13a,nv(:,1));
    hold on
    semilogx(lambda13a,nv(:,2));
    xlim([min(lambda13a) max(lambda13a)]);
    xlabel('\lambda_2 (m)')
    ylabel('I_{max}')
    legend('I_1 Max','I_2 Max')
end

if findBestF3
    %Now go nD
    %if really supporting nD, have some system where start with 2, then
    use results for 3, then 4 ....
    %for now just have in mind 3D
    s14=2.4e9;

```

```

s12=1.92e+09;
lambda13a=10.^(-1.3:0.0025:0.2);
s13a=3e8./lambda13a;
dminv=zeros(1,length(s13a));
nv=zeros(length(s13a),3);
dv=ones(3,1)*dv(1);
fprintf('\nfind3 %d\n',length(lambda13a))
for fi=1:length(s13a)
    if mod(fi,10)==0
        fprintf('%d ',fi)
    end
    s13=s13a(fi);
    lambdav=3e8./[s12; s13; s14];
    mind=inf;
    n=floor((dv*sin(thetamax*pi/180)+(lambdav/2))./lambdav);
    Ni=prod(n*2+1);
    I=zeros(Ni,Nf);
    for i=1:Nf
        if i<Nf
            rep1=prod(n(i+1:end)*2+1); %number of times to repeat each
number
        else
            rep1=1;
        end
        if i>1
            rep2=prod(n(1:i-1)*2+1); %number of times to repeat each
series
        else
            rep2=1;
        end
        tmp=-n(i):n(i);
        tmp= repmat(tmp,rep1,rep2);
        I(:,i)=tmp(:);
    end

    for i=1:Ni
        %i
        for j=1:Ni
            if i~=j
                theta=repmat(linspace(-pi,pi,ntheta),Nf,1);
                p1=2*pi*(d*1./lambdav+I(i,:)');
                p2=2*pi*(d*-1./lambdav+I(i,:)');
                q=2*pi*(d*1./lambdav+I(j,:)');
                d=norm(q-p1-dot((q-p1),(p2-p1))*(p2-p1)/norm(p2-
p1)^2);

```

```

        if d<mind
            mind=d;
        end
    end
end
end
dminv(fi)=mind;
nv(fi,:)=n;
end

figure(8)
subplot(2,1,1)
semilogx(lambda13a,dminv,'b.')
ylabel('Min Phase Spacing')
xlabel('\lambda_3 (m)')
xlim([min(lambda13a) max(lambda13a)]);
x1=0.1372;
y1=dminv(find(lambda13a>x1,1));
hold on
plot(x1,y1,'ro')
x1=3e8/1.6e9;
x1=0.22206;
y1=dminv(find(lambda13a>x1,1));
plot(x1,y1,'go')
x1=3e8/2.6e9;
x1=0.096605;
y1=dminv(find(lambda13a>x1,1));
plot(x1,y1,'mo')
legend('Min. Spacing','\lambda_3 = 13.72 cm','\lambda_3 = 22.206
cm','\lambda_3 = 9.66 cm','Location','Best')

subplot(2,1,2)
semilogx(lambda13a,nv(:,3));
hold on
semilogx(lambda13a,nv(:,1));
semilogx(lambda13a,nv(:,2));
xlim([min(lambda13a) max(lambda13a)]);
xlabel('\lambda_3 (m)')
ylabel('I_{max}')
legend('I_1 Max','I_2 Max','I_3 Max')
end

if findBestF4
    %Now go nD

```

```

        %if really supporting nD, have some system where start with 2, then
use results for 3, then 4 ....
        %for now just have in mind 3D
        s14=1.6e9;
        s12=1.0669e+09;
        s15=6.6667e+08;
        lambda13a=10.^(-1:0.01:0.08);
        lambda13a=0.23:0.001:0.25;
        s13a=3e8./lambda13a;
        dminv=zeros(1,length(s13a));
        nv=zeros(length(s13a),Nf);
        dv=ones(Nf,1)*dv(1);
        fprintf('\nfind4 %d\n',length(lambda13a))
        for fi=1:length(s13a)
            if mod(fi,10)==0
                fprintf('%d ',fi)
            end
            s13=s13a(fi);
            lambdav=3e8./[s12; s13; s14; s15];
            mind=inf;
            n=floor((dv*sin(thetamax*pi/180)+(lambdav/2))./lambdav);

            Ni=prod(n*2+1);
            I=zeros(Ni,Nf);
            for i=1:Nf
                %first do
                if i<Nf
                    rep1=prod(n(i+1:end)*2+1); %number of times to repeat each
number
                else
                    rep1=1;
                end
                if i>1
                    rep2=prod(n(1:i-1)*2+1); %number of times to repeat each
series
                else
                    rep2=1;
                end
                tmp=-n(i):n(i);
                tmp=repmat(tmp,rep1,rep2);
                I(:,i)=tmp(:);
            end

            for i=1:Ni
                for j=1:Ni

```

```

        if i~=j
            theta= repmat(linspace(-pi,pi,ntheta),Nf,1);
            p1=2*pi*(d*1./lambdav+I(i,:))';
            p2=2*pi*(d*-1./lambdav+I(i,:))';
            q=2*pi*(d*1./lambdav+I(j,:))';
            d=norm(q-p1-dot((q-p1),(p2-p1))*(p2-p1)/norm(p2-
p1)^2);

            if d<mind
                mind=d;
            end
        end
    end
    end
    dminv(fi)=mind;
    nv(fi,:)=n;
end

figure(9)
subplot(2,1,1)
semilogx(lambda13a,dminv,'b.')
ylabel('Min Phase Spacing')
xlabel('\lambda_2 (m)')
xlim([min(lambda13a) max(lambda13a)]);
x1=3e8/2.4e9;
y1=dminv(find(lambda13a>x1,1));
hold on
plot(x1,y1,'ro')
x1=3e8/1.6e9;
y1=dminv(find(lambda13a>x1,1));
plot(x1,y1,'go')
x1=3e8/2.6e9;
y1=dminv(find(lambda13a>x1,1));
plot(x1,y1,'mo')
legend('Minimum Phase Line Spacing','\lambda_2 = 12.5 cm','\lambda_2 =
18.75 cm','\lambda_2 = 11.538 cm','Location','Best')

subplot(2,1,2)
semilogx(lambda13a,nv(:,1));
hold on
semilogx(lambda13a,nv(:,2));
xlim([min(lambda13a) max(lambda13a)]);
xlabel('\lambda_2 (m)')
ylabel('I_{max}')
legend('I_1 Max','I_2 Max')

```

```

end
end

```

MATLAB script for Fig .20

```

r=500e3;
dmax=2500;
points=100;
fhighv=[2.4e9 1.92e9 1.351e9];
lambdav=3e8./fhighv;
var_est=zeros(1,points);
d=linspace(0,dmax,points); %distance vector for calculations
k=1.381e-23; %boltzmann constant
f=6e9; %Hz
B=2*r; %bandwidth
c=3e8; %speed of light
figure
for i=1:3
    lambda=lambdav(i); %wavelength
    T=3000; %noise temp
    Pt=-17; %dBW trasmit power
    Pt=10.^(Pt/10); %transmit power W
    FSPL=(4*pi*d/lambda).^2; %free space path loss
    Pn=k*T*B; %noise power
    SNR=Pt./(FSPL*Pn); %signal to noise ratio

    subplot(2,1,1)
    plot(d,10*log10(SNR),'LineWidth',1) %power is 10 log10
    hold on
    ylabel('SNR (dB)')
    xlabel('$d$ (m)', "Interpreter", "latex")
end

subplot(2,1,2)
Nsamples=1;
thetav=30*pi/180;
d1=2*(3e8./2.2555e9);
test=(4*Pn*d.^2)/(Pt*Nsamples*(d1*cos(thetav))^2); %the lambdas cancel
plot(d,test*(180/pi)^2, 'LineWidth',1)
hold on
plot(d,test/3*(180/pi)^2, 'LineWidth',1)
ylabel('$\sigma^2 \backslash, \backslash, \backslash, (\mathrm{m})^2$', "Interpreter", "latex")
xlabel('$d$ (m)', "Interpreter", "latex")
legend('$\sigma^2$', '$\sigma^2_{Ave}$', 'Location', 'Best', "Interpreter", "latex"
)

```

```

subplot(2,1,1)
legend('\lambda_1','\lambda_2','\lambda_3','Location','Best')
ylim([0 45])

```

MATLAB script for Fig. 21

```

points=100;
r=500e3;
k=1.381e-23; %boltzmann constant
f=6e9; %Hz
B=2*r; %bandwidth
c=3e8; %speed of light
T=3000; %noise temp
Pn=k*T*B; %noise power
theta=45*pi/180;
d1=2*(c/2.2555e9);
fhighv=[6e9 3e9 1e9];
lambdav=3e8./fhighv;
Pt=-17; %dBW trasmit power
Pt=10.^(Pt/10); %transmit power W
d=750;
N=1;
nruns=10000;
thetav=linspace(-50,50,points)*pi/180;
errorv=zeros(3,points);
erroravev=zeros(1,points);
errorwv=zeros(1,points);
w=lambdav/sum(lambdav);
for i=1:points
    varv=8*pi*Pn*d^3./(lambdav*Pt*N*(d1*cos(thetav(i)))^2);
    errorave=0;
    errorw=0;
    error=zeros(1,3);
    for runi=1:nruns
        error=error+abs(sqrt(varv).*randn(1,3));
        errorave=errorave+abs(sum(sqrt(varv).*randn(1,3)))/3;
        errorw=errorw+abs(sum(sqrt(varv).*randn(1,3).*w));
    end
    errorv(:,i)=error/nruns;
    erroravev(i)=errorave/nruns;
    errorwv(i)=errorw/nruns;
end
thetav=thetav*180/pi;
figure
for i=1:3

```

```

        ltext{i}=sprintf('\lambda_%d = %d cm',i,lambdav(i)*100);
    end
    ltext{4}='Ave';
    ltext{5}='Weighted Ave';
    plot(thetav,errorv(1,:), 'LineWidth',1)
    hold on
    plot(thetav,errorv(2,:), 'LineWidth',1)
    plot(thetav,errorv(3,:), 'LineWidth',1)
    plot(thetav,erroravev, 'LineWidth',1)
    plot(thetav,errorwv, 'LineWidth',1)

    xlabel('\theta (Degrees)')
    ylabel('\theta RMSE (Degrees)')
    legend(ltext, 'Location', 'best')

```

List of Symbols, Abbreviations, and Acronyms

2-D	two-dimensional
bps	bits per second
CDMA	code division multiple access
DoA	direction of arrival
GPS	Global Positioning System
Kbps	kilobytes per second
PL	path loss
RF	radio frequency
RMSE	root mean squared error
SDR	software-defined radio
SNR	signal to noise ratio
TDMA	time division multiple access
TWR	two-way ranging

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 DEVCOM ARL
(PDF) FCDD RLD CL
TECH LIB

26 DEVCOM ARL
(PDF) FCDD RLW WE
B ALLIK
B J ACKER
T G BROWN
S BUGGS
E BUKOWSKI
J COLLINS
B DAVIS
M DON
D EVERSON
M GRABNER
R HALL
J HALLAMEYER
M HAMAOU
T HARKINS
M ILG
B KLINE
J MALEY
C MILLER
B NELSON
D PETRICK
K PUGH
N SCHOMER
B TOPPER
FCDD RLW WD
L STROHM
L FAIRFAX
FCDD RLW A
F FRESCONI