Design Guide: TIDEP-01034

Entry-level blind spot detection reference design using mmWave radar

Description

This reference design provides a foundation for corner radar applications to meet entry-level blind spot detection (BSD) requirements using the AWRL1432BOOST-BSD evaluation module. The design allows users to estimate and track the position (in the azimuthal plane) and velocity of objects up to 120m.

Resources

TIDEP-01034	Design Folder
AWRL1432	Product Folder
MMWAVE-L-SDK	Product Folder
TCAN4550-Q1	Product Folder

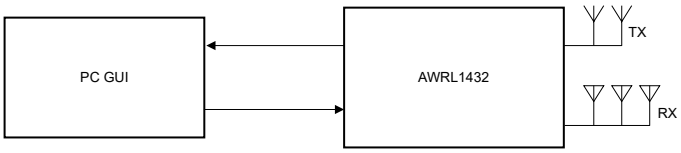
Ask our TI E2E™ support experts

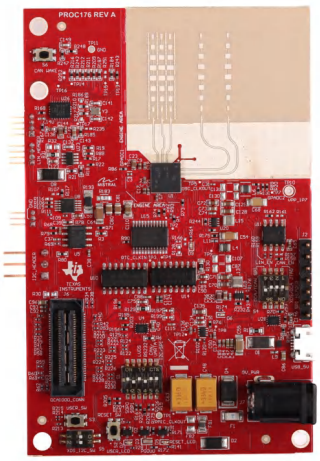
Features

- Enables car manufacturers to meet entry-level blind spot detection (BSD) requirements using a single chip radar sensor
- Detect and track objects (such as cars and trucks) up to 120 meters away with velocity of ±144 kilometers per hour (kmph)
- Antenna azimuth field of view ±60º with azimuth angular resolution of approximately 20º
- Demonstrates the following AWRL1432 capabilities:
  - Data compression
  - Max velocity extension using Chinese Remainder Theorem
  - 2D Tracking
- Improves the customer development cycle (for example, hardware redesign and software effort)

Applications

- Medium and short range radar





1 System Description

Advanced driver assistance systems (ADAS) in a vehicle provide quality-of-life and safety benefits in addition to making the relatively mundane act of driving safe and less difficult. A key safety feature is BSD, which observes the area in the rear corners of the car and alerts the driver of vehicles approaching from behind in an adjacent lane. This feature increases safety by detecting obstacles that cannot be seen by the driver in the side-view mirrors and preventing collisions when changing lanes. BSD is implemented using a variety of sensors to detect obstacles in the environment and track positions and velocities over time.

1.1 Why use Radar?

Frequency-modulated continuous-wave (FMCW) radars allow the accurate measurement of distances and relative velocities of obstacles and other vehicles; therefore, radars are useful for autonomous vehicular applications (such as lane change assist (LCA) and rear cross traffic alert (RCTA)) and car safety applications (autonomous braking and collision avoidance). An important advantage of radars over camera and light-detection and ranging (LIDAR)-based systems is that radars are relatively immune to environmental conditions such as the effects of rain, dust, and smoke. Because FMCW radars transmit a specific signal (called a chirp) and process the reflections, FMCW radars work in complete darkness and also bright daylight (radars are not affected by glare). When compared with ultrasound, radars typically have a much longer range and much faster signal transit times.

1.2 TI Corner Radar Design

The TIDEP-01034 is an introductory application where the AWRL1432 device is configured for corner radar applications to track objects within a 120m range.

1.3 Key System Specification

This reference design is based on the mmWave demo (mmwave_demo) processing chain from the mmWave low-power software development kit (MMWAVE-L-SDK). The reference design demo uses a BPM-MIMO scheme and contains additional features such as radar cube compression, maximum velocity extension, dynamic clutter removal, and a group tracker.

Table 1-1. Key System Specification

PARAMETER	SPECIFICATIONS	DETAILS
Maximum range (m)	120	This is the maximum range supported by the chirp configuration shown in Table 2-1. The maximum detectable range for a given object depends on the Radar cross section (RCS) of the object
Range resolution (m)	0.94	This is the minimum range difference over which two individual point targets can be distinguished by the chirp configuration shown in Table 2-1. The theoretical achievable resolution by AWRL1432 is 3.75cm based on 4GHz bandwidth.
Maximum velocity (kmph)	±144	This is the extended maximum velocity obtained using the Chinese Remainder Theorem on alternating frames.
Velocity resolution (kmph)	1.13	This parameter represents the capability of the radar sensor to distinguish between two or more objects at the same range but moving with different velocities.
Azimuth angular resolution (Deg)	20	This is the native azimuth angular resolution based on the virtual array aperture

2 System Overview

2.1 Block Diagram

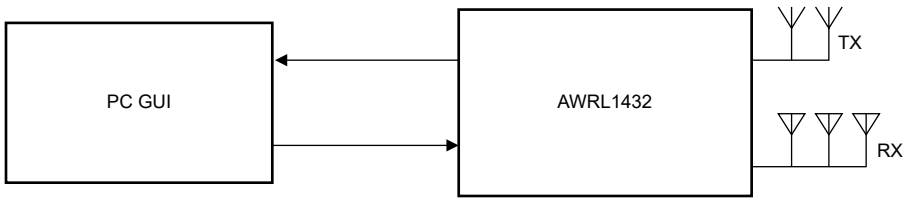


Figure 2-1. Entry-Level Blind Spot Detection System Block Diagram

2.2 Design Considerations

Today's entry-level BSD sensors typically require a radar sensor to detect and track a vehicle that is moving at 130kmph at a distance up to 120m away from the sensor. To meet these requirements, high-gain antennas with 13dB of additional gain are used on the AWRL1432BOOST-BSD to increase the max detection range up to 120m. In addition, the demo implements frame reconfiguration that is used to run alternating "slow" and "fast" frames in the RF front end. This alternation enables the maximum velocity extension feature to detect velocities up to ±144kmph.

Some vehicle architectures today require two CAN-FD interfaces for each BSD sensor to communicate with each other and the central ECU. The AWRL1432BOOST-BSD includes an integrated CAN-FD transceiver (TCAN4550-Q1), serving as a reference for system designs requiring a second CAN-FD. The TCAN4550-Q1 provides an interface between the CAN bus and the AWRL1432 through SPI.

2.3 Highlighted Products

2.3.1 AWRL1432 Single-Chip Radar Solution

TI's AWRL1432 is an integrated single-chip low-power 77GHz mmWave sensor based on FMCW radar technology, capable of operation in the 76GHz to 81GHz band. The device is partitioned into four main power domains with separate controls for each state based on use case requirements:

- RF/analog subsystem
- Front-end controller subsystem (FECSS)
- Application subsystem (APPSS)
- Hardware accelerator (HWA)

Additionally, the AWRL1432 is built with TI’s low-power 45nm RF CMOS process and enables unprecedented levels of integration in an extremely small form factor. The device has two transmitters and three receivers with a fractional-N phase-locked loop for precise and linear chirp synthesis. Based on real-only baseband architecture, the device supports an IF bandwidth of 5MHz. The presence of Arm® M4F® core, TI Radar Hardware Accelerator (HWA 1.2), and 1MB of on-chip RAM enables custom algorithm development.

2.3.2 AWRL1432BOOST-BSD Evaluation Module

The AWRL1432BOOST-BSD EVM has the following features:

- AWRL1432 radar device
- Onboard antenna (two transmitters and three receivers)
- XDS110 JTAG interface with USB connectivity for code development and debugging
- Serial port for onboard QSPI flash programming
- 60-pin, high-density (HD) connectors for raw analog-to-digital converter (ADC) data
- Onboard CAN-FD transceiver
- Onboard SPI to CAN-FD converter (TCAN4550-Q1)
- Onboard LIN PHY transceiver
- EVM is designed as booster pack to connect with other LaunchPad EVMs
- Onboard 16Mbit QSPI flash

- 12V power jack to power the board

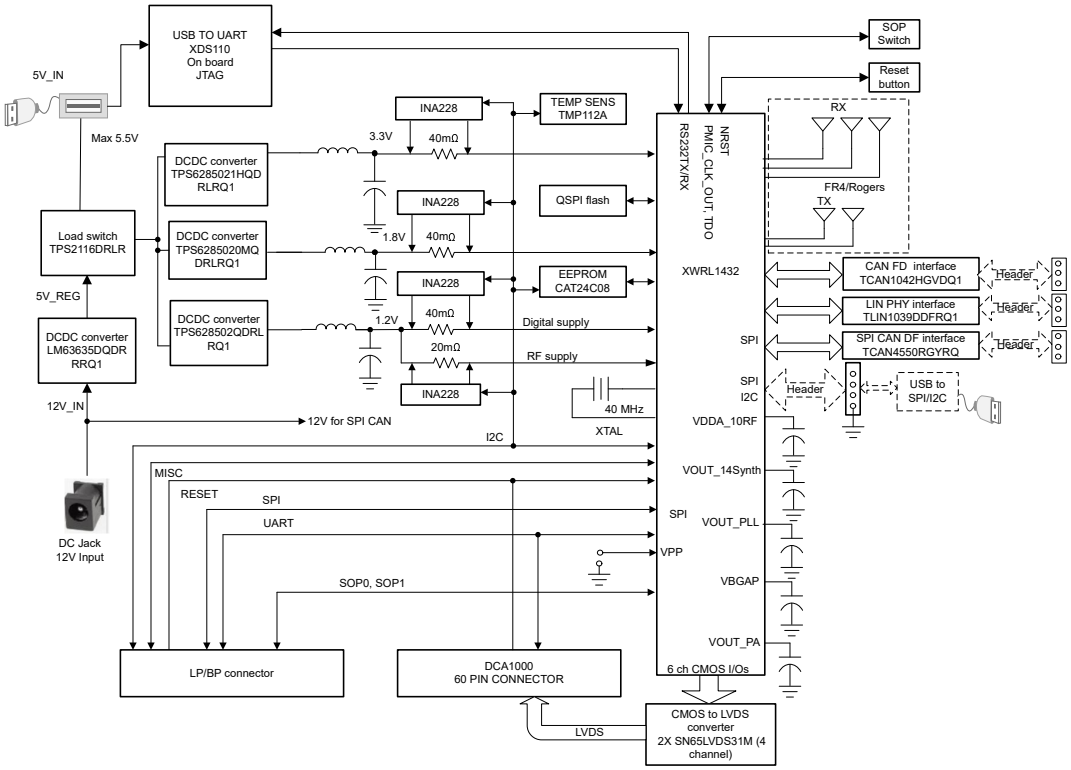


Figure 2-2. AWRL1432BOOST-BSD Block Diagram

For more details on the hardware, see the EVM User's Guide: xWRL1432BOOST-BSD. The schematics and design database are found in the following documents: xWRL1432BOOST-BSD Design Database Files and *xWRL1432BOOST-BSD Schematic, Assembly, and BOM Files*.

2.3.3 TCAN4550-Q1 Integrated CAN-FD Controller and Transceiver

The TCAN4550-Q1 is a CAN FD controller with an integrated CAN FD transceiver supporting data rates up to 8Mbps. The CAN FD controller meets the specifications of the ISO11898-1:2015 high speed controller area network (CAN) data link layer and meets the physical layer requirements of the ISO11898–2:2016 high speed CAN specification.

The TCAN4550-Q1 provides an interface between the CAN bus and the system processor through serial peripheral interface (SPI), supporting both classic CAN and CAN FD, allowing port expansion or CAN support with processors that do not support CAN FD. The device provides CAN FD transceiver functionality: differential transmit capability to the bus and differential receive capability from the bus.

The TCAN4550-Q1 includes many protection features providing device and CAN bus robustness. These features include failsafe mode, internal dominant state timeout, wide bus operating range and a time-out watchdog as examples.

2.4 System Design Theory

2.4.1 Antenna Configuration

The TIDEP-01034 uses three receivers and two transmitters, see Figure 2-3.

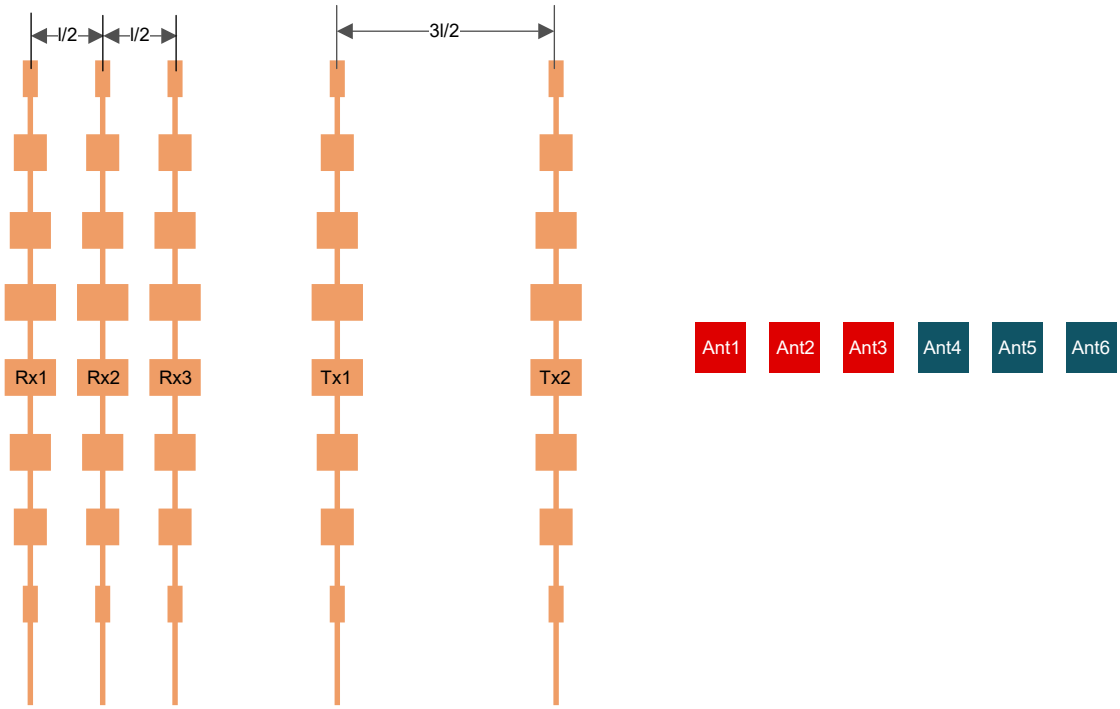


Figure 2-3. Antenna Configuration

2.4.2 Chirp Configuration and System Performance

To achieve the specific entry-level BSD use case with a max detection range of 120 m and memory availability of the AWRL1432, the chirp configuration in Table 2-1 is used. Frame reconfiguration used for this application, where the RF front end is reconfigured between alternating "slow" and "fast" frames during runtime to enable the max velocity extension feature.

Table 2-1. Chirp Configuration

PARAMETER	CONFIGURATION
Idle time (µs)	12.1 (slow), 6.9 (fast)
ADC start time (µs)	2.96
Ramp end time (µs)	23.4
Number of ADC samples	256
Frequency slope (MHz/µs)	6.7
MIMO	BPM, 2 TX pattern
Number of chirps per profile	256
Effective chirp time (µs)	35.5 (slow), 30.3 (fast)
Bandwidth (MHz)	237.9 (slow), 203.0 (fast)
Frame length (ms)	50

Table 2-2. System Performance Parameter

PARAMETER	SPECIFICATIONS
Range resolution (m)	0.94
Maximum range (m)	120
Maximum velocity (kmph)	144

Note

The configuration and parameters in Table 2-1 and Table 2-2 are based on the current application release (profile_120m_40mpsec_bsdevm_16tracks.cfg) but not limited by the device.

2.4.3 Data Path

The block diagram in Figure 2-4 shows the processing data path for the entry-level BSD application.

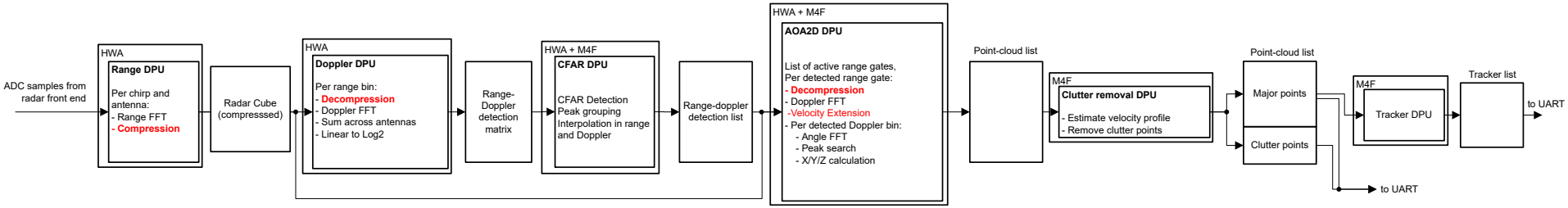


Figure 2-4. BSD Processing Datapath Flow

2.4.4 Chirp Timing

Figure 2-5 shows the timing of the chirps and subsequent processing in the system.

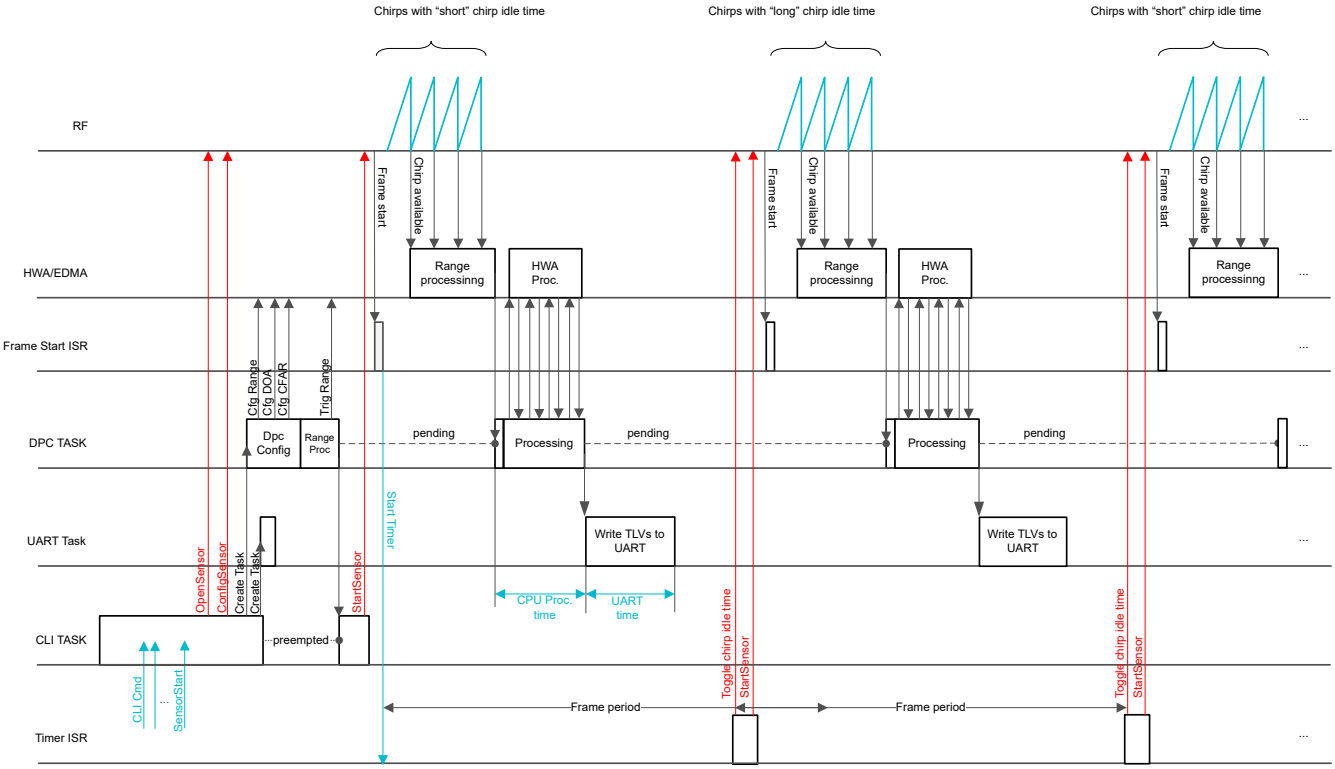


Figure 2-5. Chirp Timing Sequence

Chirp acquisition happens in the radar front end, using a BPM-MIMO scheme. Due to the implementation of the maximum velocity extension feature in the BSD demo, which relies on varying chirp durations from frame to frame, the chirp duration ("short" and "long") is reconfigured with each frame.

The core of the data path processing, from chirp acquisition to point cloud and tracker output, is divided into the following data processing units (DPUs):

- Range DPU
- Doppler DPU
- CFAR DPU
- 2D Angle-of-Arrival (AoA2D) DPU
- Clutter Removal DPU
- Group Tracker DPU

As the acquisition occurs, the Range DPU performs 1D FFT for each antenna and chirp in parallel to the acquisition, compresses the output, and stores the output in memory as the compressed radar cube.

Next, the Doppler DPU decompresses the radar cube one range bin at a time and for each bin calculates velocity information by performing Doppler FFTs. The Doppler FFT magnitudes are sum across all virtual antennas for each range bin to create a Range-Doppler detection matrix. This is done in the hardware accelerator (HWA).

After this, the CFAR DPU then computes and crosschecks detected points in both range and Doppler dimensions. The AoA2D DPU then performs the maximum velocity extension algorithm to extend the velocities of detected points beyond Nyquist limits and generates a point cloud list in Cartesian format. These DPUs utilize both the HWA and M4F to achieve this.

Finally, the Clutter Removal DPU removes detected points identified as stationary roadside clutter and the Group Tracker DPU performs object tracking based on the point cloud data. Both the final point cloud list and tracker list are transmitted over UART.

For more details on the application flow and processing, see the *mmWave low-power software development kit (MMWAVE-L-SDK)*.

2.4.5 Memory Allocation

Note

The BSD demo memory usage can always be found in the MAP file.

Two memory heaps are created, one in the local core memory, and the other in shared L3 memory, occupying 256KB shared memory of APPSS and 160KB shared memory of HWASS. The usage of the shared memory depends on the configuration. At the start time, after the CLI commands have been received and configuration completed, memory usage of these two heaps is printed out on the console. A buffer is allocated from the local core memory for the group tracker algorithm.

The memory heap usage for the CLI configuration file profile_120m_40mpsec_bsdevm_16tracks.cfg which is provided in this SDK release is shown in Table 2-3.

Table 2-3. Memory Heap Usage for CLI Configuration

Memory Heap Array	Usage	Size (Bits)	Used	Free
Shared L3 Memory	Radar cube, detection matrices	425984	263056	162928
Local Core Memory	Processing chain data	23552	10384	13168
Local Memory Buffer	Tracker	49152	19968	29184

2.4.6 Frame Reconfiguration

The timing diagram of the BSD demo processing chain during the startup and the first two frames is illustrated in Figure 2-5. The BSD demo processing chain differs from the typical SDK processing chain in the way the frames are triggered. In the typical SDK OOB demo processing chain, the RF front end is set up for an infinite sequence of frame numbers, and the RF drives the frames according to a given frame period. In contrast, in this processing chain, the RF front end is configured for only one frame, and the RTOS timer is set at the given frame rate.

Upon the RTOS timer ISR execution, the RF front end is reconfigured and initiated to run one frame at a time. Consequently, every other frame, the chirp idle time is toggled, resulting in the alternating “slow” and “fast” frames. This alternation represents the required behavior for the operation of the maximum velocity extension feature.

2.4.7 Vmax Extension

Based on the configuration parameters passed to the device, the device is able to detect velocities in the range of $-V_{max}$ to $+V_{max}$, where V_{max} is defined by the following equation:

$$V_{max} = \frac{\lambda}{4\left(T_{IdleTime} + T_{RampEndTime}\right)N_{TxAnt}}$$

where $T_{IdleTime}$ is the chirp idle time, $T_{RampEndTime}$ is chirp ramp end time, and N_{TxAnt} is the number of transmitter antennas. Both $T_{IdleTime}$ and $T_{RampEndTime}$ can be defined in the configuration profile.

To enable a higher V_{max} , $T_{IdleTime}$ and $T_{RampEndTime}$ must be decreased. However, there is a physical limit defined by the radar front end below which $T_{IdleTime}$ and $T_{RampEndTime}$ cannot be decreased. Any velocities detected beyond this V_{max} would loop around and is termed as ambiguous velocity since we are no longer able discern the actual value. In order to overcome this and increase V_{max} , the Chinese Remainder Theorem is applied.

The Chinese Remainder Theorem uses consecutive frames with alternate idle times and hence alternate V_{max} . For each detected point, an N_H number of hypotheses are set, where N_H is an odd number. For each hypothesis, the actual velocity is computed using the current frame velocity resolution $\Delta v_{CurrFrm}$ and doppler index of the detected point d_{ind_p} as:

$$v\left(i\right) = \Delta v_{CurrFrm} \times d_{ind_p} + k \times 2V_{max_{CurrFrm}}, \quad k = i - \left[\frac{N_H}{2}\right], \quad i = 0, 1, \ldots, N_H - 1$$

where i is the index of the hypothesis.

Then, the velocity of each hypothesis is mapped (folded) into the previous frame detection matrix using the previous frame Doppler resolution $\Delta v_{PrevFrm}$ to get the Doppler index position:

$$d_{ind}\left(i\_ = round\left(\frac{v(i)}{\Delta v_{PrevFrm}}\right) \mod N_{DopFFT}, \quad i = 0, 1, \ldots, N_H - 1\right.$$

In addition, the range index of each hypothesis is corrected based on the predicated range migration:

$$\mathrm{r_{ind}}\left(\mathrm{i}\right) = \mathrm{round}\left(\mathrm{r_{ind_p}} - \mathrm{v}\left(\mathrm{i}\right) \times \mathrm{T_{frame}}\right), \quad \mathrm{i} = 0, 1, \ldots, \mathrm{N_H} - 1$$

where r_{ind_p} is the range index of the detected point and T_{frame} is the frame period.

As a result of these operations, the N_H points in the previous frame detection matrix are created, each defined as a range-doppler index pair: $\{r_{ind}(i), d_{ind}(i)\}, \quad i = 0, 1, \ldots, N_H - 1$

Then, for each hypothesis, the search in the previous frame detection matrix D_{prev} is done in the vicinity (rectangular region) of the point to find the local maximum peak. The search is done in a rectangular region of size $(2D_s + 1) \times (2R_s + 1)$, where D_s and R_s are specified by the configuration.

Based on which hypothesis has the largest $p_{max}(i)$, the corresponding winning hypothesis is chosen: $p_{max}\left(i\right) = max_{v_{ind} = v_{ind}(i) - D_s}^{v_{ind}(i) + D_s} max_{r_{ind} = r_{ind}(i) - R_s}^{r_{ind}(i) + R_s} D_{prev}\left(v_{ind}, r_{ind}\right), \quad i = 0, 1, \ldots, N_H - 1$ Finally, the actual target velocity v_{actual} is selected based on the winning hypothesis:

$$v_{actual} = v\left(m\right), \quad m - argmax_{i = 0}^{N_H - 1}\left(p_{max}\left(i\right)\right)$$

This process is summarized by Figure 2-6.

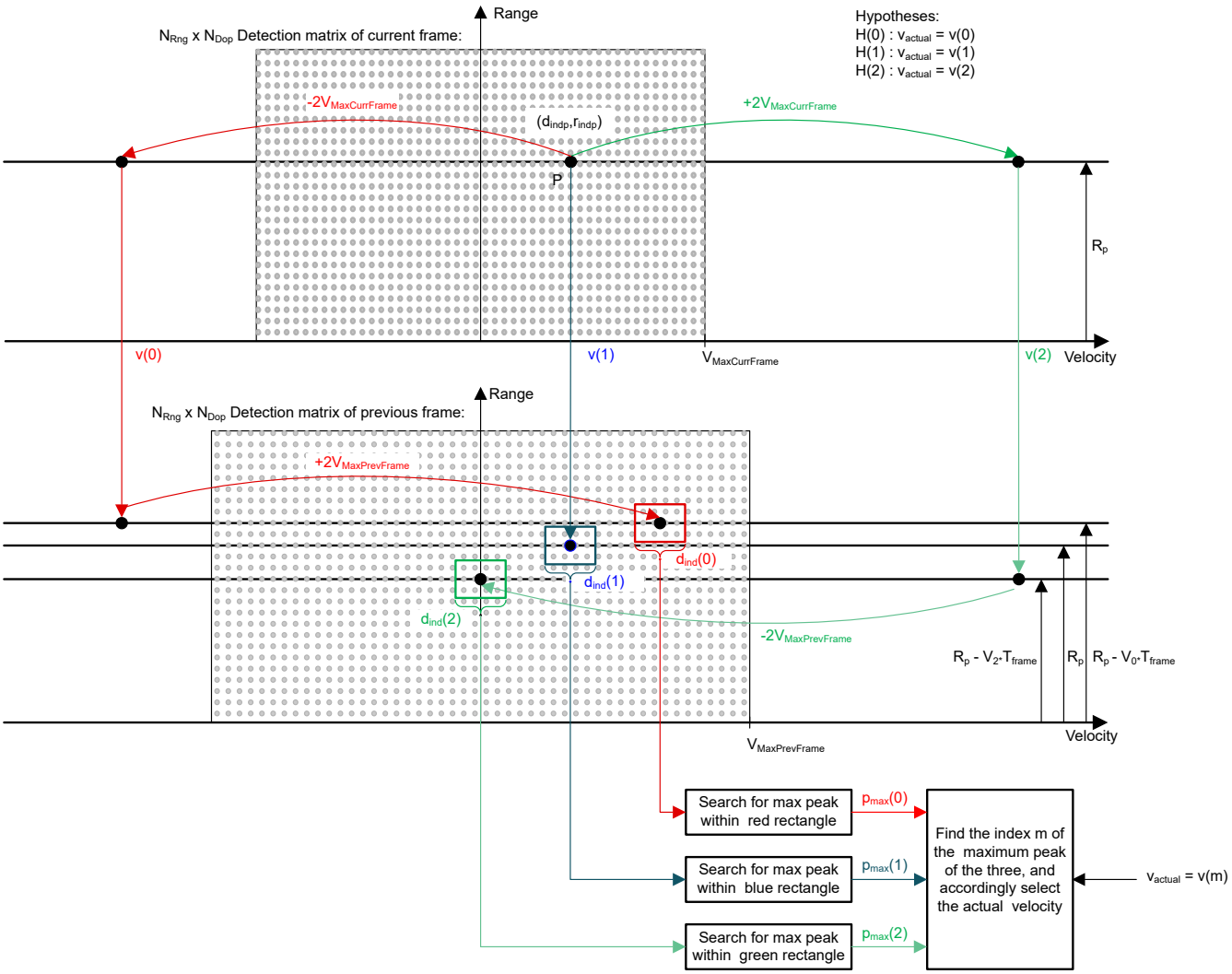


Figure 2-6. Max Velocity Extension Processing

2.4.8 Group Tracker

Real world radar targets (cars, pedestrians, and so forth) are presented to a tracking processing layer as a set of multiple reflection points. Those detection points form a group of correlated measurements with range, angle, SNR, and radial velocity. The group tracker, tracks a cluster of points (also known as group) in 2D over time based on a constant acceleration motion model. Figure 2-7 shows the main functional blocks of the group tracker algorithm. The subblocks shown in white are classical extended Kalman filter (EKF) operations. The subblocks shown in orange are additions to support multipoint grouping.

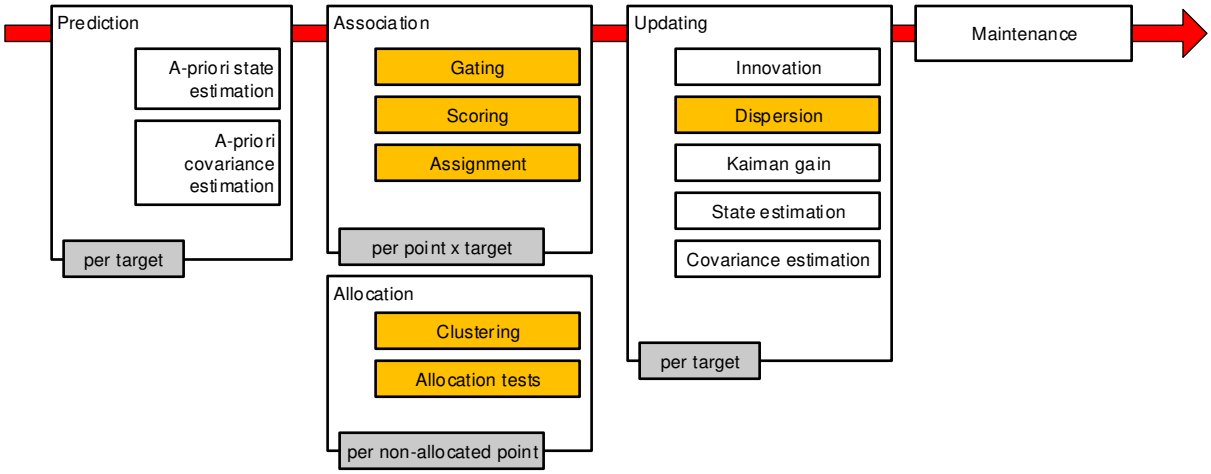


Figure 2-7. Group Tracker Block Diagram

These parameter sets can be tweaked based on test results or as per scene requirement.

Table 2-4. GTrack Parameter Sets

SCENARIO	PARAMETER SETS	CLI COMMANDS	DESCRIPTION
1	Scenery Parameters	appSceneryParams	These define the dimensions of the physical space in which the tracker operates. These also specify the radar sensor orientation and position. Any measurement points outside these boundary boxes are not used by the tracker.
2	Gating Parameters	appGatingParams	These determine the maximum volume and velocity of a tracked object and are used to associate measurement points with tracks that already exist. Points detected beyond the limits set by these parameters are not included in the set of points that make up the tracked object.
3	Allocation Parameters	appAllocParams	These are used to detect new tracks or people in the scene. When detected points are not associated with existing tracks, allocation parameters are used to cluster these remaining points and determine if that cluster qualifies as a person or target.
4	State Parameters	appStateParams	The state transition parameters determine the state of a tracking instance. Any tracking instance can be in one of three states: FREE, DETECT, or ACTIVE.
5	Max Acceleration Parameters Max number of points Max number of tracks	gtrack	Max acceleration parameters determine the maximum acceleration in the lateral, longitudinal, and vertical directions.

2.4.9 Dynamic Clutter Removal

Dynamic clutter removal is an optional feature in the BSD demo processing chain designed to eliminate reflections from stationary roadside objects and the road, referred to as "dynamic clutter." A common feature of dynamic clutter is that all stationary points in the radial velocity-azimuth angle coordinate system are positioned alongside a sinusoidal curve, referred to as the "velocity profile." The velocity profile, shown in Figure 2-8, can be represented as $V_r(\theta) = V_s\cos(\theta + \alpha)$

where V_s denotes the speed of the sensor and α denotes the sensor mounting angle.

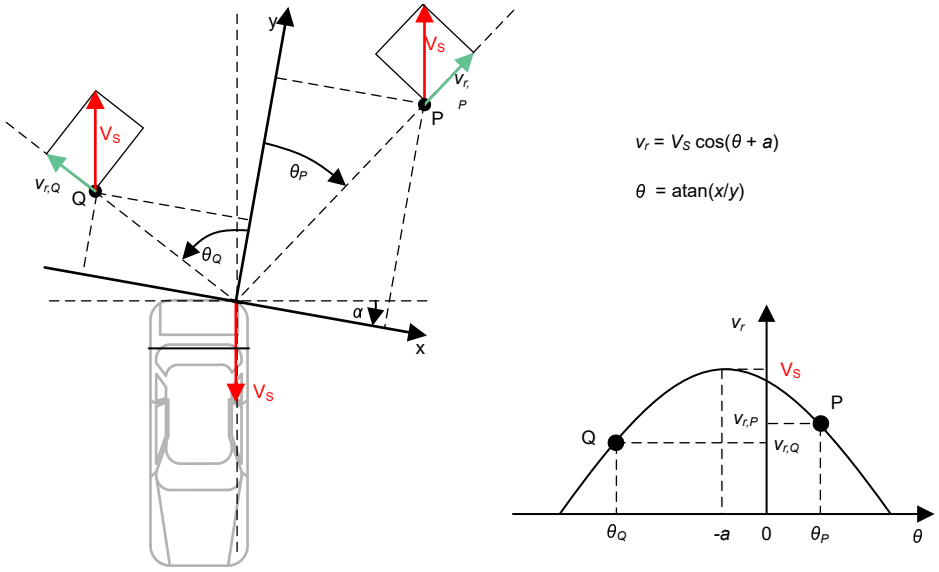


Figure 2-8. Velocity Profile of Dynamic Clutter

By estimating V_s and α from near-range detected points, the dynamic clutter removal algorithm is able to filter dynamic clutter from other points that are relevant for the BSD application.

In the BSD demo, clutter removal is performed by the Dynamic Clutter Removal DPU located between AoA2D DPU and the Group Tracker DPU as shown in Figure 2-4. The stationary points are removed from the point cloud list after the AoA2D DPU, and the remaining points, the major points, are supplied to the tracker DPU. The clutter points can optionally be saved and appended to the point cloud list, and at the end of the processing can sent out together with the major points to the GUI visualizer.

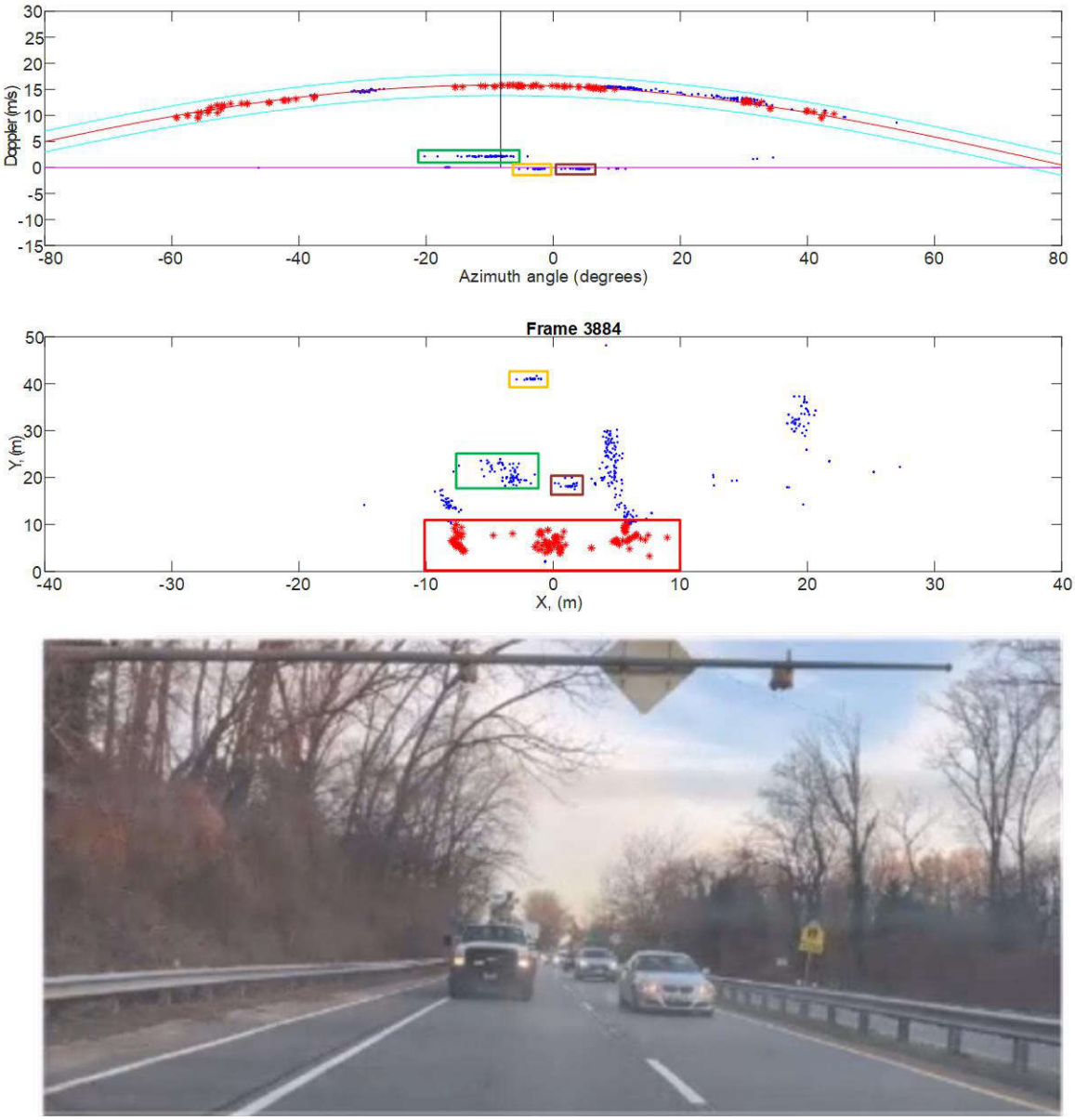


Figure 2-9. Dynamic Clutter Removal Implementation

The clutter removal implemented in the BSD demo is depicted in Figure 2-8. The bottom of the figure shows the scene facing the sensor mounted at the rear of the moving vehicle. The middle section illustrates the point clouds corresponding to the scene in the x-y plane, while the top section illustrates the same point-cloud in the radial velocity-azimuth angle plane. Points within the red rectangle in the near range highlighted in red are assumed to mainly represent reflections from the road surface. Points within the yellow, green and purple rectangles correspond to reflections from the three vehicles. The remaining points represent reflections from other stationary objects. In the velocity-azimuth plane, points attributed to stationary objects clearly align within a narrow corridor following the velocity profile curve. With knowledge of V_s and α , these points can be easily filtered out.

Note

The BSD demo currently estimates the velocity profile based on the detection points reflected from the nearby road surface behind the vehicle. The algorithm operates under the assumption that the reflections from the nearby road surface are almost constantly present. However, to enhance reliability, more reliable estimates can be obtained from other sensors, such as the vehicle odometer or another sensor positioned closer to the ground, such as the sensor employed for the “kick to open” feature.

2.4.10 CAN-FD Transceiver

The AWRL1432BOOST-BSD includes an integrated CAN-FD transceiver (TCAN4550-Q1), serving as a reference for system designs that require a second CAN-FD (in addition to the integrated CAN-FD interface on the AWRL1432). The TCAN4550-Q1 provides an interface between the CAN bus and the AWRL1432 through SPI, supporting both classic CAN and CAN-FD.

- The TCAN4550-Q1 SPI signal pins are connected to the integrated SPIA interface of the AWRL1432.
- The TCAN4550-Q1 device reset pin (RST) is connected to GPIO 6 of the AWRL1432 and is used to reset the device to the default settings and put the device into standby.
- The TCAN4550-Q1 wake-up request pin (nWKRQ) is connected to GPIO 4 of the AWRL1432 and is used to serve as an enable for a regulator that does not use the INH pin to control voltage level. This is not used in the demo as the INH pin provides voltage to enable an external high voltage regulator.
- The TCAN4550-Q1 interrupt pin (nINT) is connected to GPIO 5 of the AWRL1432 and is used to transmit all interrupt requests.
- The TCAN4550-Q1 WAKE pin is connected to a user button (S6) and is used for high voltage device local wake-up (LWU) to transition the device into standby mode.

Because the TCAN4550-Q1 provides 2K bytes of MRAM that is fully configurable for TX/RX buffer/FIFO as needed based upon the system needs, implementing a second CAN-FD functionality does not add any overhead to the AWRL1432 memory.

The TCAN4550-Q1 is configured to the timing parameters shown in Table 2-3 for the BSD demo.

Table 2-5. CAN-FD Timing Parameters

	Bit Rate (kbit/s)	Sample Point (%)	Prescaler	TSEG1	TSEG2	SJW	Tq (ns)	Nq
Nominal	500	80	2	31	8	8	50	40
Data	2000	75	1	14	5	5	25	20

For more details on the TCAN4550-Q1 features, see the TCAN4550-Q1 data sheet.

For more details on the SPI to CAN-FD driver code, see the TCAN45xx Software User's Guide.

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

The AWRL1432BOOST-BSD is an easy-to-use evaluation board for the AWRL1432 for the entry-level BSD application.

The demo runs on the AWRL1432BOOST-BSD and connects to a visualization tool running on a PC connected to the EVM over USB. Details for using this board are found in the EVM User's Guide: xWRL1432BOOST-BSD. The entry-level BSD design is an application built using the mmWave Low-power Software Development Kit (MMWAVE-L-SDK). The demo binaries found in the Radar Toolbox from the TI Resource Explorer. The MMWAVE-L-SDK version used to build this code is provided in the demo software release notes.

3.1.1 Hardware

The AWRL1432BOOST-BSD core design includes:

- AWRL1432 device: a single-chip, low-power 77GHz radar device with an integrated Arm-Cortex M4F and HWA
- Power optimized discrete DCDC power management
- The EVM also hosts a device to assist with onboard emulation and UART emulation over a USB link with the PC.

3.1.2 Software and GUI

Associated software is hosted on the TI Resource Explorer Radar Toolbox.

The MATLAB GUI for the demo is provided in the software package and displays the following plots as shown in Figure 3-1:

- X-Y scatter plot: displays the positions of the point clouds, the tracks
- Doppler range plot: displays the Doppler-range coordinates of the point cloud and tracks

Upon loading, the GUI provides users the options to record a test capture, display a test capture, or play back a recorded test capture. The GUI allows the user to connect the demo via UART (UART ports are configured based on the device manager settings) and load the configuration file to start a test capture.

3.2 Test Setup

The performance of the AWRL1432 corner radar functionality was tested using the example project available on TI Resource Explorer in the Radar Toolbox. The AWRL1432BOOST-BSD mounted on a tripod was used for the tests.

3.3 Test Results

The results in Figure 3-1 and Figure 3-2 -correspond to the maximum range detected for mid-size sedans and motorbikes.

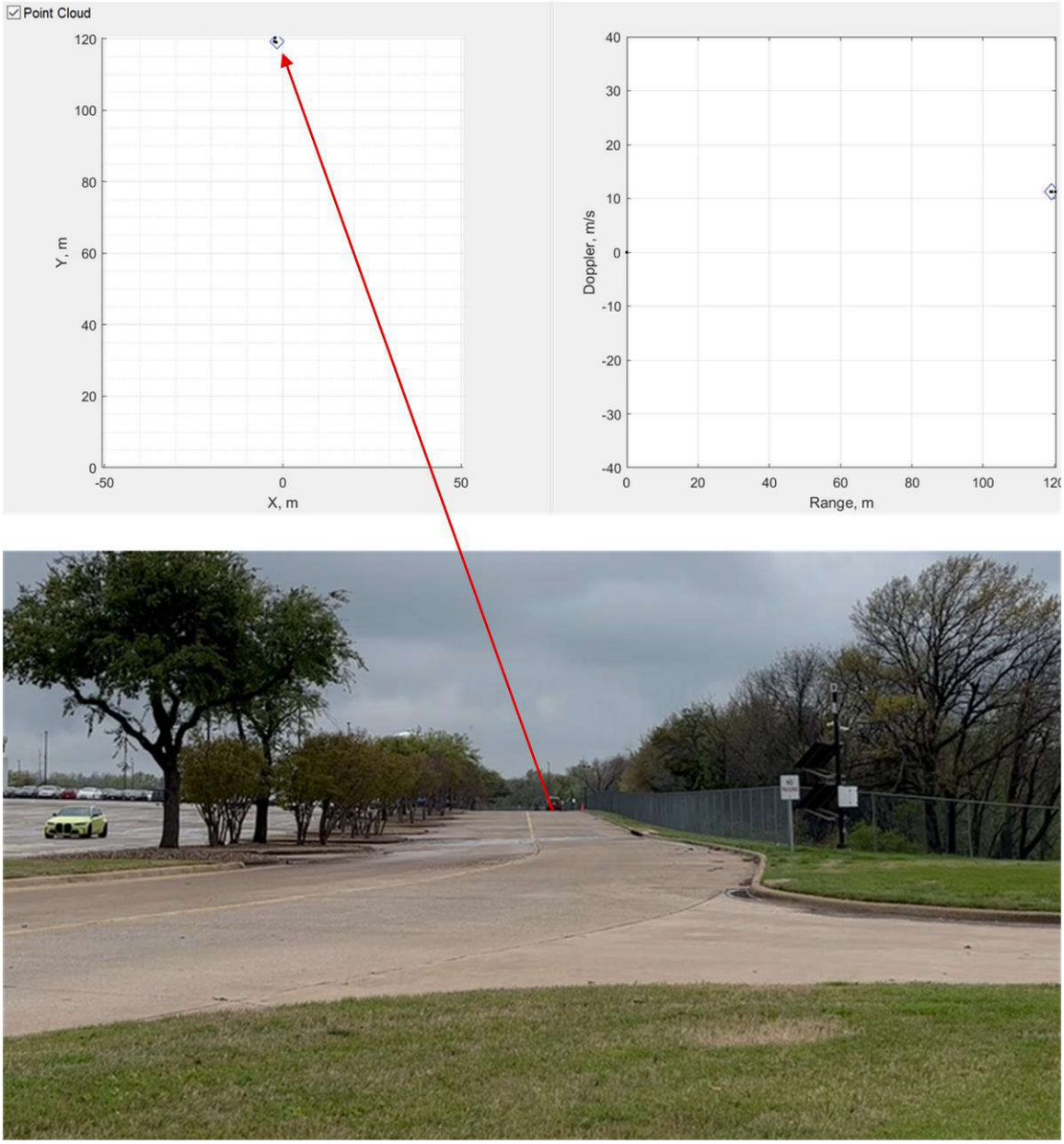


Figure 3-1. Test Result (Car Approximately 120m)

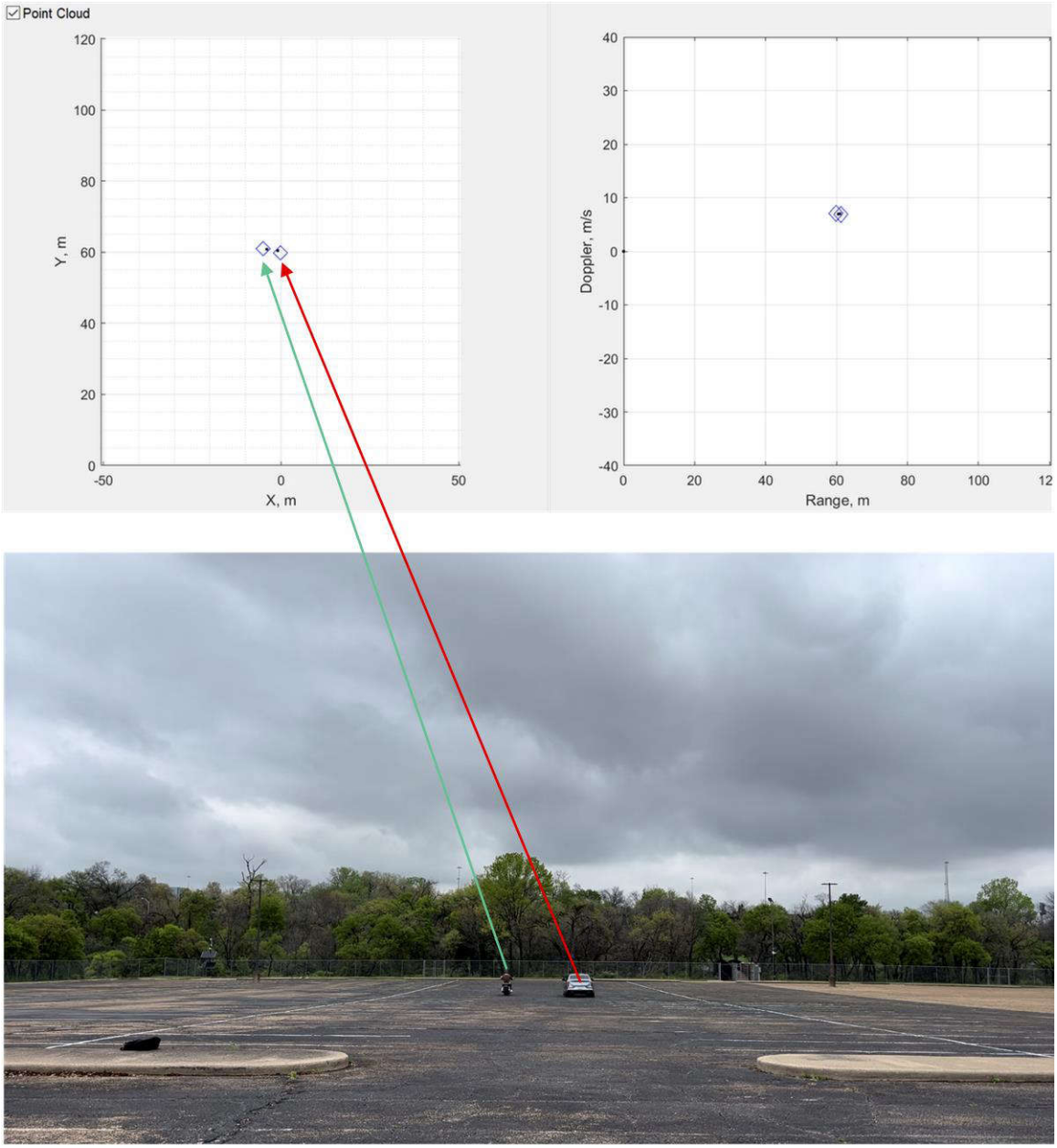


Figure 3-2. Test Result (Bike Approximately 60m)

4 Design and Documentation Support

4.1 Design Files

4.1.1 Schematics

To download the schematics, see the design files at TIDEP-01034.

4.1.2 BOM

To download the bill of materials (BOM), see the design files at TIDEP-01034.

4.2 Tools and Software

Tools

Code Composer Studio (CCS)	Code Composer Studio™ software is an integrated development environment (IDE) that supports TI's microcontroller (MCU) and embedded processor portfolios. This tool is used to build and debug the software and processing chain of the application.

Software

Application Software	Software for the TIDEP-01034 is hosted on the TI resource explorer, look for Blind Spot Detection under Example Projects at this link.

4.3 Documentation Support

1. Texas Instruments, *EVM User's Guide: xWRL1432BOOST-BSD*
2. Texas Instruments, *TCAN4550-Q1 Automotive Controller Area Network Flexible Data Rate (CAN FD) System Basis Chip with Integrated Controller and Transceiver* data sheet
3. Texas Instruments, *Group Tracker Parameter Tuning Guide for the 3D People Counting Demo*

4.4 Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

4.5 Trademarks

TI E2E™ and Code Composer Studio™, and are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265