



**UNIVERSITY OF
BIRMINGHAM**

Cognitive Real-Time System for Autonomous Vehicles

(104268-CORTEX)

**Automotive Radar and Video with GPS/IMU Ground Truth
Dataset from Trials at MIRA Test Track**

Repository Description

Project	CORTEX
Document Title:	Data Repository Description
Document Reference	104268 – Mira Trials Repository Description
Issue	Rev. A
Date:	31/03/22

Version History

Version	Date	Incorporated by	Comments
A	30/03/22	Sukhjit Pooni	1 st Version

This document is approved by the Principle Investigator (PI) of the CORTEX project.

Prof. Marina Gashinova

Date: 31/03/2022

Table of Contents

Abbreviations.....	4
1. Introduction	5
1.1 Objectives during the Trials	5
1.2 Measurement Site and Scene.....	6
2. Hardware Configuration	7
2.1 Equipment used during the Trials	8
2.2 Radar Operation	8
2.3 Radar Parameters.....	9
2.4 Video	10
3. Measurement Scenes	10
3.1 MIRA Run 1.....	11
3.2 MIRA Run 2.....	12
3.3 MIRA Run 3.....	13
3.4 MIRA Run 4.....	14
3.5 MIRA Run 5.....	15
3.6 MIRA Run 6.....	16
3.7 MIRA Run 7.....	17
4. Repository Folder Structure.....	18
5. Data Structure	19
5.1 INRAS Radarlog	19
5.2 ZED Stereo Camera Data.....	20
5.3 Velodyne Data.....	20
5.4 Mira GPS data	20
5.5 Time Synchronization data.....	20
6. MATLAB Sample Codes	21
6.1 Reshaping Data for MIMO mode.....	21
6.2 Selecting $\lambda/2$ spaced Virtual Channels	21
6.3 Reading .h5 file from Radarlog.....	22
6.4 Radarlog Data Antenna Calibration	22
6.5 Time Synchronization	22

Abbreviations

UoB	University of Birmingham
IMU	Inertial measurement unit
HDF5	Hierarchical Data Format 5
FoV	Field of view
RLG	Radarlog

1. Introduction

This report describes the measurement scenarios, and data structure of various sensors used for data collection campaign conducted by the University of Birmingham (UoB) and the project partners on 10th September 2021 at Horiba MIRA test track.

The dataset consists of seven parts as given in below with corresponding links to where they are stored in the University of Birmingham Repository. Each of these runs last for 30secs, where the stored radar data for each run can be up to 7GB. As such each run will be stored as a separate data repository.

- **MIRA Test Track Run1.** Car going round the city circuit with an average speed of 18mph. <https://doi.org/10.25500/edata.bham.00000809>
- **MIRA Test Track Run2.** Car going round the city circuit with an average speed of 14mph. <https://doi.org/10.25500/edata.bham.00000812>
- **MIRA Test Track Run3.** Car going within the city circuit, past structures such as traffic lights and metallic gantries with an average speed of 20mph. <https://doi.org/10.25500/edata.bham.00000813>
- **MIRA Test Track Run4.** Car going within the city circuit, past structures such as traffic lights and metallic gantries with an average speed of 13mph. <https://doi.org/10.25500/edata.bham.00000814>
- **MIRA Test Track Run5.** Car going round the city circuit with an average speed of 15mph. <https://doi.org/10.25500/edata.bham.00000815>
- **MIRA Test Track Run6.** Car going round the city circuit with an average speed of 27mph. <https://doi.org/10.25500/edata.bham.00000816>
- **MIRA Test Track Run7.** Car going within the city circuit, past structures such as traffic lights and metallic gantries with an average speed of 20mph. Wide variance of speed is done with this run. <https://doi.org/10.25500/edata.bham.00000817>

Chapter 2 provides a list of sensors used during the measurement campaign, their configuration, and parameters. Chapter 3 gives an overview of the measurement scenes for each of the runs. Chapter 4 gives a description of repository folder structure. Chapter 5 will detail the data structure of the different files in the repository. Chapter 6 gives minimal MATLAB codes that will aid in the processing of the collected data.

1.1 Objectives during the Trials

The objectives of the described radar trials were to collect 77GHz MIMO radar images that will allow Doppler Beam Sharpening (DBS) and further processing such as radar image segmentation

1.2 Measurement Site and Scene

The measurement site is located on Horiba MIRA test track, Nuneaton, U.K. A Google Map plan view of the trials site is shown in Figure 1 (a) and the panoramic photo of the test area is shown in Figure 1 (b).



(a)



(b)

Figure 1: Pictorial representation of the test area. (a) Google Map plan view. (b) Panoramic photo.

2. Hardware Configuration

Data collection for the trials is performed using a UoB vehicle. Most of the equipment including all the sensors is installed at the front is shown in Figure 3. The power supplies are secured at the boot of the vehicles whereas IMU is placed on the roof of the vehicles as indicated in Figure 3.

The height of the Radarlog above the ground is 0.85m.

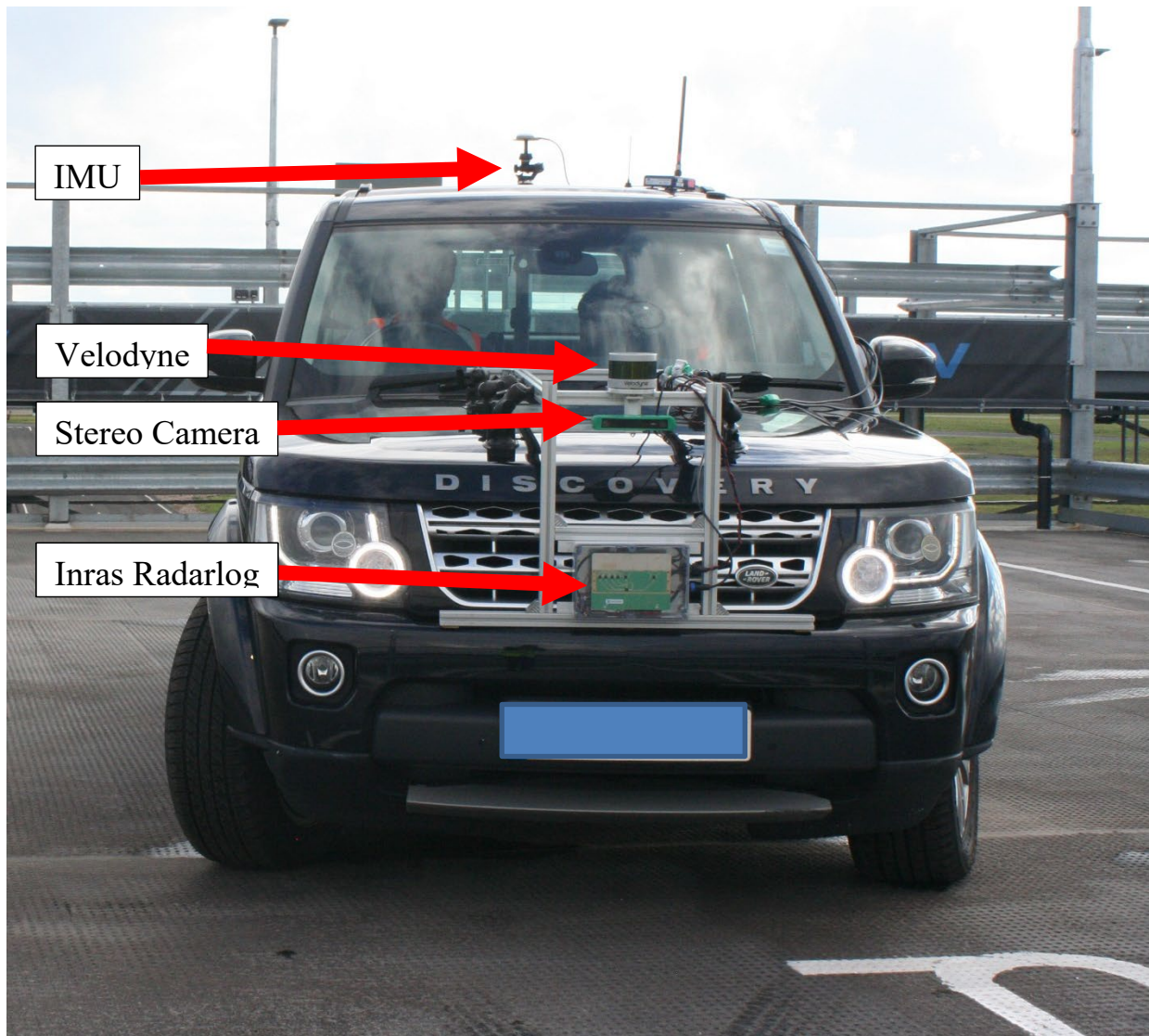


Figure 2: Equipment setup on UoB vehicle

2.1 Equipment used during the Trials

The list of equipment used during trials is mentioned in Table 2. The Radarlog, LIDAR and ZED stereo camera are configured to record timestamped data.

Table 1: Equipment used during COSMOS Trials 2.

Equipment	Model	Notes
Radars	INRAS Radarlog	MIMO mode 4Tx – 16Rx
Video/ Camera	Stereo labs ZED stereo camera	
Lidar	Velodyne VLP-16	
GPS/ Ground Truth	MIRA IMU System UoB Advanced Navigation Spatial FOG IMU/ GNSS	The MIRA IMU system is the main system used. The UoB FOG IMU was there as a backup.

2.2 Radar Operation

For the trials, the radar is operated in MIMO configuration (all transmitters are active in time-division multiplexing MIMO mode) in order to have a better spatial resolution. The chirp and frame configuration are shown in Figure 4.

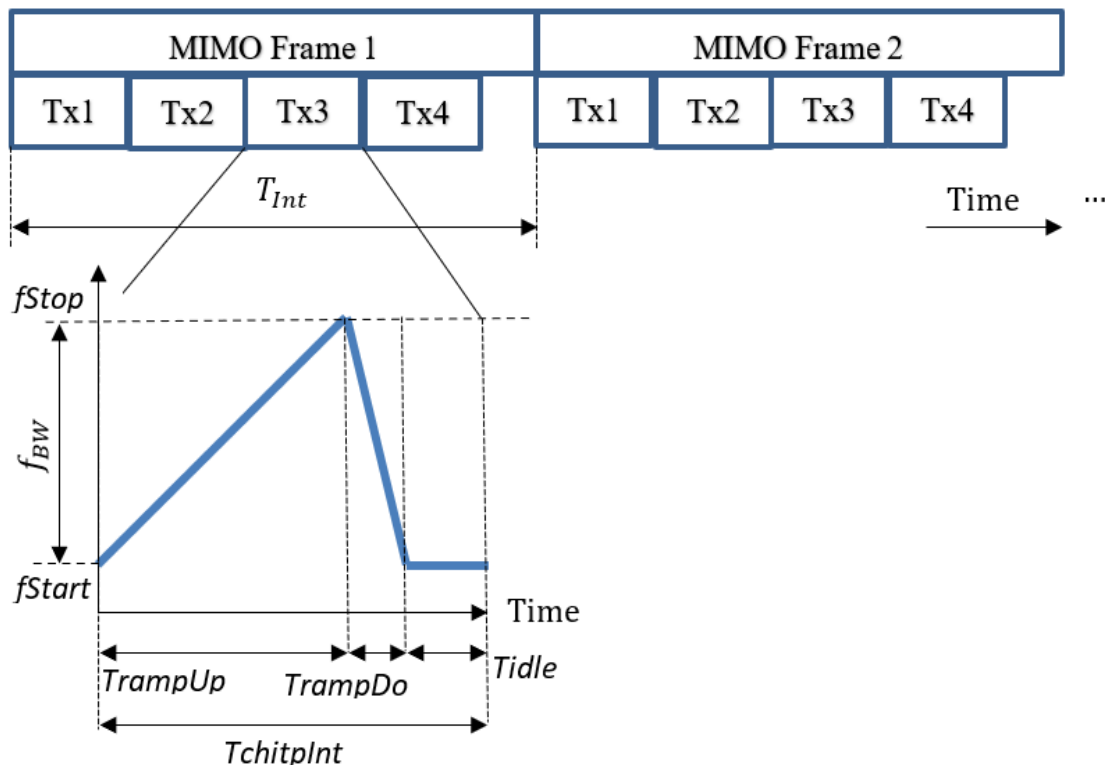


Figure 3: Chirp and Frame configuration in MIMO mode.

INRAS Radarlog is a 77GHz radar with four transmit elements at 3.5λ spacing and sixteen receive elements at $\lambda/2$ spacing. It is configured to operate in MIMO mode.

The computer connected to Radarlog runs Inras RadServe software which captures the radar data, and time stamp information. The radar data and configuration settings are then stored to a HDF5 (Hierarchical Data Format version 5) file. The RadServe software enables high data transfer rate (up to 2.5Gbit/s) and ensures synchronization between the captured Radarlog data and time stamping information.

The active transmitters of Radarlog cycle from chirp-to-chirp in the order: TX1-TX2-TX3-TX4. With four transmit elements and sixteen receive elements, the Radarlog produces a virtual array of 64-elements, but the spatial configuration of the antennas on the Radarlog, gives three overlapping virtual elements. After removing these overlapping elements, an equivalent of a 61-element $\lambda/2$ spaced antenna array (one-way only) is formed. Removal of the overlapping elements is done before MIMO processing of the radar signals from the Radarlog.

Raw data recorded to file is the de-ramped FMCW IF signal.

2.3 Radar Parameters

Operating parameters used for the radar along with the reference variable names are given in Table 4. These parameters have been determined to give the optimal range/ azimuth resolutions, negligible data transfer loss to the computer.

Table 2: Radar Parameters for CORTEX Trials

Parameter	Symbol	INRAS Radarlog	Unit
Sweep Time (Ramp-Up)	T_{rampUp}	204.8	us
Sweep Bandwidth	f_{BW}	2000	MHz
Start Frequency	f_c	76.1	GHz
Mode of Operation		MIMO	
Chirp Ramp-Down Time	T_{rampDo}	18.8	us
Chirp Interval Time	$T_{chirpInt}$	230	us
Chirps per Frame		4	Chirps
Active Frame Duration	T_{active}	0.92	ms
MIMO Frame Interval	T_{Int}	1	ms
Active Duty Cycle		92	%
Antenna Configuration		4*16	Elements
Azimuth Resolution		1.9	Degree
Sampling Frequency		10	MHz
Transmit Antenna Gain		14.4	dBi
Receive Antenna Gain		14.4	dBi
Transmit Power		10	dBm
Number of fast time samples	$n_{samples}$	2048	
Range Resolution	R_{res}	0.075	m
Maximum unambiguous range	R_{unamb}	150	m
Maximum unambiguous Doppler Velocity	v_{unamb}	± 1	ms^{-1}

2.4 Video

A ZED stereo video camera has been used to provide ground truth the radar measurements. Camera data is supplied as an H.264 encoded video file in an MP4 container and is timestamped to allow for co-registration with the radar data. However, there is some degree of error due to latency.

The computer connected to ZED captures video snapshots from the webcam, that are later synchronised with the radar data using time stamp information.

It is to be noted that the video ground truth is captured from the front radar perspective.

3. Measurement Scenes

This chapter will show example scenes, viewed by the optical ZED camera, from the different runs described by this document. For each run example scenes are shown from the beginning, middle and end of each run.

3.1 MIRA Run 1



Start of Run



Middle of Run



End of Run

3.2 MIRA Run 2



Start of Run



Middle of Run



End of Run

3.3 MIRA Run 3



Start of Run



Middle of Run



End of Run

3.4 MIRA Run 4



Start of Run



Middle of Run



End of Run

3.5 MIRA Run 5



Start of Run



Middle of Run



End of Run

3.6 MIRA Run 6



Start of Run



Middle of Run



End of Run

3.7 MIRA Run 7



Start of Run



Middle of Run



End of Run

4. Repository Folder Structure

For each data capture run the following data files are stored. The data structure of each is given in the next chapter.

- **radarlogData.h5**: raw data from the Inras Radarlog, along with chirp timing
- **video.mp4**: shows the optical video image from the left camera from the ZED
- **test_zed.svo.dat**: Contains the timestamps of each frame from the ZED stereo camera
- **test_velodyne.dat**: Contain the local timestamps from the Velodyne
- **test_gps.dat**: Contains the GPS timestamps from the Velodyne
- **miraGPSdata.txt**: Contains information from the MIRA GPS system

Across all capture runs the following file is common

- **RDL_PC_TimeServer_210910_1538.dat**. This holds the time delay information between the PC operating the radarlog and the operating the combination of the ZED plus Velodyne.

5. Data Structure

This section describes the structure of the data collected from the radars by the project team at the University of Birmingham. The structure of the important variables in this file are described within this section, with the variable name given in brackets.

5.1 INRAS Radarlog

The Inras radar data along with ints configuration parameters are stored in stored to a HDF5 (Hierarchical Data Format version 5) file.

The configuration parameters are stored in the HDF5 file as attributes. The useful parameters are listed in Table 5. Note a lot more attributes are stored, but the following are the useful ones.

Table 3: Useful Radarlog parameters stored in the HDF5 file

Name	Description	Note
CalIm	Imaginary values of the phased array calibration matrix	Is a 64-element array for the 64 virtual elements of the radarlog
CalRe	Real values of the phased array calibration matrix	Is a 64-element array for the 64 virtual elements of the radarlog
N	Number of fast time samples in the data	
TInt	Mimo frame interval time in secs	
TRampDo	Ramp down time in secs	
TRampUp	Ramp up time is secs	
Tp	Chirp Interval time (within a MIMO frame)	
fStop	Stop frequency of the FMCW signal in Hz	
fStart	Start frequency of the FMCW signal in Hz	
fs	Sampling Frequency	
kf	Ramp rate of the Chirp	
Radserver_Mult	Timing is given every "Mult" Chirps	

The stored calibration matrix is required to ensure correct MIMO beamforming, in order to account for the fixed phase deviations which can be expected to occur due to the manufacturing variations. This must be multiplied across the virtual phase array prior to beamforming.

The radarlog data is real and is stored within the HDF5 file as datasets labeled Chn1, Chn2,...,Chn16, for the receive channels Rx1, Rx2,..., Rx16 respectively. Each of these datasets are stored as a 2-dimensional matrix, where the 1st dimension corresponds to the number of chirps in the dataset and the 2nd dimension corresponds to the number of fast time samples for each chirp.

Also stored is a "ChnTime" dataset. This stores the epoch time of a given chirp. Note time for all chirps is not given, but for every "Radserver_Mult" chirps.

5.2 ZED Stereo Camera Data

From the ZED stereo camera there will be two data files available as follows:

test_zed.svo.dat. Store the counter timestamp values as int64 data types. Time stamping is made on a video frame basis

Video.mp4: Video file containing the left camera image from ZED stereo camera

5.3 Velodyne Data

Two files are stored from the Velodyne. These are as follows:

test_velodyne.dat.

Stores the point cloud data. The data structure within the file conforms to the Velodyne datagram description found in the VLP-16 manual, consisting of 1206 bytes per datagram (this is the payload length, the UDP header is not stored).

The datagrams contain the UTC timestamp (derived from the GNSS input) measured in microseconds since the top of the UTC hour (as per the description in the manual). Additionally, the hardware performance counter timestamp is appended to the start of each datagram, constituting the first 8 bytes as type int64. After this addition, the total datagram size becomes 1214 bytes.

test_gps.dat

Stores the sensor positioning data from the GNSS module, also conforming to the data structure within the VLP-16 manual. This data also contains the NMEA navigation strings to allow extraction of absolute UTC time for use with the datagram UTC top of the hour timestamp.

5.4 Mira GPS data

This holds the GPS data of the location and velocity of the UoB vehicle holding the Radarlog.

5.5 Time Synchronization data

This is binary file that holds the time delay information between the PC operating the radarlog and the operating the combination of the ZED plus Velodyne.

6. MATLAB Sample Codes

Minimal MATLAB codes for signal processing are described below.

6.1 Reshaping Data for MIMO mode

For MIMO processing it is useful to have data in matrix with following dimensions.

- Dimension1 – Number of Channels
- Dimension2 – Number of Virtual MIMO channels
- Dimension3 – Number of MIMO frames

To re-shape the raw data into a MIMO format the following MATLAB code sample is useful, where the number of transmitters is specified as nTx=4.

```
%Some parameters
nTx      = 4;
nSamples = size(rawData,1);
nRx      = size(rawData,2);
nMimoFrames = size(rawData,3)/nTx;

%Reshape
rxData = zeros(nSamples, nTx*nRx, nMimoFrames);
for iFrame = 1:nMimoFrames
    iStart      = iFrame*nTx - nTx + 1;
    iEnd        = iFrame*nTx;
    rxData(:, :, iFrame) = reshapeDataForMimo(rawData(:, :, iStart:iEnd));
end
```

6.2 Selecting $\lambda/2$ spaced Virtual Channels

The MIMO virtual channels formed by the Inras radarbook will have three overlapping channels. For beamforming by FFT it is required to have all virtual channels at $\lambda/2$ spacing with no overlap. So, select only the set of channels where there is no overlap, maintaining the $\lambda/2$ spacing. This is done with the code sample below.

```
%Index of required virtual Rx (for lambda/2 spacing)
virtRxIdx = [1:15, 17:31, 33:47, 49:64];

%Pick required virtual elements
rxData = rxData(:, virtRxIdx, :);
```

6.3 Reading .h5 file from Radarlog

The following lines of code can be used as a reference to read a single frame from Radarlog.

```
% iFrame is MIMO frame number to read
% nSamples is the number of fast time samples
% h5Filename is the filename with the radarlog data

nChs          = 16;
nChirpsPerFrame = 4;
iChirpStart   = ((iFrame-1)* nChirpsPerFrame) + 1;
dataStartIdx  = [1 iChirpStart];
dataCountIdx  = [nSamples nChirpsPerFrame];

%get data for required chirp
output = zeros(nSamples, nChs, nChirpsPerFrame);
for iCh = 1: nChs
    tmpStr = sprintf('/Chn%d', iCh);
    output(:, iCh, :) = double(h5read(h5Filename, tmpStr, dataStartIdx, ...,
                                     dataCountIdx));
end
```

6.4 Radarlog Data Antenna Calibration

The radarlog calibration data is extracted as follows from the H5 file. It is to be noted that during the trials, Radarlog was operated in MIMO configuration. Therefore, calibration file contains 64 elements, corresponding to each virtual antenna element of the radar.

```
%Get the cal data - these will be read only to
calRe  = typecast(h5readatt(h5FileName, '/', 'CalRe'), 'double');
calIm  = typecast(h5readatt(h5FileName, '/', 'CalIm'), 'double');
calData = complex(calRe, calIm);
```

6.5 Time Synchronization

The sample code to synchronise data from all the sensors is attached in the repository.