# DUAL RADAR SAR CONTROLLER

# USER GUIDE

by

JOSIAH W. SMITH

DUAL RADAR SAR CONTROLLER

USER GUIDE

Josiah W. Smith, PhD
The University of Texas at Dallas, 2022

Supervising Professor:

The following is a user guide for the Dual Radar SAR Controller graphical user interface (GUI) to operate the dual radar synthetic aperture radar (SAR) scanner. The scanner was designed in the Spring semester of 2022 by Josiah Smith (RA), Yusef Alimam (UG), and Geetika Vedula (UG) with multiple axes of motion for the radar and target under test. The system is operated by a personal computer (PC) running MATLAB. An AMC4030 motion controller is employed to control the mechanical system. An ESP32 microcontroller synchronizes the mechanical motion and radar frame firing to achieving precise positioning at high movement speeds; the software was designed by Josiah Smith (RA) and Benjamin Roy (UG). A second system is designed that employs 3-axes of motion (X-Y + rotation) for fine control over the location of the target under test. The entire system is capable of efficiently collecting data from colocated and non-colocated radars for multiband fusion imaging in addition to simple single radar imaging.

TABLE OF CONTENTS

# CHAPTER 1

# GETTING STARTED GUIDE

In this chapter, we detail the steps for getting started with the Dual Radar SAR Controller MATLAB application, referred to as the Dual Radar GUI. This software is capable of connection with multiple Texas Instruments (TI) millimeter-wave (mmWave) frequency-modulated continuous-wave (FMCW) single-chip radars. A simple introduction to FMCW radar signaling is provided in Appendix A and near-field synthetic aperture radar (SAR) imaging algorithms are briefly introduced in Appendix D.

## System Requirements

The following is a set of system requirements for the Dual Radar GUI, where essential requirements are denoted with an asterisk.

- *Windows 10/11 PC (Dual Radar GUI is not currently compatible with MacOS or Linux operating systems).
- *Multiple Ethernet ports (required for use with multiple radars).
- High speed USB 3.X to connect with large number of devices of single USB connection.
- 16 GB RAM, decent CPU, enough storage to save files from scans (on the order of 100s of MB per scan for a typical synthetic aperture dimension).
- NVIDIA GPU (optional, for image reconstruction using machine learning-based algorithms).

## Hardware Set Up

Prior to using the Dual Radar GUI application, the following hardware must be properly configured. The remainder of this guide assumes that the hardware has been configured and is ready.

- Scanner stand must be assembled.
- Mechanical system (linear actuators, stepper motors, stepper drivers, motion controller, and ESP32 synchronizer) must be assembled and connected (See Section 1.2 for more details on the connections).

- 2 DCA1000EVM data capture cards must be configured at unique IP addresses and mounted to the mechanical system. (See Chapter 6 for details to set up each DCA1000EVM).
- The 60 GHz radar (TI IWR6843ISK + MMWAVEICEBOOST) and 77 GHz radar (TI IWR1642BOOST) must be modified using the steps detailed in Chapters 7 and 8, respectively.
- 60 GHz radar (TI IWR6843ISK + MMWAVEICEBOOST) and 77 GHz radar (TI IWR1642BOOST) must be mounted to their corresponding DCA1000EVM data capture cards and connected. (See Section 1.2 for more details on the connections). Physical alignment of the two mounted radars is important to recover images as the relative position of each radar must be taken into account.
- Optionally, the base scanner must be assembled and connected.

**Software Set Up**

The following software is necessary for the set up of the hardware and operation of the Dual Radar GUI.

- TI mmWave SDK 3.5.0.4
- TI Uniflash
- TI mmWave Studio 2.1.1.0
- MATLAB 2021b (recommended, may be compatible with earlier/later releases)

## 1.1 Opening the Dual Radar GUI

After downloading the Dual Radar GUI software, the following steps are required to open the application.

- Open MATLAB and navigate to the downloaded folder containing the application file: `dual_radar_gui.mlapp`.
- Open the `dual_radar_gui.mlapp` file, which will open the MATLAB Application Designer in a separate window.

- In the MATLAB Application Designer window, press "Run" at the top to run the Dual Radar GUI application.

  - If the MATLAB path is set to the proper path (the main folder, containing the application file: `dual_radar_gui.mlapp`), the application will initialize without any issues. Otherwise, if the MATLAB path is incorrect, a selection window will open and the user will be asked to find the folder containing `dual_radar_gui.mlapp`.

  - If the proper version of TI mmWave Studio (2.1.1.0) is installed to the default location `C:\ti\mmwave_studio_02_01_01_00\`, the application will initialize without any issues. Otherwise, if a different version of mmWave Studio is installed or it is installed at a different location, a selection window will open and the user will be asked to find the `mmWaveStudio` folder (typically at `...\mmwave_studio_xx_xx_xx_xx\mmWaveStudio\`).

- After initialization (the correct folders are added to the path, etc.), the GUI will open and the indicator lamps will be red, as none of the devices are connected to the application or configured.

The following sections detail the basic operation of the Dual Radar SAR Controller GUI. Although the GUI is designed for dual radar operation, it can be applied for single radar use cases by ignoring either the 60 GHz or 77 GHz radar, referred to in the GUI as "Radar 1" and "Radar 2," respectively. For single radar use, only one radar must be connected, configured, etc.; however, even if two radars are connected and configured, the scan can be performed using only one radar, if specified by the user.

### 1.2 Connections for Interface with Devices

Since the Dual Radar GUI connects to many serial USB devices, identifying each serial device is crucial to proper operation of the application. Specifically, difference instances of TI radars or AMC4030 motion controllers will appear under an identical name at different COM ports in Device Manager and it is important to distinguish between the two radars and two AMC4030s (in the case of the base scanner).

### 1.2.1 Identifying the TI Radars and AMC4030 Motion Controllers

To identify each TI radar, first open Device Manager by searching "Device Manager" in the start menu. Then, find the most convenient USB connection (usually at a USB hub) for each radar and disconnect both radars. Connect the first radar and two COM ports will appear labeled "XDS110 Class Application/User UART Port" and "XDS110 Class Application/User Data Port." Record the COM port number corresponding to the "XDS110 Class Application/User UART Port" associated with the first radar for later use. Similarly, disconnect the first radar and connect the second radar. Then, Record the COM port number corresponding to the "XDS110 Class Application/User UART Port" associated with the second radar for later use.

To identify the AMC4030 motion controllers, follow the same procedure, which is summarized below:

(1) Disconnect both devices.

(2) Connect one device and record its COM port inside Device Manager for later use.

(3) Disconnect that device, connect the next device, and repeat step (2) until all devices are recorded.

## 1.3 Connecting the Radars

After noting the COM port for each radar, return to the Dual Radar SAR Controller GUI and navigate to the "Radar Setup" tab. This section follows the numerical order of the various panels in this tab, starting with the Serial Connections panel. Pressing "Connect Radar 1" opens a window displaying a list of the connected COM ports. Select the "XDS110 Class Application/User UART Port" corresponding to the 60 GHz IWR1643ISK radar. Similarly, select "Connect Radar 2" and choose the "XDS110 Class Application/User UART Port" corresponding to the 77 GHz xWR1642BOOST radar. If the serial connection is successful, the radar connection lamps in the top right of the GUI will be green.

## 1.4 Configuring the DCA1000EVM Settings

Assuming both DCA1000EVMs have been configured properly following the instructions in Chapter 6, the IP addresses and ports for each DCA1000EVM can be entered into the corresponding fields in the GUI. Then, pressing the "Prepare DCA 1" button prepares the

DCA1000EVM configuration and creates a .json file at
`...\mmwave_studio_xx_xx_xx_xx\mmWaveStudio\PostProc\cf1.json` with the proper configuration and checks the connection of the DCA1000EVM. If the DCA1000EVM is connected at the specified IP addresses and ports, the message "System is Connected" will appear in the MATLAB terminal. If the message "System is Disconnected" appears in the MATLAB terminal, the following may be causing the issue:

- The DCA1000EVM is disconnected from either the USB cable, Ethernet cable, or power cable.
- The DCA1000EVM is configured to a different IP address. (To change the DCA1000EVM IP address, follow the steps in Chapter 6).
- The Ethernet connection on the PC is not properly configured to the DCA1000EVM IP address. (Follow the steps in Section 7.4.2 to ensure the Ethernet port is properly configured).
- **In some cases, the system appears disconnected, despite being properly configured, when multiple DCA1000EVMs are connected. In this case, disconnect the other DCA1000EVM and attempt to prepare the DCA1000EVM again. Usually, it will succeed. Then, connect the other DCA1000EVM and attempt the connection of both DCA1000EVMs again. Typically, both DCA1000EVMs will now connect properly.**

### 1.5 Configuring the Radars

The radar configuration panels are straightforward and follow the corresponding fields in TI mmWave Studio. The user can enter the desired chirp parameters for each radar and press the respective Configuration button for each radar to send the configuration to each radar. The Serial Number for each radar is the last four digits of the serial number, which is used to store the calibration data for each unique radar.

After a configuration is sent to the radar, if a HW trigger capture is made, the radar must be power cycled before another configuration can be sent. This is a limitation of operating the radars with a HW trigger in addition to command line interface (CLI) configuration.

Alternatively, if the Hardware Trigger checkbox is unchecked, a software (SW) trigger used, which automatically ends when the number of frames are sent (or if the capture is stopped for No of Frames = 0 for infinite transmission).

The calibration functionality for each radar is detailed in Chapter 3. Once the radar has been calibrated, its calibration data are saved for reuse in future scans.

At this point, the radars and corresponding DCA1000EVMs have been connected and configured, and the scan can be set up and performed.

## 1.6 Connecting and Configuring the Scanner

First, the user must switch to the "Scanner Setup" tab at the top of the Dual Radar GUI.

The user must connect the AMC4030 motion controller by pressing the "Connect AMC4030" button. A window will appear listing the connected serial devices and their COM ports. If using the base scanner, it is important to select the correct the proper COM port labeled "USB-SERIAL CH340" corresponding to the dual radar scanner. **One way to test if the proper AMC4030 is connected is to press the "Home" button in the "Single Movement" panel on the bottom right of this tab. If the vertical dual radar scanner moves to the home position, then the correct scanner is connected. If the base scanner moves, then you have selected the incorrect COM port. You can disconnect the AMC4030 and select the proper COM port.**

The ESP32 must be connected using the "Connect ESP32" button. The user should select the COM port corresponding to the ESP32.

The AMC4030 must be configured with the appropriate parameters for the given linear actuators and settings of each stepper driver, as detailed in Sections 4.2.1 and 4.2.2, respectively. After entering the proper settings, the user can configure the AMC4030 by pressing the "Configure AMC4030" button, which send the configuration to the AMC4030.

### 1.6.1 Move to Initial Position of Scan

For many cases, the home position is not the ideal starting position for a given scan. Thus, after the AMC4030 is configured, it is recommended to move the platform to a different starting position using the "Single Movment" functionality in this tab. The user can enter a step size (both positive or negative) to move the platform and press the "Send" button to send the command which moves the platform. The "Home" button sends the platform to the home position, effectively resetting the position. It is recommended to reset the platform the home position every time the system is turned on. After the scanner is moved to the initial/starting position for the scan, the configuration can continue.

## 1.7 Configuring the SAR Scan Parameters

Under the "Configure Scan" panel, the user can enter the parameters of the scanner and desired SAR aperture. The parameters are detailed in Table 1.7. After the parameters are entered, the user can configure the scan, which updates the scan dimensions and estimated scan time fields, by pressing the "Configure Scan" button.

Table 1.7: Parameters of SAR scan.

| Parameter | Meaning |
|---|---|
| X-Max Size (mm) | Maximum dimension of linear actuator along the $x$-direction in mm |
| Y-Max Size (mm) | Maximum dimension of linear actuator along the $y$-direction in mm |
| File Name | Name of the scan[a] |
| X-Step Size (mm) | Step size between each sample along the $x$-direction in mm |
| Y-Step Size (mm) | Step size between each sample along the $y$-direction in mm |
| Periodicity | Approximate periodicity along the $x$-direction |
| X-Offset (mm) | Offset from the starting location to start collecting samples in mm[b] |
| $\Delta$X (mm) | Distance between the radars along the $x$-direction in mm |
| Num X-Steps | Number of samples along the $x$-direction |
| Num Y-Steps | Number of samples along the $y$-direction |
| Radar 1 Checkbox | To use radar 1 for scan |
| Radar 2 Checkbox | To use radar 2 for scan |

[a]The scan is saved to a folder with name corresponding to the date of the scan, so names can be reused. The GUI will check if the scan name already exists before performing the scan.

[b]Allows the effect of up-ramp of the acceleration to be diminished.

Additionally, the "Scan Notes" text area is provided for the user to record notes about the specified scan. We have found this helpful for describing the target scene, scanning parameters, etc. for later use. The scan notes are recorded into a text file and saved with the scan data for reference.

## 1.8 Starting the Scan

After the scan is configured and all devices are connected and configured, all the indicator lamps should be green. In this case, the scan is ready to commence and the user can start the command by pressing the "Start Scan" button.

During the scan, the platform will perform a raster back-and-forth scanning motion during which the radars will be triggered at the specified locations to create a uniform grid synthetic aperture elements from which to reconstruct a high-resolution image. The DCA1000EVM must be reset for each horizontal scan, which is currently the largest inefficiency in the scanning process. For each horizontal motion, HW triggers are sent by the ESP32 to each radar. After each radar receives a HW pulse, it sends a frame (typically of 4 chirps). The beat signals are streamed to the DCA1000EVM over LVDS and then to the PC over UDP via Ethernet. The reason we need to reset for each horizontal scan is currently unknown, but after the HW triggers stop at the end of the horizontal motion, either the radar or DCA1000EVM stops. The radar may stop streaming and need to be reset or the DCA1000EVM may reset since the data stops streaming. We have attempted different streaming modes for the DCA1000EVM that are supposed to change the stopping criterion of the DCA1000EVMs, but we have not seen any difference and this is likely a bug or due to operating the hardware in an unsupported manner. Additional system details can be found in [1].

The following are potential known issues that may occur during the scanning process (typically when multiple DCA1000EVMs are uses simultaneously):

- Too many packets are received to a DCA1000EVM: this is a common issue with no known cause or solution. If too many packets are received by the DCA1000EVM, we typically truncate the received data and assume that the extra data can be ignored. This does not seem to degrade image quality substantially.

- Not enough packets are received to a DCA1000EVM: this occurs rarely and also does not have a known cause or solution. If too few packets are received, we terminate the scan and reset the process. Typically, the scan can be attempted again with success after one or two attempts.

### 1.9 Loading the SAR Data

After the scan completes successfully, the user can press the "Load Data" button to load the data and store it to a convenient format for later processing. Data are saved to `data\<date>\<fileName>\` with MATLAB files to load the data and reconstruct the images using the radar imaging toolbox, as detailed in Chapter 2.

# CHAPTER 2

# RECONSTRUCTING IMAGES FROM DUAL RADAR SAR SCANS

Using the dual radar GUI, scans can be performed using 1 or 2 radars as specified in the previous chapter. In the following sections, we detail how to reconstruct images from each of the radars or using multiband signal fusion algorithms. The following steps assume that the MATLAB path remains located in the main `dual-radar-gui` folder regardless of which file is opened.

## 2.1 Reconstruct Image from Radar 1 (60 GHz)

To reconstruct the image using radar 1, open the file:

`data\<date>\<fileName>\<fileName>_radar1.m`.

The script first attempts to add the proper image reconstruction toolbox and then proceeds to load the data. When calling `DualRadarLoadAll()`, the second argument indicates the method for loading the radar data. When the value is 1.5, the algorithm interpolates the data to a uniform grid with $\lambda/4$ spacing for 77 GHz along the vertical direction. For a value of 1, the algorithm leaves the spacing with the original 60 GHz $\lambda/4$ vertical spacing.

The remaining sections of the script are self-explanatory to set the imaging parameters (volume of interest), number of voxels, display parameters, etc.

## 2.2 Reconstruct Image from Radar 2 (77 GHz)

To reconstruct the image using radar 2, open the file:

`data\<date>\<fileName>\<fileName>_radar2.m`.

The script first attempts to add the proper image reconstruction toolbox and then proceeds to load the data. When calling `DualRadarLoadAll()`, the second argument indicates the method for loading the radar data. For a value of 2, the algorithm leaves the spacing with the original 77 GHz $\lambda/4$ vertical spacing.

The remaining sections of the script are self-explanatory to set the imaging parameters (volume of interest), number of voxels, display parameters, etc.

## 2.3 Reconstruct Image with Both Radars using Matrix Fourier Transform (MFT)

To reconstruct the image with both radars using the matrix Fourier transform (MFT) method detailed in [2] and discussed in the matrix pencil algorithm paper [3], open the file:

`data\<date>\<fileName>\<fileName>_dual_radar.m`.

The script first attempts to add the proper image reconstruction toolbox and then proceeds to load the data. When calling `DualRadarLoadAll()`, the second argument indicates the method for loading the radar data. For a value of 3, the algorithm loads in both the 60 and 77 GHz radar data, applies multistatic-to-monostatic conversion [4, 5] to both sets of data, interpolates the 60 GHz data to an identical sampling grid as the 77 GHz radar, and zero-pads between the samples.

The remaining sections of the script are self-explanatory to set the imaging parameters (volume of interest), number of voxels, display parameters, etc.

### 2.4 Reconstruct Image using Super-Resolution Algorithm

Software requirements for Josiah Smith super-resolution algorithms:

1) Cuda 11.6

2) PyTorch 1.11

3) Install requirements.txt from the `dual-radar-fusion-SR` repository

After loading in the dual radar data as for the MFT method, a super-resolution algorithm can be applied. A typical call to the super-resolution algorithm with recommended settings is included below:

```
DualRadarSR(target,"./saved/fftnet1055.tar",4096,"range","min-max","norm3",1.2);
```

# CHAPTER 3

# CALIBRATING A MMWAVE RADAR

In this chapter, the calibration process using the dual radar GUI is discussed. Mathematical details of the calibration procedure can be found in [1, 4, 5].

After the radar and DCA1000EVM are connected and configured, following Sections 1.3 through 1.5, each radar can be calibrated using the dual radar GUI. Each radar must be calibrated before it can be used for scanning to compensate for constant phase errors and range bias. The serial number of the radar is used in the GUI to store the calibration data associated with that radar. It is recommended to calibrate the radar every time the system is power cycled; however, this may be unnecessary.

To calibrate the radar, a corner reflector is required.

Calibration steps:

1) Connect and configure radar.

2) Move radar to the correct position ($x$ and $y$) using the Single Command functionality of the scanner.

3) Place corner reflector in front of radar (recommended distance of 300 mm from the radar boresight) such that the center of the corner reflector is aligned vertically with the lowest RX element.

4) Follow calibration steps in GUI (calibrate empty scene (optional): moves radar to different position to capture samples of an empty scene / background clutter of surrounding area).

# CHAPTER 4

# MECHANICAL SYSTEM

In this chapter, we detail the mechanical system on which the radars are mounted. The system scans both radars in a raster pattern and synchronizes the triggers such that a uniform grid is achieved by both radars.

## 4.1 Physical Scanner Stand

## 4.2 Scanner Motion

The mechanical scanner moves the radars along a 2-D $x$-$y$ plane using linear actuators and stepper motors, which are driven by stepper drivers.

### 4.2.1 MJUnit Linear Actuators

The 8020 stand discussed in Section 4.1 holds the linear actuators. The linear actuators used are MJUnit MJ45 or MJ50 rails, which are belt driven and are capable of speeds upward of 200 mm/s. The large dual-radar scanner uses MJ50 rails, which are shown in Fig. 4.1.

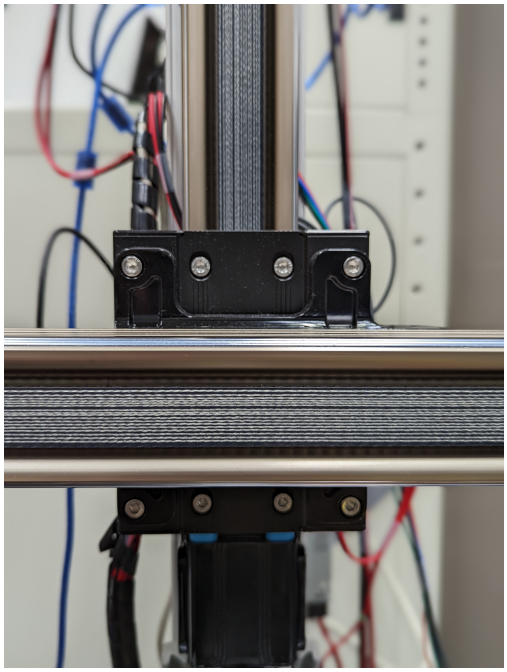


Figure 4.1: MJUnit MJ50 $x$-$y$ platform mounted to scanner stand.

### Linear Actuator AMC4030 Settings

Each linear actuator type has a different gear ratio that varies the ratio of millimeters per revolution. This ratio is important for the AMC4030 motion controller for converting the rotational motion accurately to linear (millimeters). Table 4.1 details the required settings that must be applied for each linear actuator type. See [1] for more details and comparison of various actuator types.

Table 4.1: AMC4030 Software Settings for Each Linear Actuator Type

| Actuator Type | DIST Parameter |
|---|---|
| MJ50 | 110 mm/rev |
| MJ45 | 110 mm/rev |
| Rotator | 36 mm/rev (converts mm to deg) |
| Old Screw Rail (purchased by M. E. Yanik) | 5 mm/rev |

### 4.2.2 Stepper Drivers

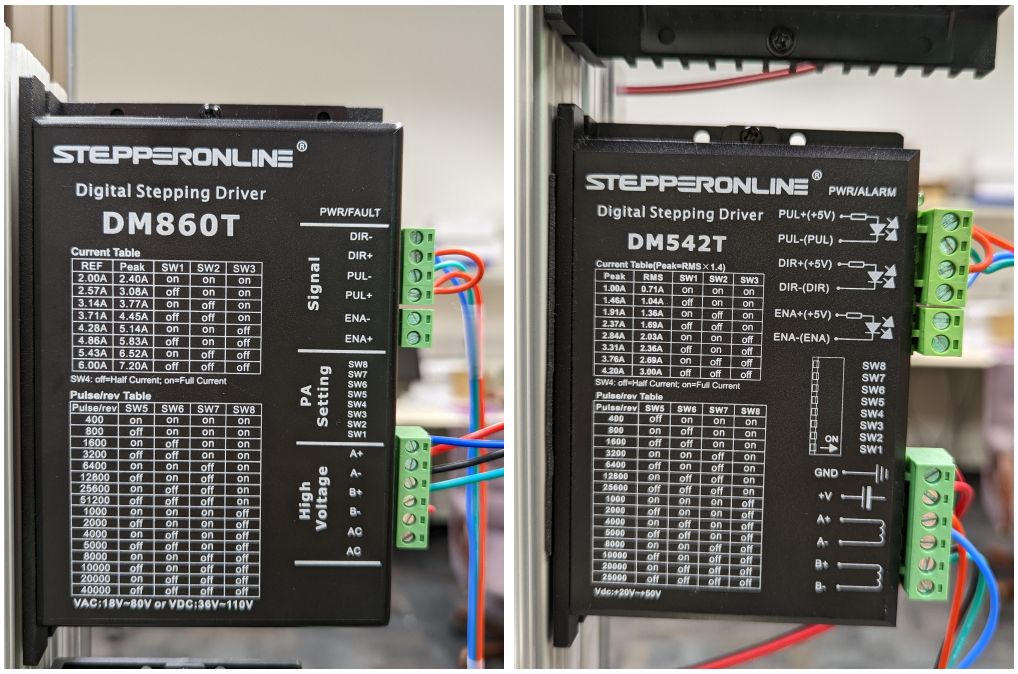
Two varieties of stepper drivers are used for the dual radar scanner. A high-power stepper driver is employed for the larger stepper motor detailed in the following section and a smaller stepper driver is used with the smaller stepper motor.

The larger stepper driver used to drive the larger NEMA34 motor operating the vertical $y$-axis is the DM860T, as shown in Fig. 4.2a. A 50 V power supply is employed to power the stepper driver and stepper motor. The larger motor must be used for the vertical axis due to the load of the horizontal linear actuator. The weight of the radars is small compared to the heavy horizontal linear rail.

The NEMA23 motor operating the horizontal $x$-axis is driven by a DM542T stepper driver, as shown in Fig. 4.2b. A 24 V power supply is employed to power this stepper driver and stepper motor. The smaller motor can be used for the horizontal axis because the required torque is lower for the horizontal motion.

### Stepper Driver Settings

The stepper driver settings for each driver are listed on the back of the stepper drivers and are pictured in Fig. 4.2. There are two primary settings for the stepper drivers: 1) available current and 2) pulses per revolution. Both settings are controlled by a set of DIP switches on the side of the stepper driver between the wiring connections (to the right in the images in Fig. 4.2) and are detailed as follows:




(a) (b)

Figure 4.2: Stepper drivers: (a) DM860T high-power stepper driver for driving the NEMA34 motor at the vertical $y$-axis. (b) DM542T stepper driver for driving the NEMA23 motor at the horizontal $x$-axis.

1) **Current:** We recommend setting the stepper driver such that the maximum available current is provided to the stepper motor. This is not thoroughly tested; however, we find that, particularly for the vertical $y$-axis, low current can cause incorrect positioning due to the heavy load. Typically, this setting requires setting switches SW1, SW2, and SW3 to the off position. *NOTE: Please check with the stepper motor documentation to ensure that the current setting is compatible and will not damage the stepper motor.*

2) **Pulses Per Revolution:** the pulses per revolution (pulse/rev) setting controls the precision of the motion. If the stepper driver is set to $N$ pulses/rev, the stepper motor will complete exactly 1 rotation when the AMC4030 sends $N$ pulses. Using the analysis from Section 4.2.1, the conversion from pulses to linear motion (millimeters) can be computed as follows. Let $M$ be the mm/rev setting corresponding to the linear actuator being used, as shown in Table 4.1. The linear distance per pulse can be computed as $M/N$. For example, if an MJ50 linear actuator is used, such as for the large dual radar scanner, $M = 110$ mm/rev and suppose $N = 20000$ pulse/rev. Hence, the conversion is $M/N = 0.0055$ mm/pulse. This computation is necessary for synchronizing the

motion using the synchronizer software discussed in Section 4.4 [1]. We recommend using a high value of number of pulses to achieve highly precise positioning since the wavelength is on the order of millimeters and typical spacing is $\lambda/4$, where $\lambda$ is the wavelength of the center frequency (62 GHz or 79 GHz).

### 4.2.3 Stepper Motors


(a) (b)

Figure 4.3: Stepper motors: (a) NEMA34 motor at the vertical $y$-axis. (b) NEMA23 motor at the horizontal $x$-axis.

The stepper motors are shown in Fig. 4.3. A NEMA34 motor is used for the vertical $y$-axis and a NEMA23 motor is used for the horizontal $x$-axis. The motors are connected to the linear actuators using the black metal housing shown in Fig. 4.3. Within the housing is a cylindrical connector that tightens around each shaft (of the linear actuator and stepper motor). There are many sizes of these cylindrical connectors corresponding to different stepper motor shaft diameters. *NOTE: The connector must be tightened very tightly around each shaft or the connection will not be strong enough for the motion, particularly for the vertical y-axis. However, the connectors are somewhat fragile, so be careful when tightening not to strip out the bolt heads or damage the connector.*

The wiring of the stepper motors and stepper drivers is discussed later in Section 4.3.

### 4.2.4 AMC4030 Motion Controller

The motion of the entire system is controlled by a FUYU AMC4030 motion controller. The AMC4030 motion controller may be obscure, but it has an excellent MATLAB API that allows for direct control from MATLAB, enabling automation of scanning patterns and synchronization. The AMC4030 connections are detailed later in Section 4.3.

#### AMC4030 Software

The AMC4030 software can be downloaded from the FUYU website `https://www.fuyumotion.com/manual/`. We have had difficulty opening the software on certain computers and are not sure why that is. However, even though the AMC4030 software GUI does not open, the API interface does work and the dual radar GUI and still control the scanner.

The AMC4030 shows up in Windows' Device Manager as a CH340. The driver included by FUYU does not always work for some reason, but we have had success with the driver at this website `https://sparks.gogo.co.nz/ch340.html`.

### 4.2.5 ESP32 Microcontroller

The ESP32 microcontroller is used to synchronize the radar transmissions along the horizontal motion. The synchronizer, detailed in Section 4.4, overcomes two primary issues: 1) uniform spacing of radar transmissions without constant velocity and 2) synchronizing both radars to transmit at the same physical locations although being horizontally displaced.

The ESP32 is mounted to a breakout board, as shown in Fig. 4.4 to make the connections to accessible for each GPIO pin. The wiring of the ESP32 is discussed in Section 4.3 and the synchronizer software is detailed in Section 4.4.

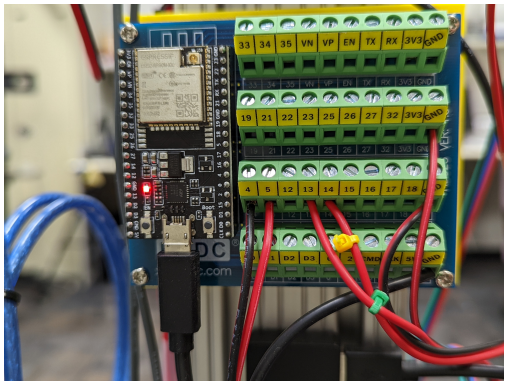



Figure 4.4: ESP32 microcontroller wired for dual radar synchronization.

## 4.3 Wiring

The wiring of the mechanical system is crucial for proper operation and should be one of the first steps in debugging when an issue arises. This section details the various connections required for the entire system.

A high-level view of the wiring is shown in Fig. 4.5. Although this figure does not include the power connections for each element, the required power connections are detailed for each component in the following sections.

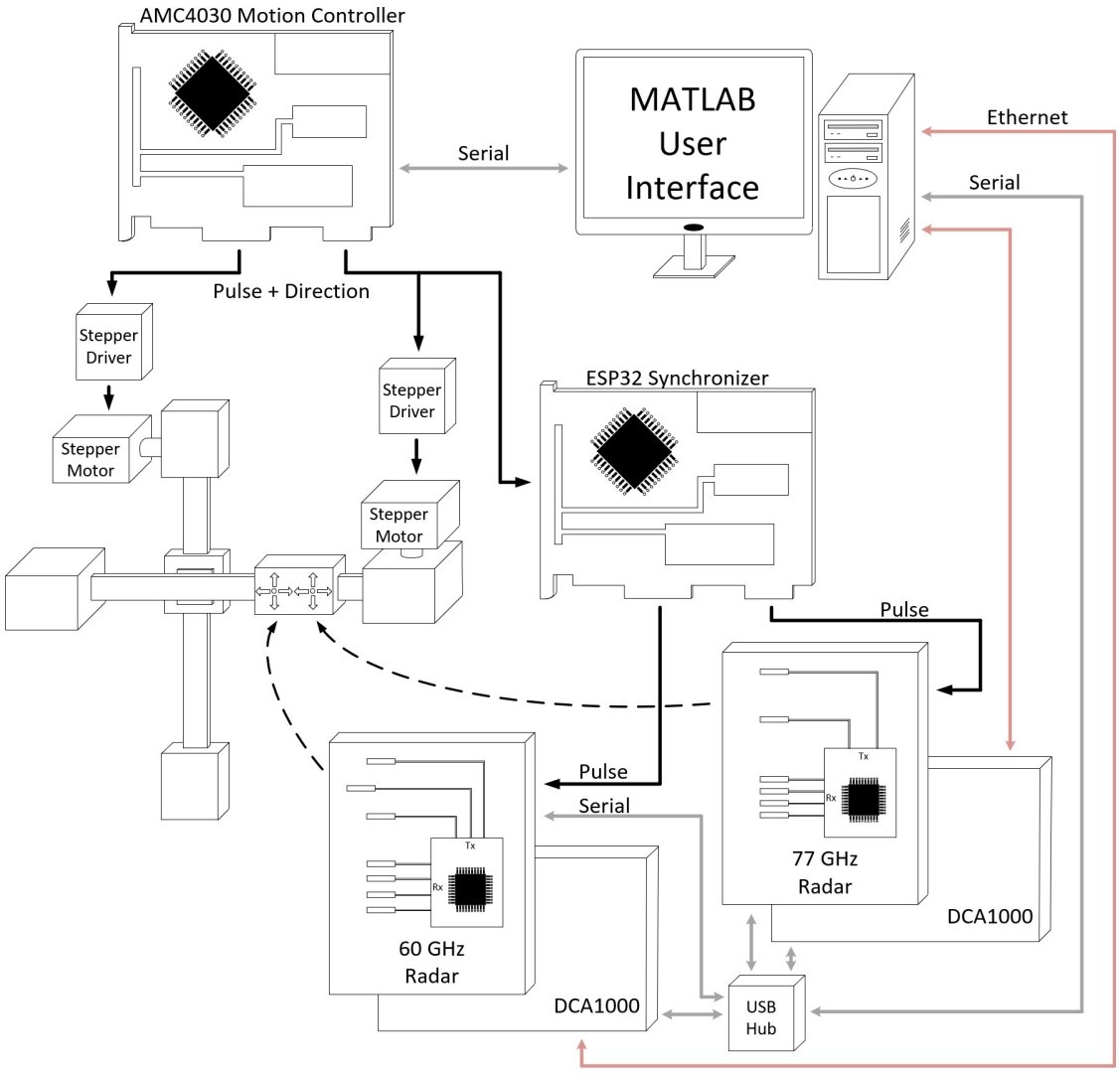
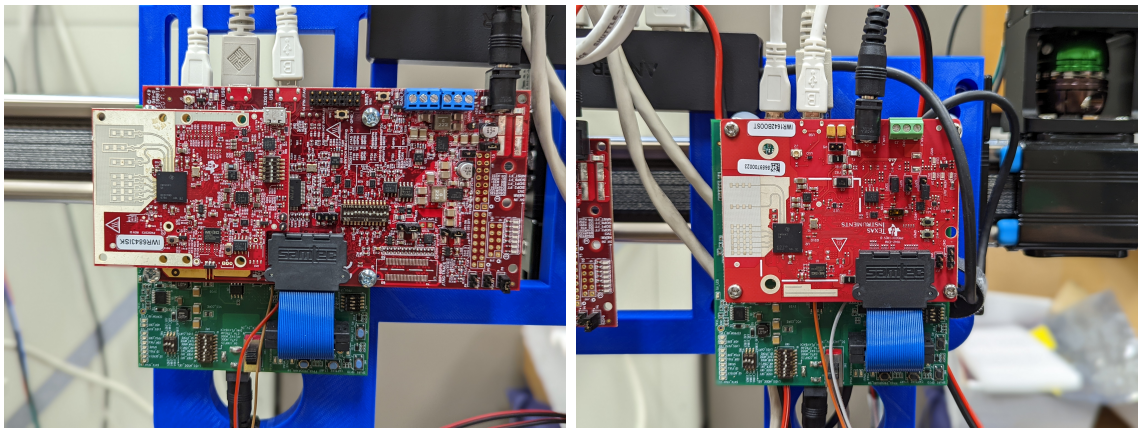



Figure 4.5: High-level diagram of the wiring required for the dual radar system. This illustration neglects power required for each device.

### 4.3.1 Radar and DCA1000EVM Wiring

As shown in Fig. 4.6, both radars are mounted to DCA1000EVMs, which are mounted to the back plate (blue 3-D printed PLA plastic). Each DCA1000EVM must be set to a different IP address, as discussed in Chapter 6.



(a) (b)

Figure 4.6: Radars: (a) 60 GHz radar (considered radar 1). (b) 77 GHz radar (considered radar 2). Both radars are mounted to DCA1000EVM boards via the 60-pin blue cables shown. Each DCA1000EVM is programmed to a different IP address following Chapter 6.

The required connections are detailed as follows:

1. **Connection between radars and DCA1000EVMs:** the radars are connected to the DCA1000EVMs using the 60-pin blue cables.

2. **Connection between radars and PC:** the radars are connected to the PC via a serial connection. The white USB cables shown are connected to a USB hub attached to the blue back plate. A USB extension cable connects that USB hub to another USB hub mounted to the dual radar scanner stand. The second USB hub is connected to the PC. The correct USB connection on each radar is detailed in Chapters 7 and 8.

3. **Connection between DCA1000EVMs and PC:** the DCA1000EVMs are connected to the PC via a serial connection and Ethernet connection. The USB connection is identical to that between the radars and PC as the DCA1000EVMs connect via USB to the same USB hub attached to the blue back plate. The Ethernet connection is made directly to the PC. Two long Ethernet cables are used to make this connection.

4. **Connection between radars and ESP32:** the radars are connected to the ESP32 synchronizer via simple cables capable of a 3.3 V or 5 V signal. The HW pulse is sent from the ESP32 to trigger the radar at the proper spatial location to create a uniform grid despite the non-uniform motion. The connection of each radar is different and are detailed in Chapters 7 and 8. *NOTE: Connecting the ESP32 to the radars is EXTREMELY important. We recommend testing the connection with all the cables prior to assembling any system.*

5. **Power connection for the radars and DCA1000EVMs:** the radars and DCA1000EVMs each require a 5 V  3 A power supply. The power is provided to the blue back plate by a 24 V power supply, detailed in Section 4.3.7, which is connected to a downconverter to provide a 5 V power supply. The connection is split off as necessary and connectors are provided for each radar and DCA1000EVM.

### 4.3.2 ESP32 Wiring

A properly wired ESP32 is shown in Fig. 4.7. The ESP32 is connected to the AMC4030 to count the direction and pulses sent to the stepper driver and is connected to the radars to send the HW trigger pulses.

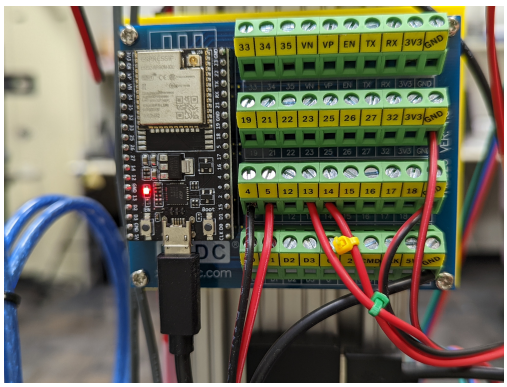


Figure 4.7: ESP32 microcontroller wired for dual radar synchronization.

The required connections are detailed as follows:

1. **Connection between ESP32 and AMC4030:** the ESP32 is connected to the AMC4030 to monitor the direction and pulse pins in order to count the pulses sent to the stepper driver. The connections are shown in Fig. 4.8. The ESP32 is connected with the black and dark red wires to the DIR1 and PUL1 pins on the AMC4030. The

black wire (connected to DIR1 on the AMC4030) is connected to GPIO pin 4 on the ESP32 and the red wire (connected to PUL1 on the AMC4030) is connected to the GPIO pin 5 on the ESP32, as shown in Fig. 4.7. Additionally, the ESP32 must be connected to the ground of the AMC4030 to create a common ground.

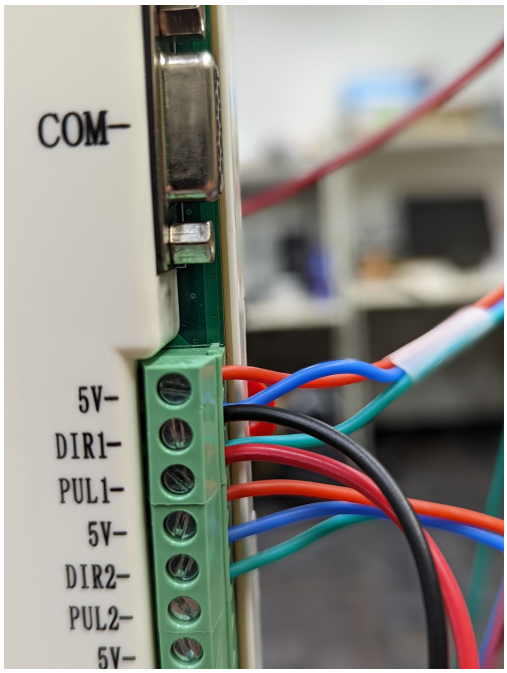



Figure 4.8: ESP32 connection with the AMC4030. The ESP32 is connected with the black and dark red wires to the DIR1 and PUL1 pins on the AMC4030. The black wire (connected to DIR1 on the AMC4030) is connected to GPIO pin 4 on the ESP32 and the red wire (connected to PUL1 on the AMC4030) is connected to the GPIO pin 5 on the ESP32, as shown in Fig. 4.7.

2. **Connection between ESP32 and radars:** the radars are connected to the ESP32 synchronizer via simple cables capable of a 3.3 V or 5 V signal. The HW pulse is sent from the ESP32 to trigger the radar at the proper spatial location to create a uniform grid despite the non-uniform motion. The connection of each radar is different and are detailed in Chapters 7 and 8. As shown in Fig. 4.7, radar 1 (60 GHz radar indicated with the green zip tie) is connected to GPIO pin 13 on the ESP32 and radar 2 (77 GHz radar indicated with the yellow zip tie) is connected to GPIO pin 13 on the ESP32. Additionally, both radars must be connected to the ground of the ESP32 to create a common ground. *NOTE: Connecting the ESP32 to the radars is EXTREMELY important. We recommend testing the connection with all the cables prior to assembling any system.*

3. **Connection between the ESP32 and PC:** the ESP32 is powered by the PC and receives commands from the PC for the various motion states of the scanner. The connection is made over a serial USB cable that is attached from the ESP32 to the USB hub mounted on the scanner stand and then to the PC.

4. **Power connection for the ESP32:** the ESP32 is powered by the USB connection with the PC.

### 4.3.3 AMC4030 Wiring

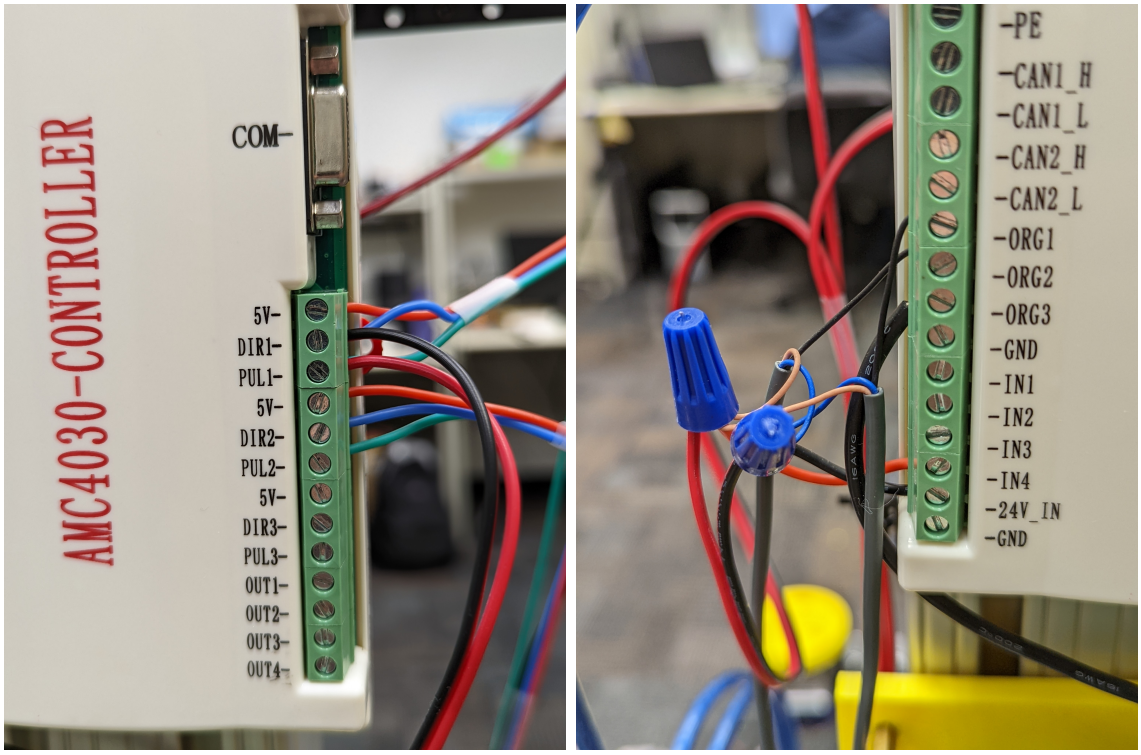
The AMC4030 receives commands from the PC, sends pulses to the stepper drivers, is connected to the ESP32 for synchronization, and is connected to homing switches to set the home position of the scanner.

The required connections are detailed as follows:

1. **Connection between AMC4030 and stepper drivers:** the stepper drivers are connected to AMC4030 using 3 wires (red, blue, and green), as shown in Figs. 4.9a and 4.10. The red wire is for a constant high (5 V) signal and is connected to the stepper driver's PUL+ and DIR+ pins and to the AMC4030's 5 V pin. The blue wire is for the direction pin, which indicates which direction the motion is to take, and is connected to the DIR- pin on the stepper driver and the DIR pin on the AMC4030. The green wire is for the pulses sent from the AMC4030 to the stepper driver and is connected to the PUL- pin on the stepper driver and the PUL pin on the AMC4030. The horizontal $x$-axis is connected to axis 1 on the AMC4030 and the vertical $y$-axis is connected to axis 2 on the AMC4030.

2. **Connection between AMC4030 and homing switches:** two homing switches are connected to the AMC4030, 1 for each of the for the $x$- and $y$-axes. The wiring is shown in Fig. 4.9b and the homing switch is shown in Fig. 4.11 properly mounted to the linear actuator. The homing switch blue wire is connected to a 24 V power supply. The light brown wire on the homing switch is connected to ground. The black wire on the homing switch is connected to the ORG1 or ORG2 pin on the AMC4030 corresponding to the horizontal $x$-axis or vertical $y$-axis, respectively.

3. **Connection between ESP32 and AMC4030:** the ESP32 is connected to the AMC4030 to monitor the direction and pulse pins in order to count the pulses sent to the stepper driver. The connections are shown in Fig. 4.8. The ESP32 is connected




(a) (b)

Figure 4.9: AMC4030 wiring: (a) AMC4030 connections with the stepper drivers and ESP32. See Figs. 4.10 and 4.7 for corresponding connections on the stepper drivers and ESP32, respectively. (b) AMC4030 connection to the homing switches. The homing switch blue wire is connected to a 24 V power supply. The light brown wire on the homing switch is connected to ground. The black wire on the homing switch is connected to the ORG1 or ORG2 pin on the AMC4030 corresponding to the horizontal $x$-axis or vertical $y$-axis, respectively.

with the black and dark red wires to the DIR1 and PUL1 pins on the AMC4030. The black wire (connected to DIR1 on the AMC4030) is connected to GPIO pin 4 on the ESP32 and the red wire (connected to PUL1 on the AMC4030) is connected to the GPIO pin 5 on the ESP32, as shown in Fig. 4.7. Additionally, the ESP32 must be connected to the ground of the AMC4030 to create a common ground.

4. **Connection between AMC4030 and PC:** the AMC4030 is connected to the PC over a serial USB cable connected to the USB hub mounted to the scanner stand.

5. **Power connection for the AMC4030:** the AMC4030 is powered by a 24 V power supply, as shown in Fig. 4.9b. The power supply is connected to the 24V_IN and GND pins on the AMC4030.

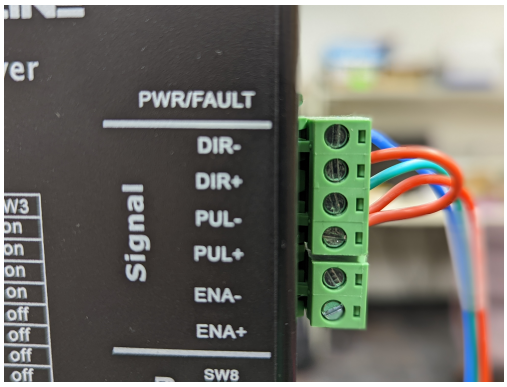



Figure 4.10: Stepper driver connection to the AMC4030. The red wire is for a constant high (5 V) signal and is connected to the stepper driver's PUL+ and DIR+ pins and to the AMC4030's 5 V pin. The blue wire is for the direction pin, which indicates which direction the motion is to take, and is connected to the DIR- pin on the stepper driver and the DIR pin on the AMC4030. The green wire is for the pulses sent from the AMC4030 to the stepper driver and is connected to the PUL- pin on the stepper driver and the PUL pin on the AMC4030. See Fig. 4.9a for the corresponding connections on the AMC4030. The horizontal $x$-axis is connected to axis 1 on the AMC4030 and the vertical $y$-axis is connected to axis 2 on the AMC4030.

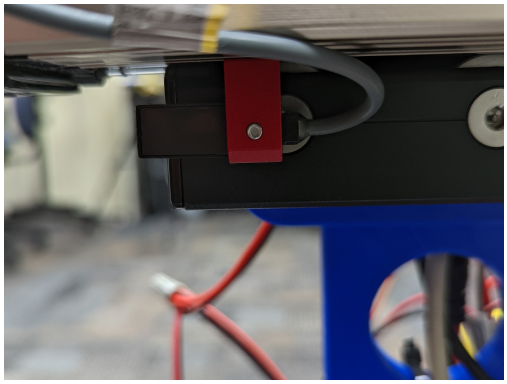


Figure 4.11: Homing switch connected to AMC4030 and mounted to linear actuator.

### 4.3.4 Stepper Driver Wiring

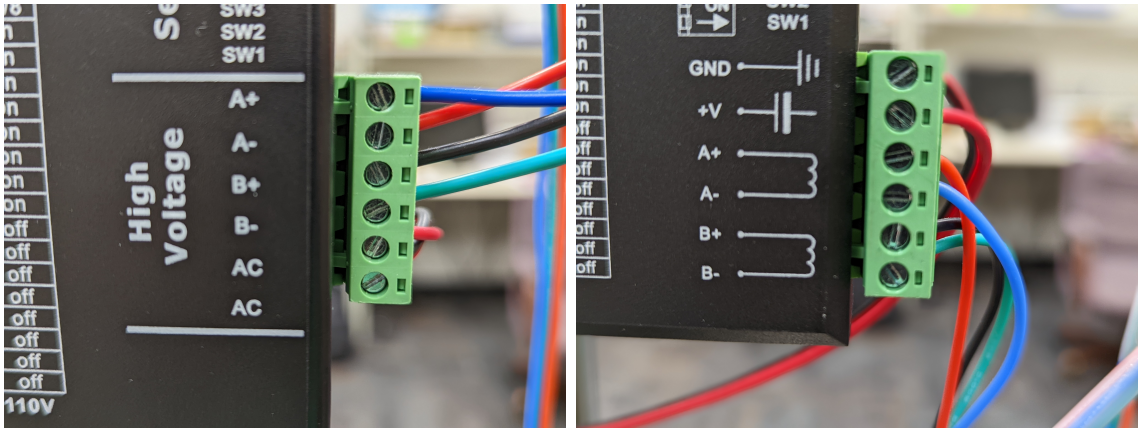
The stepper driver is connected to the AMC4030 and stepper motors.

The required connections are detailed as follows:

1. **Connection between AMC4030 and stepper drivers:** the stepper drivers are connected to AMC4030 using 3 wires (red, blue, and green), as shown in Figs. 4.9a and 4.10. The red wire is for a constant high (5 V) signal and is connected to the stepper driver's PUL+ and DIR+ pins and to the AMC4030's 5 V pin. The blue wire

is for the direction pin, which indicates which direction the motion is to take, and is connected to the DIR- pin on the stepper driver and the DIR pin on the AMC4030. The green wire is for the pulses sent from the AMC4030 to the stepper driver and is connected to the PUL- pin on the stepper driver and the PUL pin on the AMC4030. The horizontal $x$-axis is connected to axis 1 on the AMC4030 and the vertical $y$-axis is connected to axis 2 on the AMC4030.

2. **Connection between the stepper drivers and stepper motors:** the stepper drivers are connected the stepper motors using a standard 4 wire connection, as shown in Fig. 4.12. The wiring is shown with two different configurations. The A+ and A- pins on the stepper driver should be connected to the red and blue wires of the stepper motor, and the B+ and B- pins on the stepper driver should be connected to the black and green wires of the stepper motor. The direction of the stepper motor can be reversed by switching the connection of either the A or B pair of wires, as shown in Figs. 4.12a and 4.12b.




(a) (b)

Figure 4.12: Stepper driver connection with stepper motors: (a) DM860T connection with NEMA34 stepper motor at the vertical $y$-axis. (b) DM542T connection with the NEMA23 stepper motor at the horizontal $x$-axis. The wiring is shown with two different configurations. The A+ and A- pins on the stepper driver should be connected to the red and blue wires of the stepper motor, and the B+ and B- pins on the stepper driver should be connected to the black and green wires of the stepper motor. The direction of the stepper motor can be reversed by switching the connection of either the A or B pair of wires, as shown in (a) and (b).

3. **Power connection for the stepper drivers:** The stepper drivers are connected to either a 24 V or 50 V power supply on the pins as shown in Fig. 4.12.

### 4.3.5 Stepper Motor Wiring

The stepper motors are connected to the stepper drivers and mounted to the linear actuators, as shown in Fig. 4.3. See 4.2.3 for more details regarding the mounting.

The required connections are detailed as follows:

1. **Connection between the stepper drivers and stepper motors:** the stepper drivers are connected the stepper motors using a standard 4 wire connection, as shown in Fig. 4.12. The wiring is shown with two different configurations. The A+ and A- pins on the stepper driver should be connected to the red and blue wires of the stepper motor, and the B+ and B- pins on the stepper driver should be connected to the black and green wires of the stepper motor. The direction of the stepper motor can be reversed by switching the connection of either the A or B pair of wires, as shown in Figs. 4.12a and 4.12b.

2. **Power connection for the stepper motors:** the power for the stepper motors is provided from the stepper drivers. There is no need for any additional power connections.

### 4.3.6 PC Wiring

The PC controls the entire scan and must be connected to the radars, DCA1000EVMs, ESP32, and AMC4030. The PC must be powered on, running Windows (10 is recommended, 11 is not tested), and MATLAB 2021 or 2022 recommended.

The required connections are detailed as follows:

1. **Connection between radars and PC:** the radars are connected to the PC via a serial connection. The white USB cables shown are connected to a USB hub attached to the blue back plate. A USB extension cable connects that USB hub to another USB hub mounted to the dual radar scanner stand. The second USB hub is connected to the PC. The correct USB connection on each radar is detailed in Chapters 7 and 8.

2. **Connection between DCA1000EVMs and PC:** the DCA1000EVMs are connected to the PC via a serial connection and Ethernet connection. The USB connection is identical to that between the radars and PC as the DCA1000EVMs connect via USB to the same USB hub attached to the blue back plate. The Ethernet connection is made directly to the PC. Two long Ethernet cables are used to make this connection.

3. **Connection between the ESP32 and PC:** the ESP32 is powered by the PC and receives commands from the PC for the various motion states of the scanner. The connection is made over a serial USB cable that is attached from the ESP32 to the USB hub mounted on the scanner stand and then to the PC.

4. **Connection between AMC4030 and PC:** the AMC4030 is connected to the PC over a serial USB cable connected to the USB hub mounted to the scanner stand.

#### 4.3.7 Power Supply Wiring

The dual radar scanner uses 4 power supplies: 3x 24 V power supplies and 1x 50 V power supply. The power supplies mounted to the base of the scanner stand. The power supplies are connected to the radars, DCA1000EVMs, AMC4030, and stepper drivers as discussed in the Sections above.

### 4.4 Dual Radar SAR Position Synchronizer

The dual radar synchronizer is a crucial element to the system as it overcomes two integral challenges for dual radar scanning:

1. At high speeds, the linear actuators do not follow a simple constant velocity model, but rather accelerate at the start of each motion and decelerate at the end of each motion [1]. Hence, if the radar is programmed to trigger at regular time intervals (uniform periodicity), the positions of each trigger will not follow a regular uniform grid. Rather, they will be spaced unevenly and the position of each sample is unknown. The synchronizer tracks the position of the platform regardless of the velocity and acceleration and sends HW triggers to the radars so they trigger at regular spatial intervals to create an ideal uniform grid.

2. To recover an image, the position of both radars must be known at each sample. The synchronizer tracks the position of the platform and sends a HW trigger to each radar when that radar reaches the next breakpoint so as to align the captures horizontally.

#### 4.4.1 Summary of Functionality

The dual radar synchronizer is written in FreeRTOS and implemented on an ESP32.

**Position Estimation via Pulse Counting:** The synchronizer counts the pulses sent from the AMC4030 to track the exact position of the platform across the horizontal motion.

The AMC4030 sends pulses unevenly following the acceleration profile, as discussed in [1]. The synchronizer can convert the number of pulses to spatial distance using the analysis in Section 4.2.2.

**HW Trigger Sent to Radars:** The synchronizer sends a HW pulse to the radars when the platform reaches a breakpoint. Each breakpoint is specified by the user to denote the spatial sampling interval. Knowing the relative distance between the radars, the synchronizer tracks the location of each radar independently and sends the HW trigger when each radar meets the next breakpoint. Hence, the scanner can perform two-direction scanning where the synchronizer counts up and then down and sends the HW trigger pulse at the specified breakpoints.

# CHAPTER 5

# USING THE BASE SCANNER

The base scanner GUI can be accessed under the Tools menu in the dual radar GUI. Functionality for operating the base scanner is simple and detailed in the base scanner GUI. At the time of writing this manual, the base scanner is being built and the functionality is limited. Over the summer of 2022, Yusef Alimam will be attempting to automate data collection using the base scanner and will improve the functionality.

# CHAPTER 6

# CHANGING DCA1000EVM IP ADDRESSES AND PORTS

The following is a simple guide for changing the configuration of a DCA1000EVM. To operate multiple DCA1000EVM devices on the same PC, the IP addresses and ports associated with each DCA1000EVM must be unique. Hence, these steps must be carefully followed to ensure each DCA1000EVM is properly configured for multi-radar coordination.

## 6.1 If the Current DCA1000EVM IP Address is Unknown

If the current IP address is unknown, the following steps are necessary to reset the DCA1000EVM to factory defaults prior to setting a new IP address.

1) Place SW2.6 to the ON position (towards pin 11).

2) Power cycle the DCA1000EVM (This loads the default Ethernet settings from the DCA1000EVM's FPGA).

3) The IP address for the DCA1000EVM is now 192.168.33.180.

4) Set the IPv4 address on active PC LAN port to 192.168.33.30 (see the DCA1000EVM Quick Start Guide).

    a) Once the device is properly connected to the PC, open the start menu and search "View Network Connections," as shown in Fig. 7.3.

    b) Inside the "Network Connections" of Control Panel, right click on the Ethernet port of choice and select "Properties."

    c) "Local Area Connection Properties" window will open. Right click on "Internet Protocol Version 4 (TCP/IPv4)."

    d) "Internet Protocol Version 4 (TCP/IPc4)" window will open. Set the IP address field to 192.168.33.30.

    e) The Subnet mask field can remain the default 255.255.255.0.

    f) Press "OK" on all the windows and you can close "Network Connections."

5) Follow the instructions in the following Section to set a new IP address to the DCA1000EVM.

### 6.2 Setting New IP Addresses and Ports to the DCA1000EVM

1) Modify DCA1000EVM configuration file (.json). The typical location is "C:/ti/mmwave_studio_xx_xx_xx_xx/mmWaveStudio/PostProc/."

   a) Lines 9–20 will have the format given in Fig. 6.1.

```
[9]   "ethernetConfig": {
[10]  "DCA1000IPAddress": "192.168.33.180",          < - - current DCA IP Address
[11]  "DCA1000ConfigPort": 4096,
[12]  "DCA1000DataPort": 4098
[13]  },
[14]  "ethernetConfigUpdate": {
[15]  "systemIPAddress": "192.168.33.30",            < - - future System IP Address
[16]  "DCA1000IPAddress": "192.168.33.180",          < - - future DCA IP Address
[17]  "DCA1000MACAddress": "12.34.56.78.90.12",
[18]  "DCA1000ConfigPort": 4096,
[19]  "DCA1000DataPort": 4098
[20]  },
```

Figure 6.1: Lines 9–20 of the .json configuration file used to change the DCA1000EVM IP addresses and ports.

   b) Ensure that the IP address on line 10 matches the current IP address of the DCA1000EVM.

   c) Change the IP addresses on lines 15 - 16 to the new IP address.

   d) Save the json file under a new name, such as: newIP.json.[1]

2) Send the new configuration file to DCA1000EVM.

   a) Power cycle the DCA1000EVM.

   b) Open Powershell.

   c) Use the following commands to enter the correct directory and update the DCA1000EVM's EEPROM:

```
cd "C:\ti\mmwave_studio_xx_xx_xx_xx\mmWaveStudio\PostProc\"
```

[1]If you want to use different IP address with mmWave Studio, make sure the json is called `cf.json`. mmWave Studio will use and modify the `cf.json` in the `PostProc` folder. Hence, it is recommended that you use different .json files for each radar named `cf1.json`, for example.

```
.\DCA1000EVM_CLI_Control.exe eeprom newIP.json
```

3) Set the IPv4 address on active PC LAN port to the new System IP Address (see the DCA1000EVM Quick Start Guide).
    a) Once the device is properly connected to the PC, open the start menu and search "View Network Connections," as shown in Fig. 7.3.
    b) Inside the "Network Connections" of Control Panel, right click on the Ethernet port of choice and select "Properties."
    c) "Local Area Connection Properties" window will open. Right click on "Internet Protocol Version 4 (TCP/IPv4)."
    d) "Internet Protocol Version 4 (TCP/IPc4)" window will open. Set the IP address field to 192.168.xxx.xxx (the **NEW** DCA1000EVM System IP Address).
    e) The Subnet mask field can remain the default 255.255.255.0.
    f) Press "OK" on all the windows and you can close "Network Connections."
4) Update the .json file
    a) In the .json file, change line 10 to match the IP address on line 15.
    b) Save changes.
5) Verify changes
    a) Place SW2.6 to the OFF position (towards pin 6).
    b) Power cycle the DCA1000EVM. (This loads the Ethernet settings from the DCA1000EVM's EEPROM).
    c) Open Powershell.
    d) Use the following commands to enter the correct directory and verify system status:

```
cd "C:\ti\mmwave_studio_xx_xx_xx_xx\mmWaveStudio\PostProc\"
.\DCA1000EVM_CLI_Control.exe query_sys_status newIP.json
```

    e) If response is "System is connected" then the device is functioning properly. If the response is "System is disconnected" then ensure that the following are true:
        i SW2.6 is in the OFF position (towards pin 6).
        ii PC's IP Address is set to the correct IP Address for the system and not the IP Address for the DCA.
        iii Correct .json file is called when running `query_sys_status` command.

# CHAPTER 7

# USING HARDWARE TRIGGER WITH TI 6843 60 GHZ RADAR

The following is a simple demo for enabling the hardware (HW) trigger with a Texas Instruments IWR6843ISK 60 GHz radar in conjunction with the Dual Radar SAR Controller application.

## 7.1 Hardware Requirements

1) DCA1000EVM with modifications (see Section 7.3.1).
2) MMWAVEICBOOST with modifications (see Section 7.3.2).
3) IWR6843ISK with modifications (see Section 7.3.3).
4) ESP32 microcontroller to send HW trigger pulses.

## 7.2 Software Requirements

1) TI mmWave SDK 3.5.0.4
2) TI Uniflash
3) TI mmWave Studio 2.1.1.0
4) Dual Radar SAR Controller software (requires MATLAB 2021b (recommended, may be compatible with earlier/later releases)).

## 7.3 Hardware Set Up

### 7.3.1 DCA1000EVM Hardware Configuration

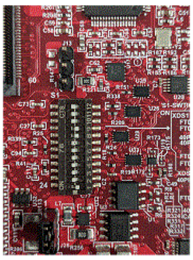
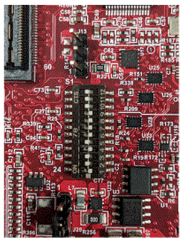
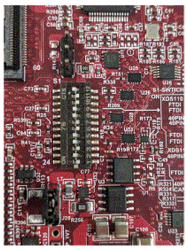
1) Remove R120 (suggested here).
2) Connect to the MMWAVEICBOOST with 60 pin connector, as shown on page 16 here.
3) Connect to the PC over USB on the RADAR/FTDI connnector (J1 on DCA1000EVM) and over Ethernet.
4) Connect 5V/3A power.

### 7.3.2 MMWAVEICBOOST Hardware Configuration

1) Remove R346 and short R348 (suggested here and here).
    a) Follow Josiah's E2E post:
    b) Download xWR6843 EVM Schematic Drawing, Assembly Drawing, and Bill of Materials - SWRR164C.zip from here.
    c) On page 9 of PROC074B(001)_Sch.pdf (for rev B of the MMWAVEICBOOST), under "RNR FOR SYNC IN", 40PIN_SYNC_IN needs to be routed to RADAR_SYNC_IN.
    d) From that diagram, DCA_SYNC_IN is shorted via R346 to RADAR_SYNC_IN.
    e) Hence, remove R346 and place 0 ohm resistor over R348. Now 40PIN_SYNC_IN is routed to RADAR_SYNC_IN.
2) Switch Settings
    a) From page 11 here, use the switch settings on S1 for DCA1000 Mode, as shown in Fig. 7.1.
3) Connect to Microcontroller (MCU)
    a) Follow Josiah's E2E post:
    b) On page 8 of PROC074B(001)_Sch.pdf and page 19 of here, 40PIN_SYNC_IN is pin 9 of J5 **(IMPORTANT)**. See picture here for pin 9 of J5 input. Ground is pin 4 of J5 or pin 2 of J6.
4) Connect to the PC over USB on XDS110_USB/J1 and attach IWR6843 as shown on page 16 here.
5) Connect 5V/3A power.

### 7.3.3 IWR6843ISK Hardware Configuration

1) Switch Settings
    a) From step 1 here or page 45 here, use the switch settings on S1, as shown in Fig. 7.2.
2) Connect to MMWAVEICEBOOST, as shown on page 16 here.

| Reference Designator | (Default Position) Position for STAND ALONE Mode (1) | Position for DCA1000 Mode | Position for 40-Pin LP/BP |
|---|---|---|---|
| |  |  |  |
| S1.12 | ON | ON | ON |
| S1.11 | ON | ON | OFF |
| S1.10 | ON | ON | OFF |
| S1.9 | OFF | OFF | ON |
| S1.8 | OFF | OFF | ON |
| S1.7 | ON | OFF | OFF |
| S1.6 | ON | OFF | OFF |
| S1.5(2) | ON | ON | OFF |
| S1.4 | ON | OFF | ON |
| S1.3 | ON | ON | OFF |
| S1.2 | ON | ON | ON |
| S1.1 | OFF | OFF | OFF |

(1) Standalone mode means starter kit and MMWAVEICBOOST connected together.
(2) S1.5 has RS232 connections from 40 pin/FTDI/60 pin/XDS110, ON postion routes UART to XDS110 (Application/user UART COM port).

Figure 7.1: Switch Settings for MMWAVEICBOOST

## 7.4 Software Set Up

### 7.4.1 Flash the SDK Demo Binary to the MMWAVEICBOOST (See TI SDK 3.5.0.4 User Guide Section 4.2)

1) Set the device to Flash Programming Mode by bridging SOP0 and SOP2 as shown in step 5 here.

2) Power cycle the MMWAVEICBOOST.

3) Once the device is properly connected to the PC, download the demo firmware using Unifash. (Typically under the path: "C:/ti/mmwave_sdk_03_05_00_04/packages/ti/demo/xwr68xx/mmw").

4) Once the download is complete, set the device to Functional Mode by bridging only SOP0 and remove the bridge on SOP2.

5) Power cycle the MMWAVEICBOOST.

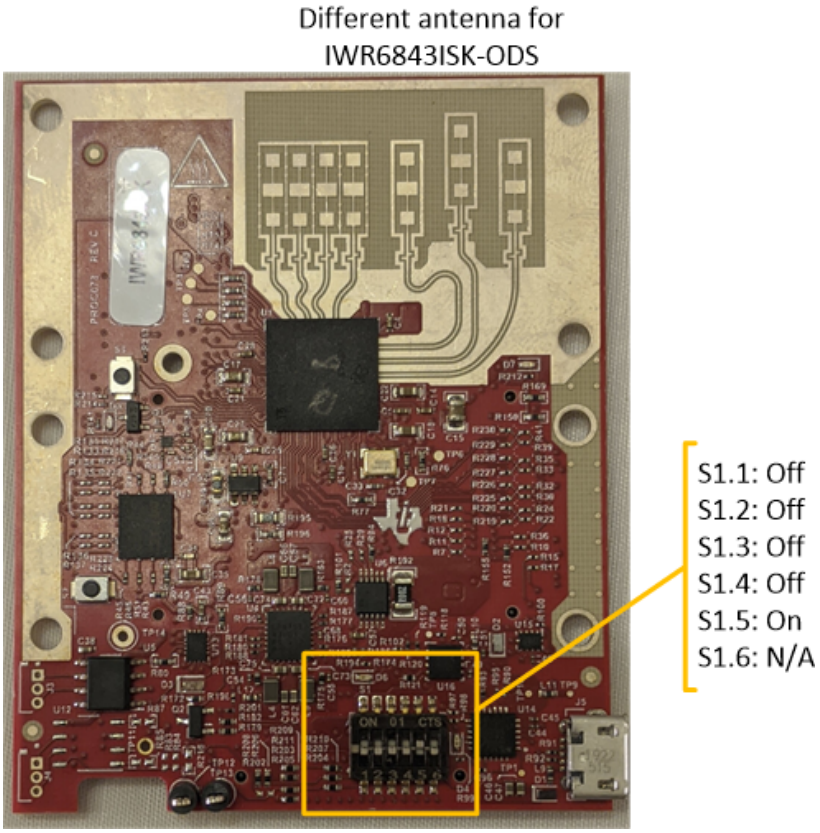



Figure 7.2: Switch Settings for IWR6843ISK

### 7.4.2 Set Up the DCA1000EVM on Proper IP Address (See the DCA1000EVM Quick Start Guide)

1) Once the device is properly connected to the PC, open the start menu and search "View Network Connections," as shown in Fig. 7.3.

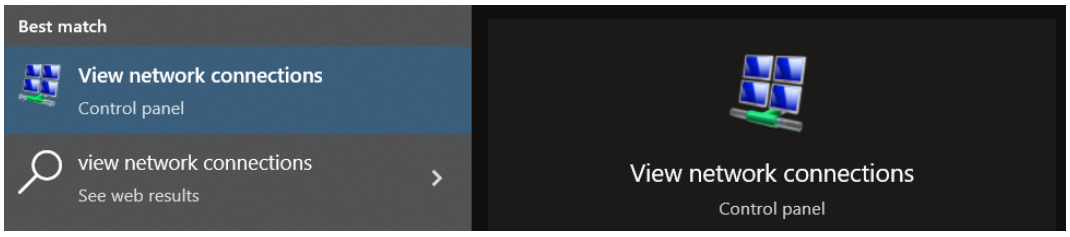



Figure 7.3: Example of Windows start menu search result for viewing network connections

2) Inside the "Network Connections" of Control Panel, right click on the Ethernet port of choice and select "Properties."

3) "Local Area Connection Properties" window will open. Right click on "Internet Protocol Version 4 (TCP/IPv4)."

4) "Internet Protocol Version 4 (TCP/IPc4)" window will open. Set the IP address field to 192.168.33.30, or the desired IP address if different. (See Chapter 6 on changing the IP address of the DCA1000EVM using the DCA1000 CLI Utility, which is necessary for a dual radar setup).

5) The Subnet mask field can remain the default 255.255.255.0.

6) Press "OK" on all the windows and you can close "Network Connections."

### 7.4.3 Open the Dual Radar SAR Controller MATLAB Application

1) Open MATLAB and navigate to the folder containing "dual_radar_gui.mlapp" and open that file. The MATLAB App Designer window will open the application.

2) Press "Run" at the top of the page.

3) The app will open and all the indicators will be red as none of the devices are connected to the application or configured.

4) Assuming you have installed mmWave Studio 2.1.1.0 to the typical location, it will open normally. Otherwise, it will ask you to find the installation location of mmWave Studio 2.1.1.0.

5) Press "Connect Radar 1."

6) A window will appear asking to select a serial COM port. Select the COM port corresponding to the entry in device manager labeled "XDS110 Class Application/User UART."

7) Under the DCA 1 Configuration panel, enter the correct System and DCA IP addresses and ports for the radar. Press "Prepare DCA 1." If a Windows Defender Firewall window opens, give access to private networks and public networks (check both boxes) and select "Allow access." (The optional drop down menu in the bottom right corner of the DCA 1 Configuration panel is used to select from a set of defaults with different patterns and is straightforward).

8) Under the Radar 1 (60 GHz) Configuration panel, enter the desired chirp parameters and press "Configure Radar 1." The Radar 1 Configuration lamp should change to a green color to indicate the configuration was successful. If the configuration stalls and the lamp is stuck on yellow, this indicates a communication error between the PC and radar, and the radar must be disconnected and power cycled. Then, attempt these steps again.

9) Under the Radar 1 (60 GHz) Configuration panel, press "Start" to start the 60 GHz radar.

    a) The DCA1000EVM will start waiting for data over LVDS.

    b) The MMWAVEICBOOST and IWR6843ISK will wait for HW trigger from MCU.

    c) If everything is working properly at this point, the D7 LED on the IWR6843ISK and the DS2 LED on the MMWAVEICBOOST will turn on.

10) Start the MCU sending pulses. If everything is working properly the DATA_TRAIN_PRG LED on the DCA1000EVM will be flashing while the radar is triggered.

11) Once the MCU stops sending pulses, the radar and DCA1000EVM will automatically stop the capture. (Pressing the "stop" button is only necessary if you set No of Frames to 0, for infinite capture mode, with HW trigger disabled, indicating the SW trigger will be used).

# CHAPTER 8

# USING HARDWARE TRIGGER WITH TI 1642 77 GHZ RADAR

The following is a simple demo for enabling the hardware (HW) trigger with a Texas Instruments xWR1642BOOST 77 GHz radar in conjunction with the Dual Radar SAR Controller application.

## 8.1 Hardware Requirements

1) DCA1000EVM with modifications (see Section 8.3.1).

3) xWR1642BOOST with modifications (see Section 8.3.2).

4) ESP32 microcontroller to send HW trigger pulses.

## 8.2 Software Requirements

1) TI mmWave SDK 3.5.0.4

2) TI Uniflash

3) TI mmWave Studio 2.1.1.0

4) Dual Radar SAR Controller software (requires MATLAB 2021b (recommended, may be compatible with earlier/later releases)).

## 8.3 Hardware Set Up

### 8.3.1 DCA1000EVM Hardware Configuration

1) Remove R120 (suggested here).

2) Connect to the MMWAVEICBOOST with 60 pin connector, as shown on page 16 here.

3) Connect to the PC over USB on the RADAR/FTDI connnector (J1 on DCA1000EVM) and over Ethernet.

4) Connect 5V/3A power.

### 8.3.2 xWR1642BOOST Hardware Configuration

1) Connect to Microcontroller (MCU)

   a) Follow Josiah's E2E post:

   b) For IWR1642Boost, on page 10 of PROC049B(002)_Sch.pdf, AR_SYNC_IN is pin 9 of J6 **(IMPORTANT)**.

   c) For AWR1642Boost, On page 10 of PROC011C(002)_Sch.pdf, AR_SYNC_IN is pin 9 of J6 **(IMPORTANT)**.

   d) For newest revisions, the R165 resistor is already shorted, enabling the HW trigger input.

   e) Ground is pin 4 of J6 or pin 2 of J5.

2) Connect to the PC over.

3) Connect 5V/3A power.

## 8.4 Software Set Up

### 8.4.1 Flash the SDK Demo Binary to the xWR1642BOOST (See TI SDK 3.5.0.4 User Guide Section 4.2)

1) Set the device to Flash Programming Mode by bridging SOP0 and SOP2. Similar bridging is shown on the MMWAVEICBOOST in step 5 here.

2) Power cycle the xWR1642BOOST.

3) Once the device is properly connected to the PC, download the demo firmware using Unifash. (Typically under the path: "C:/ti/mmwave_sdk_03_05_00_04/packages/ti/demo/xwr68xx/mmw").

4) Once the download is complete, set the device to Functional Mode by bridging only SOP0 and remove the bridge on SOP2.

5) Power cycle the xWR1642BOOST.

### 8.4.2 Set Up the DCA1000EVM on Proper IP Address (See the DCA1000EVM Quick Start Guide)

1) Once the device is properly connected to the PC, open the start menu and search "View Network Connections," as shown in Fig. 7.3.

2) Inside the "Network Connections" of Control Panel, right click on the Ethernet port of choice and select "Properties."

3) "Local Area Connection Properties" window will open. Right click on "Internet Protocol Version 4 (TCP/IPv4)."

4) "Internet Protocol Version 4 (TCP/IPc4)" window will open. Set the IP address field to 192.168.33.30, or the desired IP address if different. (See Chapter 6 on changing the IP address of the DCA1000EVM using the DCA1000 CLI Utility, which is necessary for a dual radar setup).

5) The Subnet mask field can remain the default 255.255.255.0.

6) Press "OK" on all the windows and you can close "Network Connections."

### 8.4.3 Open the Dual Radar SAR Controller MATLAB Application

1) Open MATLAB and navigate to the folder containing "dual_radar_gui.mlapp" and open that file. The MATLAB App Designer window will open the application.

2) Press "Run" at the top of the page.

3) The app will open and all the indicators will be red as none of the devices are connected to the application or configured.

4) Assuming you have installed mmWave Studio 2.1.1.0 to the typical location, it will open normally. Otherwise, it will ask you to find the installation location of mmWave Studio 2.1.1.0.

5) Press "Connect Radar 2."

6) A window will appear asking to select a serial COM port. Select the COM port corresponding to the entry in device manager labeled "XDS110 Class Application/User UART."

7) Under the DCA 2 Configuration panel, enter the correct System and DCA IP addresses and ports for the radar. Press "Prepare DCA 2." If a Windows Defender Firewall window opens, give access to private networks and public networks (check both boxes) and select "Allow access." (The optional drop down menu in the bottom right corner of the DCA 2 Configuration panel is used to select from a set of defaults with different patterns and is straightforward).

8) Under the Radar 2 (77 GHz) Configuration panel, enter the desired chirp parameters and press "Configure Radar 2." The Radar 2 Configuration lamp should change to a green color to indicate the configuration was successful. If the configuration stalls and the lamp is stuck on yellow, this indicates a communication error between the PC and radar, and the radar must be disconnected and power cycled. Then, attempt these steps again.

9) Under the Radar 2 (77 GHz) Configuration panel, press "Start" to start the 77 GHz radar.

    a) The DCA1000EVM will start waiting for data over LVDS.

    b) The MMWAVEICBOOST and IWR6843ISK will wait for HW trigger from MCU.

    c) If everything is working properly at this point, the D7 LED on the IWR6843ISK and the DS2 LED on the MMWAVEICBOOST will turn on.

10) Start the MCU sending pulses. If everything is working properly the DATA_TRAIN_PRG LED on the DCA1000EVM will be flashing while the radar is triggered.

11) Once the MCU stops sending pulses, the radar and DCA1000EVM will automatically stop the capture. (Pressing the "stop" button is only necessary if you set No of Frames to 0, for infinite capture mode, with HW trigger disabled, indicating the SW trigger will be used).

# APPENDIX A

# PRELIMINARIES OF FMCW SIGNALING

## A.1 Monostatic FMCW Signal Model

### A.1.1 FMCW Chirp

In this section we derive the simple signal model for a monostatic (Tx and Rx are assumed to be at the same position in space) scenario with a single, ideal point scatterer (reflector). By definition, an FMCW signal, called an FMCW chirp, has a frequency linearly increasing with time. We can express this relationship as

$$f(t) \triangleq f_0 + Kt, \quad 0 \leq t \leq T, \tag{A.1}$$

where $f$ is the instantaneous frequency as a function of $t$, $f_0$ is the start frequency (frequency at time $t = 0$), $K$ is the slope of the chirp, the bandwidth covered by the chirp is given by $B = KT$, and $t$ is called the "fast time" variable.

The definition of instantaneous frequency is given by

$$f(t) \triangleq \frac{1}{2\pi}\frac{\partial}{\partial t}\phi(t), \tag{A.2}$$

where $\phi(t)$ is the phase of the signal, containing the frequency content, e.g.,

$$m(t) \triangleq e^{j\phi(t)}. \tag{A.3}$$

Hence, the phase term $\phi(t)$ can be expressed as

$$\phi(t) = 2\pi \int_0^t f(t')dt'. \tag{A.4}$$

Substituting (A.1) into (A.4) yields

$$\begin{aligned} \phi(t) &= 2\pi \int_0^t (f_0 + Kt')dt', \\ &= 2\pi \left[ f_0 t' + 0.5K(t')^2 \right]_0^t, \\ &= 2\pi(f_0 t + 0.5Kt^2). \end{aligned} \tag{A.5}$$

### A.1.2 FMCW Beat Signal

Thus, by the definition of the FMCW chirp as a sinusoidal signal whose frequency linearly increases with time, we can derive the phase of the signal and express the transmitted FMCW pulse by substituting (A.5) into (A.3) as

$$m(t) = e^{j\phi(t)} = e^{j2\pi(f_0 t + 0.5Kt^2)}, \quad 0 \leq t \leq T, \tag{A.6}$$

where $T$ is the duration of the FMCW pulse.

Assuming the monostatic radar antenna element is located at $(x', y', z')$ and the point scatterer is located at $(x_0, y_0, z_0)$, the distance between the radar and point target can be expressed as

$$R_0 = \sqrt{(x_0 - x')^2 + (y_0 - y')^2 + (z_0 - z')^2}. \tag{A.7}$$

The FMCW pulse, expressed in (A.6) is transmitted from the antenna, propagates through space traveling a distance of $R_0$ to the point scatter, reflects from the point scatterer, travels another $R_0$ back to the radar. The round-trip time delay required for this propagation is given by

$$\tau_0 = \frac{2R_0}{c}, \tag{A.8}$$

where $c$ is the speed of light.

As a result, the signal received at the radar is a time-delayed and scaled version of the transmitted signal as

$$\begin{aligned}
\hat{s}(t) &= \frac{\sigma}{R_0^2} m(t - \tau_0), \\
&= \frac{\sigma}{R_0^2} e^{j2\pi(f_0(t-\tau_0) + 0.5K(t-\tau_0)^2)}, \\
&= \frac{\sigma}{R_0^2} e^{j2\pi(f_0 t - f_0\tau_0 + 0.5Kt^2 - K\tau_0 t + 0.5K\tau_0^2)}, \\
&= \frac{\sigma}{R_0^2} e^{j2\pi(f_0 t + 0.5Kt^2 - f_0\tau_0 - K\tau_0 t + 0.5K\tau_0^2)}, \\
&= \frac{\sigma}{R_0^2} \underbrace{e^{j2\pi(f_0 t + 0.5Kt^2)}}_{m(t)} e^{-j2\pi(f_0\tau_0 + K\tau_0 t - 0.5K\tau_0^2)},
\end{aligned} \tag{A.9}$$

where $\sigma$ is known as the "reflectivity" of the scatterer (how reflective the point target is) and the $1/R_0^2$ term is the round-trip path loss or amplitude decay (the intuition here is that farther targets give weaker reflections). Note that the received signal $\hat{s}(t)$ contains a factor which is the transmitted signal $m(t)$.

The next step in the signal chain is known as "dechirping" and removes this factor of $m(t)$ by multiplying the conjugate of the received signal by the transmitted signal. After dechirping, the signal is known as the IF signal or beat signal. The beat signal can be expressed as

$$\begin{aligned} s(t) &= m(t)\hat{s}^*(t), \\ &= e^{j2\pi(f_0 t+0.5Kt^2)}\frac{\sigma}{R_0^2}e^{-j2\pi(f_0 t+0.5Kt^2)}e^{j2\pi(f_0\tau_0+K\tau_0 t-0.5K\tau_0^2)}, \\ &= \frac{\sigma}{R_0^2}e^{j2\pi(f_0 t+0.5Kt^2)-j2\pi(f_0 t+0.5Kt^2)}e^{j2\pi(f_0\tau_0+K\tau_0 t-0.5K\tau_0^2)}, \\ &= \frac{\sigma}{R_0^2}e^{j2\pi(f_0\tau_0+K\tau_0 t-0.5K\tau_0^2)}. \end{aligned} \tag{A.10}$$

In radar literature, it is common practice to express the beat signal as a function of $k$ rather than $t$, where $k(t)$ is the instantaneous wavenumber corresponding to the instantaneous frequency $f(t)$ given by

$$k(t) \triangleq \frac{2\pi}{c}f(t). \tag{A.11}$$

Substituting (A.1) into (A.11) yields

$$k(t) = \frac{2\pi}{c}(f_0 + Kt), \quad 0 \le t \le T. \tag{A.12}$$

Hence, let us express $s(t)$ as a function of $k$ by rewriting (A.10) in terms of (A.12). Note that for short distances $\tau_0^2$ is negligible and the last term in (A.10) can be ignored. Substituting (A.8) into (A.10) and recalling the definitions in (A.1) and (A.11) yields

$$\begin{aligned} s(t) &= \frac{\sigma}{R_0^2}e^{j2\pi(f_0\tau_0+K\tau_0 t)}, \\ &= \frac{\sigma}{R_0^2}e^{j2\pi\tau_0(f_0+Kt)}, \\ &= \frac{\sigma}{R_0^2}e^{j2\pi\frac{2R_0}{c}(f_0+Kt)}, \\ &= \frac{\sigma}{R_0^2}e^{j2\pi\frac{2R_0}{c}f(t)}, \\ &= \frac{\sigma}{R_0^2}e^{j2R_0\frac{2\pi}{c}f(t)}, \\ s(k) &= \frac{\sigma}{R_0^2}e^{j2R_0 k}, \quad 0 \le t \le T. \end{aligned} \tag{A.13}$$

More commonly, the beat signal is expressed as

$$s(k) = \frac{\sigma}{R_0^2}e^{j2kR_0} \tag{A.14}$$

We have derived a compact representation of the FMCW beat signal, $s(k)$. It is clear from (A.14) that the frequency of the beat signal corresponds directly with the radial distance, known as the "range", $R_0$. For this single point scatter case, the radar beat signal $s(k)$ is a single tone sinusoid whose frequency corresponds with $R_0$. Hence, the Fourier transform of $s(k)$ would have a single peak at a position corresponding to the range $R_0$.

This section detailed the derivation of the radar beat signal. However, now that the beat signal has been derived, it can be applied simply by the definition of $s(k)$ in (A.14) without needing to go through all steps every time.

### A.1.3 Multiple Targets

Suppose $N$ point scatterers are in the radar FOV such that the $n$-th point scatterer is located at $(x_n, y_n, z_n)$ and has reflectivity $\sigma_n$. In this case, the transmit signal is the same, but the received signal is now a sum (by superposition) of the received signals from each of the point scatterers. As a result, the radar beat signal can be written as

$$s(k) = \sum_{n=1}^{N} \frac{\sigma_n}{R_n^2} e^{j2kR_n}, \quad 0 \leq t \leq T, \tag{A.15}$$

which is clearly a sum of sinusoidal signals whose frequencies depend on the distances $R_n$. Hence, the Fourier transform of (A.15) would result in multiple peaks such that the location of the $n$-th peak corresponds with the distance $R_n$.

Alternatively, if we model the target as a continuous set of point targets, rather than a discrete set as in (A.15), where the target is located in a volume $V$ inside $(x, y, z)$ space, we express the reflectivity as a continuous function $p(x, y, z)$ and the summation expressed in (A.15) becomes an integral as

$$s(k) = \iiint_V \frac{p(x, y, z)}{R^2} e^{j2kR} dx dy dz, \tag{A.16}$$

where

$$R = \sqrt{(x - x')^2 + (y - y')^2 + (z - z')^2}, \tag{A.17}$$

recall the position of the antenna is $(x', y', z')$.

## A.2 Range Resolution and Maximum Resolvable Range

Knowing the target range information is present in the frequency of the beat signal (A.14), we examine the minimum resolvable distance between two targets in the scene and the maximum resolvable range.

The simplest method of identifying the frequency content of a given signal is the Fourier transform. According to Fourier transform theory, frequency components can be resolved if separated by at least the reciprocal of the observation time ($T$) as

$$\Delta f > \frac{1}{T}, \tag{A.18}$$

where $\Delta f$ is the change in frequency of the beat signal

Considering a monostatic radar with two targets separated by a distance $\Delta R$, the difference in frequency between the two beat signals, $\Delta f$, is expressed as

$$\Delta f = \frac{2K\Delta R}{c}. \tag{A.19}$$

Combining (A.18) with (A.19) yields

$$\frac{2K\Delta R}{c} > \frac{1}{T} \Rightarrow \Delta R > \frac{c}{2KT}, \tag{A.20}$$

$$\Delta R > \frac{c}{2B}. \tag{A.21}$$

This minimum resolvable distance is commonly known as the radar range resolution for ultra-wide-band (UWB) systems. For an automotive radar with a several GHz of bandwidth, the range resolution will be on the order of centimeters. For example, a common 4 GHz chirp yields a range resolution of 3.75 cm.

Similar analysis can be performed to compute the maximum resolvable range of a given set of chirp parameters. The maximum range $R_{max}$ yields an IF frequency of $f_{max} = 2KR_{max}/c$. Assuming a complex baseband IF signal, the maximum frequency is limited by the ADC sampling rate $f_S$ as

$$f_S > \frac{2KR_{max}}{c}. \tag{A.22}$$

The expression above can be rearranged yielding the maximum range as a function of the chirp slope, the sampling frequency, and the speed of light as

$$R_{max} < \frac{f_S c}{2K}. \tag{A.23}$$

# APPENDIX B

## SPATIAL FOURIER TRANSFORM AND RELATIONS

Neglecting amplitude terms, the 1-D, 2-D and 3-D spatial Fourier transforms can be defined as [4]

$$\mathrm{FT}_{1\mathrm{D}}^{(u)}\left[s(u)\right] = S(k_u) = \int s(u)e^{-jk_u u}du, \tag{B.1}$$

$$\mathrm{FT}_{2\mathrm{D}}^{(u,v)}\left[s(u,v)\right] = S(k_u,k_v) = \iint s(u,v)e^{-j(k_u u+k_v v)}dudv, \tag{B.2}$$

$$\mathrm{FT}_{3\mathrm{D}}^{(u,v,w)}\left[s(u,v,w)\right] = S(k_u,k_v,k_w) = \iiint s(u,v,w)e^{-j(k_u u+k_v v+k_w w)}dudvdw. \tag{B.3}$$

Similarly, the 1-D, 2-D and 3-D inverse spatial Fourier transforms can be expressed as

$$\mathrm{IFT}_{1\mathrm{D}}^{(k_u)}\left[S(k_u)\right] = s(u) = \int S(k_u)e^{jk_u u}du, \tag{B.4}$$

$$\mathrm{IFT}_{2\mathrm{D}}^{(k_u,k_v)}\left[S(k_u,k_v)\right] = s(u,v) = \iint S(k_u,k_v)e^{j(k_u u+k_v v)}dudv, \tag{B.5}$$

$$\mathrm{IFT}_{3\mathrm{D}}^{(k_u,k_v,k_w)}\left[S(k_u,k_v,k_w)\right] = s(u,v,w) = \iiint S(k_u,k_v,k_w)e^{j(k_u u+k_v v+k_w w)}dudvdw. \tag{B.6}$$

A shift in the spatial domain results in a corresponding phase shift in the spatial spectral domain. The example given here is in the 3-D spatial domain but holds true for the 2-D and 1-D cases also:

$$\mathrm{FT}_{3\mathrm{D}}^{(u,v,w)}\left[s(u-u_0,v-v_0,w-w_0)\right] = e^{-j(k_u u_0+k_v v_0+k_w w_0)}S(k_u,k_v,k_w). \tag{B.7}$$

Similarly, a shift in the spatial spectral domain results in a phase shift in the spatial domain:

$$\mathrm{IFT}_{3\mathrm{D}}^{(k_u,k_v,k_w)}\left[S(k_u-k_0^u,k_v-k_0^v,k_w-k_0^w)\right] = e^{j(k_u u_0+k_v v_0+k_w w_0)}s(u,v,w). \tag{B.8}$$

These spatial Fourier transform definitions and relations are useful in deriving the reconstruction algorithms discussed in the subsequent appendices.

# APPENDIX C

# METHOD OF STATIONARY PHASE

In this chapter, we discuss the important topic of the Method of Stationary Phase (MSP) required for many of the subsequent imaging algorithms detailed in following chapters. A highly recommended exercise to the reader is to follow the example derivation in Section C.3.1 closely and derive the helpful approximations given in (C.22)–(C.27) showing every step. Additionally, a careful review of the spatial Fourier relationships in Appendix B is recommended.

As discussed in [6, 7, 8], the general form of the $n$-dimensional Method of Stationary Phase (MSP) can be expressed as the following. A rigorous mathematical perspective is offered in [8], whereas our discussion does not comprehensively address the underlying assumptions and constraints. Rather, this section is meant to serve as a resource to researchers and engineers to apply the results of the MSP approximation to near-field spherical wave decomposition problems.

Given an oscillatory integral with a wide phase variation of the form

$$I(\mathbf{x}) = \int g(\mathbf{x}) e^{jf(\mathbf{x})} d\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n, \tag{C.1}$$

where $f(\mathbf{x})$ is assumed to be twice-continuously differentiable, the major contribution to the quantity $I(\mathbf{x})$ is from the stationary points, $\mathbf{x}_0$, which are calculated by

$$\nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} = 0 \tag{C.2}$$

Thus, the integral can be approximated by

$$I(\mathbf{x}) \approx \frac{g(\mathbf{x}_0)}{\sqrt{\det \mathbf{A}}} e^{jf(\mathbf{x}_0)}, \tag{C.3}$$

where $\mathbf{x}_0$ is the set of stationary points and $\mathbf{A}$ is the Hessian matrix of $f(\mathbf{x})$ evaluated at $\mathbf{x}_0$ and defined as

$$\mathbf{A} = \left( \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right) \Bigg|_{\mathbf{x}=\mathbf{x}_0}. \tag{C.4}$$

For the derivations required in this article, we can limit $n$ to 1 or 2 dimensions.

## C.1 1-D Method of Stationary Phase

The 1-D MSP can be written as the following. The following integral with the same assumptions as the MSP,

$$I(u) = \int g(u)e^{jf(u)}du, \tag{C.5}$$

can be approximated as

$$I(u) \approx \frac{g(u_0)}{\sqrt{f''(u_0)}}e^{jf(u_0)}, \tag{C.6}$$

where $u_0$ is the stationary point calculated by

$$\left.\frac{\partial f(u)}{\partial u}\right|_{u=u_0} = 0, \tag{C.7}$$

and $f''(u_0)$ is the second derivative of $f(u)$ evaluated at the stationary point $u_0$.

## C.2 2-D Method of Stationary Phase

Similarly, for the 2-D case, the integral,

$$I(u,v) = \iint g(u,v)e^{jf(u,v)}dudv, \tag{C.8}$$

can be approximated by

$$I(u,v) \approx \frac{g(u_0,v_0)}{\sqrt{f_{uu}f_{vv} - f_{uv}^2}}e^{jf(u_0,v_0)}, \tag{C.9}$$

where the stationary points $u_0$, $v_0$ are calculated by

$$\left.\frac{\partial f(u,v)}{\partial u}\right|_{(u=u_0,v=v_0)} = 0, \tag{C.10}$$

$$\left.\frac{\partial f(u,v)}{\partial v}\right|_{(u=u_0,v=v_0)} = 0, \tag{C.11}$$

and $f_{uu}$, $f_{vv}$, $f_{uv}$ are the second partial derivatives of $f(u,v)$ evaluated at the stationary points.

## C.3 Useful MSP Identities

Using the aforementioned method for the 1-D and 2-D cases, the MSP is applied to several integrals and the corresponding approximations are provided for reference in this section.

We will demonstrate the steps for the approximation below which have been applied to the other spherical wavefronts to yield the relations in (C.22)–(C.27).

### C.3.1 Example MSP Derivation

We consider the linear array case with a monostatic single antenna array being scanned along the $x$-axis at the positions labeled $x'$. Further, we consider a 1-D target at some line $z_0$ in the $x$-$z$ plane, where the $x$ and $x'$ coordinate systems are coincident. Thus, the radar beat signal can be modeled, neglecting path loss, as

$$s(x', k) = \int p(x) e^{j2kR} dx, \tag{C.12}$$

where $R$ is the radial distance from each of the antenna locations $(x', 0)$ to the target locations $(x, z)$ and is expressed as

$$R = \sqrt{(x - x')^2 + z_0^2}. \tag{C.13}$$

It is desired to approximate the spherical wavefront term in (C.12), $e^{j2kR}$, as a more tractable expression. Thus, the MSP is exploited. For generality, the following substitutions are made $u = x'$, $r = 2k$, $w = z_0$. The 1-D spatial Fourier transform (B.1) is performed over the $u$ dimension of the spherical wave term and the spatial translation property (B.7) is applied as

$$\mathrm{FT}_{1\mathrm{D}}^{(u)} \left[ e^{jr\sqrt{(x-u)^2+w^2}} \right] = e^{-jk_u x} \int e^{jr\sqrt{u^2+w^2} - jk_u u} du. \tag{C.14}$$

The MSP will be applied to the Fourier integral in (C.14), implying for this example

$$g(u) = 1, \tag{C.15}$$

$$f(u) = r\sqrt{u^2 + w^2} - k_u u. \tag{C.16}$$

Using (C.7), the stationary point $u_0$ can be computed as

$$\left. \frac{\partial f(u)}{\partial u} \right|_{u=u_0} = \frac{r u_0}{\sqrt{u_0^2 + w^2}} - k_u = 0, \tag{C.17}$$

$$u_0 = \frac{k_u w}{\sqrt{r^2 - k_u^2}}, \tag{C.18}$$

$$f(u_0) = w\sqrt{r^2 - k_u^2} \tag{C.19}$$

Finally, $u_0$ can be substituted into (C.6) ignoring the factor of $1/f''(u_0)$ as

$$\int e^{jr\sqrt{u^2+w^2} - jk_u u} du \approx e^{jw\sqrt{r^2-k_u^2}}. \tag{C.20}$$

Substituting (C.20) into (C.14) yields

$$\mathrm{FT}_{1\mathrm{D}}^{(u)}\left[e^{jr\sqrt{(x-u)^2+w^2}}\right] = e^{-jk_u x + jw\sqrt{r^2-k_u^2}}. \tag{C.21}$$

Taking the 1-D inverse spatial Fourier transform of (C.21) results in (C.22), labeled Approximation 1 below. This example illustrates the key steps of the spherical wave decomposition using the method of stationary phase. Similar analysis has been employed on the other examples below yielding the corresponding approximations using the MSP.

### C.3.2 Useful MSP Approximations

**Approximation 1:**

$$e^{jr\sqrt{(x-u)^2+w^2}} \approx \int e^{jk_u(u-x)+jk_w w} dk_u, \tag{C.22}$$

where

$$k_w^2 = r^2 - k_u^2. \tag{C.23}$$

**Approximation 2:**

$$\frac{e^{jr\sqrt{(x-u)^2+(y-v)^2+w^2}}}{\sqrt{(x-u)^2+(y-v)^2+w^2}} \approx \iint \frac{1}{k_w} e^{jk_u(u-x)+jk_v(v-y)+jk_w w} dk_u dk_v, \tag{C.24}$$

where

$$k_w^2 = r^2 - k_u^2 - k_v^2. \tag{C.25}$$

**Approximation 3:**

$$e^{jr\sqrt{(x-u)^2+(z-w)^2}} \approx \iint e^{jk_u(u-x)+jk_w(w-z)} dk_u dk_w. \tag{C.26}$$

**Approximation 4:**

$$e^{jr\sqrt{(x-u)^2+(y-v)^2+(z-w)^2}} \approx \iiint e^{jk_u(u-x)+jk_v(v-y)+jk_w(w-z)} dk_u dk_v dk_w. \tag{C.27}$$

# APPENDIX D

# EFFICIENT NEAR-FIELD SAR IMAGE RECONSTRUCTION ALGORITHMS FOR VARIOUS GEOMETRIES

In this chapter, we detail the efficient image reconstruction algorithms for several synthetic aperture radar (SAR) scanning geometries. The algorithms in this section have been derived elsewhere and are included for the benefit of the reader.

## D.1 1-D Linear Synthetic Array 1-D Imaging - Fourier-based

In this section, we derive the image reconstruction algorithm for recovering a 1-D reflectivity function from a 1-D linear SAR scenario in the near-field [9, 10, 11, 12, 13]. Given a 1-D linear SISO synthetic array whose elements are located at the points $(y', Z_0)$ in the $y$-$z$ plane and a 1-D target with reflectivity function $p(y)$ located at the points $(y, z_0)$, the isotropic beat signal can be written as

$$s(y', k) = \int \frac{p(y)}{R^2} e^{j2kR} dy, \tag{D.1}$$

where

$$R = \sqrt{(y - y')^2 + (z_0 - Z_0)^2}. \tag{D.2}$$

Ignoring amplitude terms, applying the MSP derived in (C.22), the spherical phase term in (D.1) can be substituted yielding

$$s(y', k) = \iint p(y) e^{j(k'_y(y'-y)+k_z(z_0-Z_0))} dy dk'_y, \tag{D.3}$$

where

$$k_z = \sqrt{4k^2 - k_y^2}. \tag{D.4}$$

Rearranging the phase terms in (D.3), a forward spatial Fourier transform on $y$ and inverse spatial Fourier transform on $y'$ become evident as

$$s(y', k) = \int \left[ \int p(y) e^{-jk'_y y} dy \right] e^{j(k'_y y' + k_z(z_0 - Z_0))} dk'_y. \tag{D.5}$$

The term inside the brackets can be rewritten as the spatial-spectral representation of the target reflectivity function, $P(k_y)$. Then, performing a forward Fourier transform along $y'$ on both sides simplifies the expression as the following. Note that the distinction between

the primed and unprimed domains can be dropped in the spatial Fourier domain as they coincide.

$$s(y',k) = \int \left[P(k_y)e^{jk_z(z_0-Z_0)}\right] e^{jk'_y y'} dk'_y, \tag{D.6}$$

$$S(k_y,k) = P(k_y)e^{jk_z(z_0-Z_0)}, \tag{D.7}$$

$$P(k_y) = S(k_y,k)e^{-jk_z(z_0-Z_0)}. \tag{D.8}$$

For wideband waveforms, (D.8) is evaluated at multiple wavenumbers thus coherent summation is performed over $k$. Hence, the complete expression for the Fourier-based 1-D image reconstruction algorithm for a 1-D linear SISO synthetic array is

$$p(y) = \int \mathrm{IFT}_{1\mathrm{D}}^{(k_y)} \left[\mathrm{FT}_{1\mathrm{D}}^{(y')}[s(y',k)]e^{-jk_z(z_0-Z_0)}\right] dk. \tag{D.9}$$

## D.2 1-D Linear Synthetic Array 2-D Imaging - Range Migration Algorithm

In this section we derive the image reconstruction algorithm for recovering a 2-D reflectivity function from a 1-D linear SAR scenario in the near-field [9, 10, 11, 12, 13]. Given a 1-D linear SISO synthetic array whose elements are located at the points $(y', Z_0)$ in the $y$-$z$ plane and a 2-D target with reflectivity function $p(y,z)$ located at the points $(y,z)$, the isotropic beat signal can be written as

$$s(y',k) = \iint \frac{p(y,z)}{R^2} e^{j2kR} dydz, \tag{D.10}$$

where

$$R = \sqrt{(y-y')^2 + (z-Z_0)^2}. \tag{D.11}$$

Ignoring amplitude terms, applying the MSP derived in (C.22), the spherical phase term in (D.10) can be substituted yielding

$$s(y',k) = \iiint p(y,z)e^{j(k'_y(y'-y)+k_z(z-Z_0))} dydzdk'_y, \tag{D.12}$$

where

$$k_z = \sqrt{4k^2 - k_y^2}. \tag{D.13}$$

Leveraging conjugate symmetry of the spherical wavefront, (D.12) can be rewritten in the following form to exploit the spatial Fourier transform on $z$

$$s^*(y',k) = \iiint p(y,z)e^{j(k'_y(y'-y)-k_z(z-Z_0))} dydzdk'_y, \tag{D.14}$$

where $(\cdot)^*$ is the complex conjugate operation.

Rearranging the phase terms in (D.14), a forward spatial Fourier transform on $y$-$z$ and inverse spatial Fourier transform on $y'$ become evident as

$$s^*(y',k) = \int \left[ \iint p(y,z) e^{-j(k_y' y + k_z z)} dy dz \right] e^{j(k_y' y' + k_z Z_0)} dk_y'. \tag{D.15}$$

The term inside the brackets can be rewritten as the spatial-spectral representation of the target reflectivity function, $P(k_y, k_z)$. Then, performing a forward Fourier transform along $y'$ on both sides simplifies the expression as the following. Note that the distinction between the primed and unprimed domains can be dropped in the spatial Fourier domain as they coincide.

$$s^*(y',k) = \int \left[ P(k_y,k_z) e^{jk_z Z_0} \right] e^{jk_y' y'} dk_y', \tag{D.16}$$

$$\tilde{S}(k_y,k) = \mathrm{FT}_{1\mathrm{D}}^{(y')}[s^*(y',k)], \tag{D.17}$$

$$\tilde{S}(k_y,k) = P(k_y,k_z) e^{jk_z Z_0}, \tag{D.18}$$

$$P(k_y,k_z) = \tilde{S}(k_y,k) e^{-jk_z Z_0}. \tag{D.19}$$

The direct relationship between $P(k_y, k_z)$ and $\tilde{S}(k_y, k)$ is now obvious in (D.19); however, $P(k_y, k_z)$ is sampled on a uniform $k_y$-$k_z$ grid and $\tilde{S}(k_y, k)$ is sampled on a uniform $k_y$-$k$ grid. Before the reflectivity function can be recovered using in inverse Fourier transform, $\tilde{S}(k_y, k)e^{-jk_z Z_0}$ must be interpolated to a uniform $k_y$-$k_z$ grid using Stolt interpolation, represented by the $\mathcal{S}[\cdot]$ operator, to account for the curvature of the wavefront [14].

$$S(k_y,k_z) = \mathcal{S}\left[ \tilde{S}(k_y,k) e^{-jk_z Z_0} \right]. \tag{D.20}$$

Finally, the complete expression for the Fourier-based 2-D image reconstruction algorithm for a 1-D linear SISO synthetic array can be written as

$$p(y,z) = \mathrm{IFT}_{2\mathrm{D}}^{(k_y,k_z)} \left[ \mathcal{S}\left[ \mathrm{FT}_{1\mathrm{D}}^{(y')}[s^*(y',k)] e^{-jk_z Z_0} \right] \right]. \tag{D.21}$$

### D.3 2-D Rectilinear Array 2-D Imaging - Fourier-based

In this section, we derive the image reconstruction algorithm for recovering a 2-D reflectivity function from a 2-D rectilinear SAR scenario in the near-field [15, 16]. Given a 2-D rectilinear SISO synthetic array whose elements are located at the points $(x', y', Z_0)$ in $x$-$y$-$z$ space and

a 2-D target with reflectivity function $p(x, y)$ located at the points $(x, y, z_0)$, the isotropic beat signal can be written as

$$s(x', y', k) = \iint \frac{p(x, y)}{R^2} e^{j2kR} dxdy, \tag{D.22}$$

where

$$R = \sqrt{(x - x')^2 + (y - y')^2 + (z_0 - Z_0)^2}. \tag{D.23}$$

Assuming the points of the target scene are closely located, the $R^{-2}$ factor in (D.22) can be approximated as $R^{-1}$ [4]. Applying the MSP derived in (C.24), the spherical phase term in (D.22) can be substituted yielding

$$s(x', y', k) = \iiiint \frac{p(x, y)}{k_z} e^{j(k'_x(x'-x)+k'_y(y'-y))} e^{jk_z(z_0-Z_0)} dxdydk'_x dk'_y, \tag{D.24}$$

where

$$k_z = \sqrt{4k^2 - k_x^2 - k_y^2}. \tag{D.25}$$

Rearranging the phase terms in (D.24), a forward spatial Fourier transform on $x$-$y$ and inverse spatial Fourier transform on $x'$-$y'$ become evident as

$$s(x'y', k) = \iint \left[ \iint \frac{p(x, y)}{k_z} e^{-j(k'_x x + k'_y y)} dxdy \right] e^{j(k'_x x' + k'_y y') + jk_z(z_0 - Z_0)} dk'_x dk'_y. \tag{D.26}$$

The term inside the brackets can be rewritten as the spatial-spectral representation of the target reflectivity function. Then, performing a forward Fourier transform along $x'$-$y'$ on both sides simplifies the expression as the following. Note that the distinction between the primed and unprimed domains can be dropped in the spatial Fourier domain as they coincide.

$$s(x', y', k) = \int \left[ \frac{P(k_x, k_y)}{k_z} e^{jk_z(z_0 - Z_0)} \right] e^{j(k'_x x' + k'_y y')} dk'_x dk'_y, \tag{D.27}$$

writing

$$S(k_x, k_y, k) = \frac{P(k_x, k_y)}{k_z} e^{jk_z(z_0 - Z_0)}, \tag{D.28}$$

$$P(k_x, k_y) = S(k_y, k) k_z e^{-jk_z(z_0 - Z_0)}. \tag{D.29}$$

For wideband waveforms, (D.29) is evaluated at multiple wavenumbers thus coherent summation is performed over $k$. Hence, the complete expression for the Fourier-based 2-D image reconstruction algorithm for a 2-D rectilinear SISO synthetic array is

$$p(x, y) = \int \mathrm{IFT}_{\mathrm{2D}}^{(k_x, k_y)} \left[ \mathrm{FT}_{\mathrm{2D}}^{(x', y')} [s(x', y', k)] k_z e^{-jk_z(z_0 - Z_0)} \right] dk. \tag{D.30}$$

### D.4 2-D Rectilinear Array 3-D Imaging - Range Migration Algorithm

In this section we derive the image reconstruction algorithm for recovering a 3-D reflectivity function from a 2-D rectilinear SAR scenario in the near-field [14, 17, 18, 19, 20, 21, 22, 23, 5, 4, 1, 24, 25, 26, 27, 28]. Given a 2-D rectilinear SISO synthetic array whose elements are located at the points $(x', y', Z_0)$ in $x$-$y$-$z$ space and a 3-D target with reflectivity function $p(x, y, z)$ located at the points $(x, y, z)$, the isotropic beat signal can be written as

$$s(x', y', k) = \iiint \frac{p(x, y, z)}{R^2} e^{j2kR} dxdydz, \tag{D.31}$$

where

$$R = \sqrt{(x - x')^2 + (y - y')^2 + (z - Z_0)^2}. \tag{D.32}$$

Assuming the points of the target scene are closely located, the $R^{-2}$ factor in (D.31) can be approximated as $R^{-1}$ [4]. Applying the MSP derived in (C.24), the spherical phase term in (D.31) can be substituted yielding

$$s(x', y', k) = \iint \left[ \iiint \frac{p(x, y, z)}{k_z} e^{j(k'_x(x'-x)+k'_y(y'-y))} e^{jk_z(z-Z_0)} dxdydz \right] dk'_x dk'_y, \tag{D.33}$$

where

$$k_z = \sqrt{4k^2 - k_x^2 - k_y^2}. \tag{D.34}$$

Leveraging conjugate symmetry of the spherical wavefront, (D.33) can be rewritten in the following form to exploit the spatial Fourier transform on $z$

$$s^*(x', y', k) = \iint \left[ \iiint \frac{p(x, y, z)}{k_z} e^{j(k'_x(x'-x)+k'_y(y'-y))} e^{-jk_z(z-Z_0)} dxdydz \right] dk'_x dk'_y, \tag{D.35}$$

where $(\cdot)^*$ is the complex conjugate operation.

Rearranging the phase terms in (D.35), a forward spatial Fourier transform on $x$,$y$,$z$ and inverse spatial Fourier transform on $x'$,$y'$ become evident as

$$s^*(x', y', k) = \iint \left[ \iiint \frac{p(x, y, z)}{k_z} e^{-(jk'_x x + jk'_y y + jk_z z)} dxdydz \right] e^{j(k'_x x' + k'_y y') + jk_z Z_0} dk'_x dk'_y. \tag{D.36}$$

The term inside the brackets can be rewritten as the spatial-spectral representation of the target reflectivity function. Then, performing a forward Fourier transform along $x'$, $y'$ on

both sides simplifies the expression as the following. Note that the distinction between the primed and unprimed domains can be dropped in the spatial Fourier domain as they coincide.

$$s^*(x', y', k) = \int \left[P(k_x, k_y, k_z)e^{jk_z Z_0}\right] e^{j(k'_x x' + k'_y y')} dk'_x dk'_y, \tag{D.37}$$

writing

$$\tilde{S}(k_x, k_y, k) = \mathrm{FT}_{\mathrm{2D}}^{(x',y')}[s^*(x', y', k)], \tag{D.38}$$

$$\tilde{S}(k_x, k_y, k) = \frac{P(k_x, k_y, k_z)}{k_z} e^{jk_z Z_0}, \tag{D.39}$$

$$P(k_x, k_y, k_z) = \tilde{S}(k_x, k_y, k) k_z e^{-jk_z Z_0}. \tag{D.40}$$

The direct relationship between $P(k_x, k_y, k_z)$ and $\tilde{S}(k_x, k_y, k)$ is now obvious in (D.40); however, $P(k_x, k_y, k_z)$ is sampled on a uniform $k_x$-$k_y$-$k_z$ grid and $\tilde{S}(k_x, k_y, k)$ is sampled on a uniform $k_x$-$k_y$-$k$ grid. Before the reflectivity function can be recovered using an inverse Fourier transform, $\tilde{S}(k_x, k_y, k)e^{-jk_z Z_0}$ must be interpolated to a uniform $k_x, k_y$-$k_z$ grid using the Stolt interpolation, represented by the $\mathcal{S}[\cdot]$ operator, to account for the curvature of the wavefront [14].

$$S(k_x, k_y, k_z) = \mathcal{S}\left[\tilde{S}(k_x, k_y, k) k_z e^{-jk_z Z_0}\right]. \tag{D.41}$$

Finally, the complete expression for the Fourier-based 3-D image reconstruction algorithm for a 2-D rectilinear SISO synthetic array can be written as

$$p(x, y, z) = \mathrm{IFT}_{\mathrm{3D}}^{(k_x,k_y,k_z)}\left[\mathcal{S}\left[\mathrm{FT}_{\mathrm{2D}}^{(x',y')}[s^*(x', y', k)] k_z e^{-jk_z Z_0}\right]\right]. \tag{D.42}$$

### D.5 1-D Circular Synthetic Array 2-D Imaging - Polar Formatting Algorithm

In this section, we derive the image reconstruction algorithm for recovering a 2-D reflectivity function from a 1-D circular SAR scenario in the near-field [29, 30, 31]. Given a 1-D circular SISO synthetic array whose elements are located at the points $(R_0 \cos\theta, R_0 \sin\theta)$ in the $x$-$z$ plane at $y = 0$, where $R_0$ and $\theta$ are the constant radial distance from the antenna elements to the origin and the angular dimension, respectively, and a 2-D target with reflectivity function $p(x, z)$ located at the points $(x, z)$, the isotropic beat signal can be written as

$$s(\theta, k) = \iint \frac{p(x, z)}{R^2} e^{j2kR} dx dz, \tag{D.43}$$

where

$$R = \sqrt{(x - R_0 \cos\theta)^2 + (z - R_0 \sin\theta)^2}. \tag{D.44}$$

The MSP derived in (C.26) can be applied to the spherical phase term in (D.43) after the following substitutions

$$x' = R_0 \cos\theta, \tag{D.45}$$

$$z' = R_0 \cos\theta, \tag{D.46}$$

$$k'_x = k_r \cos\alpha, \tag{D.47}$$

$$k'_z = k_r \cos\alpha, \tag{D.48}$$

$$k_r^2 = k_x'^2 + k_z'^2, \tag{D.49}$$

yielding

$$e^{j2kR} \approx \iint e^{j(k'_x(x'-x)+k'_z(z'-z))} dk'_x dk'_z. \tag{D.50}$$

Neglecting path loss, (D.43) and (D.50) can be combined as

$$s(\theta, k) = \iiiint p(x, z) e^{j(k'_x(x'-x)+k'_z(z'-z))} dx dz dk'_x dk'_z. \tag{D.51}$$

Rearranging the phase terms in (D.51), a forward spatial Fourier transform on $x$-$z$ and inverse spatial Fourier transform on $x'$-$z'$ become evident as

$$s(\theta, k) = \iint \left[ \iint p(x, z) e^{-j(k'_x x + k'_z z)} dx dz \right] e^{j(k'_x x' + k'_z z')} dk'_x dk'_z. \tag{D.52}$$

The term inside the brackets can be rewritten as the spatial spectral representation of the target reflectivity function, $P(k_x, k_z)$. Then using the relations (D.45)-(D.49), the expression in (D.52) can be rewritten as

$$s(\theta, k) = \iint P(k_x, k_z) e^{j(k_r \cos\theta R_0 \cos\alpha + k_r \sin\theta R_0 \sin\alpha)} k_r dk_r d\alpha. \tag{D.53}$$

Rewriting the spectral $P(k_x, k_z)$ as its equivalent spectral polar form $P(\alpha, k_r)$ and simplifying the phase term

$$s(\theta, k) = \int \left[ \int P(\alpha, k_r) e^{jk_r R_0 cos(\theta - \alpha)} d\alpha \right] k_r dk_r. \tag{D.54}$$

The term inside the brackets in (D.54) is a convolution operation in the $\theta$ domain, where the $\theta$ and $\alpha$ domains are coincident and can be exploited using Fourier relations by taking a Fourier transform across $\theta$ on both sides of the equation as

$$S(k_\theta, k) = \int P(k_\theta, k_r) \mathrm{FT}_{1\mathrm{D}}^{(\theta)} \left[ e^{jk_r R_0 \cos\theta} \right] k_r dk_r. \tag{D.55}$$

Considering only the values lying on the Ewald sphere, $k_r^2 = 4k^2$ imposes a $\delta$-function behavior of the integrand in (D.55) with respect to $k_r$ [32]. As such, (D.55) can be simplified as such, substituting $k_r = 2k$,

$$P(k_\theta, k_r) = S(k_\theta, k) G^*(k_\theta, k), \tag{D.56}$$

where

$$G(k_\theta, k) = \mathrm{FT}_{1\mathrm{D}}^{(\theta)} \left[ e^{j2kR_0 \cos\theta} \right]. \tag{D.57}$$

The spatial spectral reflectivity function in polar coordinates can be recovered from (D.56) as

$$P(\theta, k_r) = \mathrm{IFT}_{1\mathrm{D}}^{(k_\theta)} \left[ S(k_\theta, k) G^*(k_\theta, k) \right]. \tag{D.58}$$

Finally, the reflectivity function $p(x, z)$ can be recovered using a nonuniform FFT (NUFFT) [33] or via interpolation to the rectangular spatial Fourier domain $k_x$-$k_z$ followed by a uniform IFFT. This interpolation operation, known as the polar formatting algorithm (PFA), is denoted by $\mathcal{P}[\cdot]$. Thus, the final step in the image recovery process is (prime notation will be ignored for the remainder of this derivation as the primed and unprimed coordinate systems are coincident)

$$p(x, z) = \mathrm{IFT}_{2\mathrm{D}}^{(k_x, k_z)} \left[ \mathcal{P}[P(\theta, k_r)] \right]. \tag{D.59}$$

Finally, the complete expression for the Fourier-based 2-D image reconstruction algorithm for a 1-D circular SISO synthetic array can be written as

$$p(x, z) = \mathrm{IFT}_{2\mathrm{D}}^{(k_x, k_z)} \left[ \mathcal{P} \left[ \mathrm{IFT}_{1\mathrm{D}}^{(k_\theta)} \left[ S(k_\theta, k) \mathrm{FT}_{1\mathrm{D}}^{(\theta)} \left[ e^{j2kR_0 \cos\theta} \right]^* \right] \right] \right]. \tag{D.60}$$

## D.6 2-D Cylindrical Synthetic Array 3-D Imaging - Polar Formatting Algorithm

In this section, we derive the image reconstruction algorithm for recovering a 3-D reflectivity function from a 2-D cylindrical SAR (also known as ECSAR) scenario in the near-field [32, 34, 35, 36, 37, 38]. Given a 2-D cylindrical SISO synthetic array whose elements are located

at the points $(R_0 \cos\theta, y', R_0 \sin\theta)$ in $x$-$y$-$z$ space, where $R_0$ and $\theta$ are the constant radial distance from the antenna elements to the origin and the angular dimension, respectively, and a 3-D target with reflectivity function $p(x, y, z)$ located at the points $(x, y, z)$, the isotropic beat signal can be written as

$$s(\theta, y', k) = \iiint \frac{p(x, y, z)}{R^2} e^{j2kR} dx dy dz, \tag{D.61}$$

where

$$R = \sqrt{(x - R_0 \cos\theta)^2 + (y - y')^2 + (z - R_0 \sin\theta)^2}. \tag{D.62}$$

The MSP derived in (C.27) can be applied to the spherical phase term in (D.61) after the following substitutions

$$x' = R_0 \cos\theta, \tag{D.63}$$

$$z' = R_0 \cos\theta, \tag{D.64}$$

$$k'_x = k_r \cos\alpha, \tag{D.65}$$

$$k'_z = k_r \cos\alpha, \tag{D.66}$$

$$k_r^2 = k_x'^2 + k_z'^2 = 4k^2 - k_y'^2, \tag{D.67}$$

yielding

$$e^{j2kR} \approx \iiint e^{j(k'_x(x'-x)+k'_y(y'-y)+k'_z(z'-z))} dk'_x dk'_y dk'_z. \tag{D.68}$$

Neglecting path loss, (D.61) and (D.68) can be combined as

$$s(\theta, y', k) = \iiint \left[ \iiint p(x, y, z) e^{j(k'_x(x'-x)+k'_y(y'-y)+k'_z(z'-z))} dx dy dz \right] dk'_x dk'_y dk'_z. \tag{D.69}$$

Rearranging the phase terms in (D.69), a forward spatial Fourier transform on $x$-$y$-$z$ and inverse spatial Fourier transform on $x'$-$y'$-$z'$ become evident as

$$s(\theta, y', k) = \iiint \left[ \iiint p(x, y, z) e^{-j(k'_x x + k'_y y + k'_z z)} dx dy dz \right] e^{j(k'_x x' + k'_y y' + k'_z z')} dk'_x dk'_y dk'_z. \tag{D.70}$$

The term inside the brackets can be rewritten as the spatial spectral representation of the target reflectivity function, $P(k_x, k_y, k_z)$. Then using the relations (D.63)-(D.67), the expression in (D.70) can be rewritten as

$$s(\theta, y', k) = \iiint P(k_x, k_y, k_z) e^{j(k_r \cos\theta R_0 \cos\alpha} e^{j(k_r \sin\theta R_0 \sin\alpha) + k'_y y')} k_r dk'_y dk_r d\alpha. \tag{D.71}$$

Taking a Fourier transform on both side with respect to $y'$, rewriting the spectral $P(k_x, k_y, k_z)$ as its equivalent spectral polar form $P(\alpha, k_y, k_r)$, and simplifying the phase term yields (prime notation will be dropped for the remainder of this derivation as the primed and unprimed coordinate systems are coincident)

$$s(\theta, k_y, k) = \int \left[ \int P(\alpha, k_y, k_r) e^{jk_r R_0 cos(\theta - \alpha)} d\alpha \right] k_r dk_r. \tag{D.72}$$

The term inside the brackets in (D.72) is a convolution operation in the $\theta$ domain, where the $\theta$ and $\alpha$ domains are coincident and can be exploited using Fourier relations by taking a Fourier transform across $\theta$ on both sides of the equation as

$$S(k_\theta, k_y, k) = \int P(k_\theta, k_y, k_r) \mathrm{FT}_{1\mathrm{D}}^{(\theta)} \left[ e^{jk_r R_0 \cos\theta} \right] k_r dk_r. \tag{D.73}$$

Considering only the values lying on the Ewald sphere, $k_r^2 = 4k^2 - k_y^2$ imposes a $\delta$-function behavior of the integrand in (D.73) with respect to $k_r$ [32]. As such, (D.73) can be simplified as such, substituting $k_r = \sqrt{4k^2 - k_y^2}$,

$$P(k_\theta, k_y, k_r) = S(k_\theta, k_y, k) G^*(k_\theta, k_y, k), \tag{D.74}$$

where

$$G(k_\theta, k_y, k) = \mathrm{FT}_{1\mathrm{D}}^{(\theta)} \left[ e^{j\sqrt{4k^2 - k_y^2} R_0 \cos\theta} \right]. \tag{D.75}$$

The spatial spectral reflectivity function in polar coordinates can be recovered from (D.74) as

$$P(\theta, k_y, k_r) = \mathrm{IFT}_{1\mathrm{D}}^{(k_\theta)} \left[ S(k_\theta, k_y, k) G^*(k_\theta, k_y, k) \right]. \tag{D.76}$$

Finally, the reflectivity function $p(x, y, z)$ can be recovered using a nonuniform FFT (NUFFT) [33] or via interpolation to the rectangular spatial Fourier domain $k_x$-$k_y$-$k_z$ followed by a uniform IFFT. This interpolation operation, known as the polar formatting algorithm (PFA), is denoted by $\mathcal{P}[\cdot]$. Thus, the final step in the image recovery process is

$$p(x, y, z) = \mathrm{IFT}_{3\mathrm{D}}^{(k_x, k_y, k_z)} \left[ \mathcal{P}[P(\theta, k_y, k_r)] \right]. \tag{D.77}$$

Finally, the complete expression for the Fourier-based 3-D image reconstruction algorithm for a 2-D cylindrical SISO synthetic array can be written as

$$p(x, y, z) = \mathrm{IFT}_{3\mathrm{D}}^{(k_x, k_y, k_z)} \left[ \mathcal{P} \left[ \mathrm{IFT}_{1\mathrm{D}}^{(k_\theta)} \left[ S(k_\theta, k_y, k) \mathrm{FT}_{1\mathrm{D}}^{(\theta)} \left[ e^{j\sqrt{4k^2 - k_y^2} R_0 \cos\theta} \right]^* \right] \right] \right]. \tag{D.78}$$

## REFERENCES


[1] M. E. Yanik, D. Wang, and M. Torlak, "Development and demonstration of MIMO-SAR mmWave imaging testbeds," *IEEE Access*, vol. 8, pp. 126 019–126 038, Jul. 2020.

[2] Z. Li, S. Papson, and R. M. Narayanan, "Data-level fusion of multilook inverse synthetic aperture radar images," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 5, pp. 1394–1406, Apr. 2008.

[3] J. Wang, P. Aubry, and A. Yarovoy, "Wavenumber-domain multiband signal fusion with matrix-pencil approach for high-resolution imaging," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 4037–4049, Apr. 2018.

[4] M. E. Yanik and M. Torlak, "Near-field MIMO-SAR millimeter-wave imaging with sparsely sampled aperture data," *IEEE Access*, vol. 7, pp. 31 801–31 819, Mar. 2019.

[5] M. E. Yanik, D. Wang, and M. Torlak, "3-D MIMO-SAR imaging using multi-chip cascaded millimeter-wave sensors," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Ottawa, ON, Canada, Nov. 2019, pp. 1–5.

[6] C. Cook, *Radar signals: an introduction to theory and application*. Elsevier, 2012.

[7] A. Papoulis, "Systems and transforms with applications in optics," *McGraw-Hill Series in System Science*, 1968.

[8] J. McClure and R. Wong, "Multidimensional stationary phase approximation: boundary stationary point," *J. Comput. Appl. Mathematics*, vol. 30, no. 2, pp. 213–225, May 1990.

[9] S. Paul, F. Schwartau, M. Krueckemeier, R. Caspary, C. Monka-Ewe, J. Schoebel, and W. Kowalsky, "A systematic comparison of near-field beamforming and Fourier-based backward-wave holographic imaging," *IEEE Open J. Antennas Propag.*, vol. 2, pp. 921–931, Aug. 2021.

[10] M. A. Maisto, R. Pierri, and R. Solimene, "Sensor arrangement in monostatic subsurface radar imaging," *IEEE Open J. Antennas Propag.*, vol. 2, pp. 3–13, Nov. 2020.

[11] M. Soumekh, "Wide-bandwidth continuous-wave monostatic/bistatic synthetic aperture radar imaging," in *Proc. IEEE Int. Conf. Image Process. (ICIP98)*, Chicago, IL, USA, Oct. 1998, pp. 361–365.

[12] ——, *Synthetic aperture radar signal processing*. New York: Wiley, 1999, vol. 7.

[13] J. W. Smith, O. Furxhi, and M. Torlak, "An FCNN-based super-resolution mmWave radar framework for contactless musical instrument interface," *IEEE Trans. Multimedia*, pp. 1–1, May 2021.

[14] J. M. Lopez-Sanchez and J. Fortuny-Guasch, "3-D radar imaging using range migration techniques," *IEEE Trans. Antennas Propag.*, vol. 48, no. 5, pp. 728–737, May 2000.

[15] Q. Guo, J. Liang, T. Chang, and H.-L. Cui, "Millimeter-wave imaging with accelerated super-resolution range migration algorithm," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 11, pp. 4610–4621, Jul. 2019.

[16] M. E. Yanik and M. Torlak, "Millimeter-wave near-field imaging with two-dimensional SAR data," in *Proc. SRC Techcon*, no. P093929, Austin, TX, USA, Sep. 2018.

[17] X. Zhuge and A. G. Yarovoy, "A sparse aperture MIMO-SAR-based UWB imaging system for concealed weapon detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 1, pp. 509–518, Jul. 2010.

[18] T. Savelyev, X. Zhuge, B. Yang, P. Aubry, A. Yarovoy, L. Ligthart, and B. Levitas, "Comparison of 10-18 GHz SAR and MIMO-based short-range imaging radars," *Int. J. Microw. Wireless Techn.*, vol. 2, no. 3–4, p. 369, Aug. 2010.

[19] B. Fan, J. Gao, H. Li, Z. Jiang, and Y. He, "Near-field 3D SAR imaging using a scanning linear MIMO array with arbitrary topologies," *IEEE Access*, vol. 8, pp. 6782–6791, Dec. 2019.

[20] N. Mohammadian, O. Furxhi, R. Short, and R. Driggers, "SAR millimeter wave imaging systems," in *Proc. SPIE*, vol. 10994, Baltimore, MD, USA, May 2019, p. 109940A.

[21] X. Zhuge and A. G. Yarovoy, "Three-dimensional near-field MIMO array imaging using range migration techniques," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3026–3033, Feb. 2012.

[22] R. Zhu, J. Zhou, G. Jiang, and Q. Fu, "Range migration algorithm for near-field MIMO-SAR imaging," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2280–2284, Nov. 2017.

[23] L. Qiao, Y. Wang, Z. Shen, Z. Zhao, and Z. Chen, "Compressive sensing for direct millimeter-wave holographic imaging," *Appl. Opt.*, vol. 54, no. 11, pp. 3280–3289, Apr. 2015.

[24] J. W. Smith, S. Thiagarajan, R. Willis, Y. Makris, and M. Torlak, "Improved static hand gesture classification on deep convolutional neural networks using novel sterile training technique," *IEEE Access*, vol. 9, pp. 10 893–10 902, Jan. 2021.

[25] D. M. Sheen, T. E. Hall, D. L. McMakin, A. M. Jones, and J. R. Tedeschi, "Three-dimensional radar imaging techniques and systems for near-field applications," in *Proc. SPIE*, vol. 9829, Baltimore, MD, USA, May 2016, p. 98290V.

[26] D. Sheen, D. McMakin, and T. Hall, "Near-field three-dimensional radar imaging techniques and applications," *Appl. Opt.*, vol. 49, no. 19, pp. E83–E93, Jun. 2010.

[27] D. M. Sheen, D. L. McMakin, and T. E. Hall, "Three-dimensional millimeter-wave imaging for concealed weapon detection," *IEEE Trans. Microw. Theory Techn.*, vol. 49, no. 9, pp. 1581–1592, Sep. 2001.

[28] D. M. Sheen, A. M. Jones, and T. E. Hall, "Simulation of active cylindrical and planar millimeter-wave imaging systems," in *Proc. SPIE*, vol. 10634, Orlando, FL, USA, May 2018, p. 1063408.

[29] S. Demirci, H. Cetinkaya, M. Tekbas, E. Yigit, C. Ozdemir, and A. Vertiy, "Back-projection algorithm for ISAR imaging of near-field concealed objects," in *Proc. XXXth URSI Gen. Assem. Sci. Symp. (URSI GASS)*, Istanbul, Turkey, Aug. 2011, pp. 1–4.

[30] G. Jia and W. Chang, "Modified back projection reconstruction for circular FMCW SAR," in *Proc. Inter. Radar Conf.*, Lille, France, Oct. 2014, pp. 1–5.

[31] J. K. Gao, Y. L. Qin, B. Deng, H. Q. Wang, J. Li, and X. Li, "Terahertz wide-angle imaging and analysis on plane-wave criteria based on inverse synthetic aperture techniques," *J. Infrared Millim. Terahertz Waves*, vol. 37, no. 4, pp. 373–393, Jan. 2016.

[32] R. K. Amineh, N. K. Nikolova, and M. Ravan, *Real-Time Three-Dimensional Imaging of Dielectric Bodies Using Microwave/Millimeter Wave Holography.* John Wiley & Sons, 2019.

[33] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "Efficient terahertz wide-angle NUFFT-based inverse synthetic aperture imaging considering spherical wavefront," *Sensors*, vol. 16, no. 12, p. 2120, Dec. 2016.

[34] J. Fortuny-Guasch and J. M. Lopez-Sanchez, "Extension of the 3-D range migration algorithm to cylindrical and spherical scanning geometries," *IEEE Trans. Antennas Propag.*, vol. 49, no. 10, pp. 1434–1444, Oct. 2001.

[35] J. Detlefsen, A. Dallinger, S. Huber, S. Schelkshorn, and F. H. F. und Schaltungen, "Effective reconstruction approaches to millimeter-wave imaging of humans," in *Proc. XXVIIIth URSI Gen. Assem. Sci. Symp. (URSI GASS)*, New Delhi, India, Oct. 2005, pp. 23–29.

[36] J. Laviada, A. Arboleya-Arboleya, Y. Álvarez, B. González-Valdés, and F. Las-Heras, "Multiview three-dimensional reconstruction by millimetre-wave portable camera," *Sci. Rep.*, vol. 7, no. 1, pp. 6479–6479, Jul. 2017.

[37] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "An efficient algorithm for MIMO cylindrical millimeter-wave holographic 3-D imaging," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 11, pp. 5065–5074, Aug. 2018.

[38] J. W. Smith, M. E. Yanik, and M. Torlak, "Near-field MIMO-ISAR millimeter-wave imaging," in *Proc. IEEE Radar Conf. (RadarConf)*, Florance, Italy, Sep. 2020, pp. 1–6.

## BIOGRAPHICAL SKETCH

Josiah W. Smith received the BSEE degree (*summa cum laude*) in electrical engineering from The University of Texas at Dallas in 2019, where he is currently pursuing a PhD degree in electrical engineering specializing in communications engineering.

In 2019, he was an undergraduate research intern at the Texas Analog Center of Excellence (TxACE) working on radar reception for hand gesture recognition and high resolution near-field MIMO synthetic aperture radar imaging algorithms. During the summer of 2020, he developed real-time human-computer interaction algorithms for mmWave radar with imec-USA. In the summer of 2021, he developed advanced deep learning and data-driven algorithms for user experience enhancement at Apple in the Display Technologies group.

Mr. Smith was awarded the Texas Instruments Analog Excellence Graduate Fellowship in August 2019 and the Louis Beecherl, Jr. Graduate Research Fellowship in August 2021. In 2020, he was awarded first alternate for the student paper competition in the *IEEE Radar Conf. 2020*, ranking 6/143 and best poster award at the *2020 TxACE Symposium*. He is a student IEEE member. His current research interests include new regime radar imaging algorithm development, ultrawideband radar imaging algorithms, terahertz radar, radar perception, computer vision, machine learning, millimeter-wave sensing, and phased array signal processing.

**CURRICULUM VITAE**

# Josiah W. Smith

## Contact Information:

Department of Electrical and Computer Eng.
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

## Educational History:

BSEE, Electrical Engineering, The University of Texas at Dallas, 2019

## Employment History:

Research Assistant, The University of Texas at Dallas, August 2019 – May 2022
Machine Learning Intern, Apple - EE Display Team, March 2021 – August 2021
Computational Imaging Intern, imec-USA, March 2020 – August 2020
Undergraduate Researcher, Texas Analog Center of Excellence (TxACE), January 2019 – August 2019

## Publications :

[J1] **J. W. Smith**, S. Thiagarajan, R. Willis, Y. Makris and M. Torlak, "Improved Static Hand Gesture Classification on Deep Convolutional Neural Networks Using Novel Sterile Training Technique," *IEEE Access*, vol. 9, pp. 10893–10902, Jan. 2021.
[J2] **J. W. Smith**, O. Furxhi, M. Torlak, "An FCNN-Based Super-Resolution mmWave Radar Framework for Contactless Musical Instrument Interface," *IEEE Trans. on Multimedia*, May 2021.
[J3] **J. W. Smith** and M. Torlak, "Efficient 3-D Near-Field MIMO-SAR Imaging for Irregular Scanning Geometries," *IEEE Access*, vol. 10, pp. 10283–10294. Jan. 2022.
[J4] **J. W. Smith** and M. Torlak, "Deep learning-based multiband signal fusion for 3-D SAR super-resolution," *IEEE Trans. Image Process.*, submitted for publication.
[J5] **J. W. Smith** and M. Torlak, "Survey of emerging systems and algorithms for near-field THz SAR imaging," *IEEE Trans. THz Sci. Technol.*, to be submitted.
[C1] **J. W. Smith**, M. E. Yanik and M. Torlak, "Near-Field MIMO-ISAR Millimeter-Wave Imaging," in *Proc. IEEE Radar Conf. (RadarConf)*, Florence, Italy, Sep. 2020, pp. 1–6.
[C2] **J. W. Smith**, Y. Alimam, G. Vedula, and M. Torlak, "A vision transformer approach for efficient near-field SAR super-resolution under array perturbation," in *Proc. IEEE Tex. Symp. Wirel. Microw. Circuits Syst. (WMCS)*, Waco, TX, USA, Apr. 2022.

[C3] C. Vasieleiou, **J. W. Smith**, S. Thiagarajan, M. Nigh, Y. Makris, and M. Torlak, "Efficient CNN-based super resolution algorithms for mmWave mobile radar imaging," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, submitted for publication.

## Professional Recognitions and Honors:

Louis Beecherl, Jr. Graduate Research Fellowship, Aug. 2021
Best Poster Award, TxACE Symposium, Oct. 2020
First Alternate Student Paper (Ranked 6/143), IEEE Radar Conference 2020, Sep. 2020
Texas Instruments Analog Excellence Graduate Fellowship, Aug. 2019
First Place Senior Capstone Project, UTDesign II Expo, May 2019
First Place Senior Capstone Project, UTDesign I Expo, Dec. 2018
Greater Texas Foundation Removing Educational Barriers Endowed Scholarship, Aug. 2017
Academic Excellence Full-Ride Scholarship, The University of Texas at Dallas, Aug. 2016

## Professional Memberships:

Institute of Electrical and Electronics Engineers (IEEE), 2016 – present