Cognitive Real-Time System for Autonomous Vehicles (104268-CORTEX)

Data Structure Description

Project	CORTEX
Document Title:	UoB Collected Data Structure
Document Reference	104268 – UoBDataStruct1
Issue	Rev A
Date:	30/03/20

Version History

Version	Date	Incorporated by	Comments
A	30/03/20	Sukhjit Pooni Scott Cassidy	First Release

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

Prof. Marina Gashinova

Date: TBD

1 Introduction

This report describes the structure of the data collected from the instrumentation, used for the CORTEX project, by the project team at the University of Birmingham.

2 INRAS Radarbook Data

The Inras radarbook is a MIMO capable radar system with 4 transmit elements and 8 receive elements. The raw data from this radar is collected using MATLAB and is stored as a MATLAB ".mat" file.

2.1 Stored Data Variables

The structure of the important variables in this file are described within this section, with the variable name given in brackets.

NOTE: Only those variables used in the signal processing are described. There are other variables in the file that are used for checks during testing, but these are not used during processing and so will not be described here.

2.1.1 Inras Collected ADC Data (rawData)

This is the raw (uncalibrated) ADC data collected from the radarbook stored as a matrix with the following dimensions. Note that this data is real and NOT complex.

- Dimension1 – Number of samples
- Dimension2 – Number of physical receive channels
- Dimension3 – Number of chirps

2.1.2 Read Sampling Frequency (fsRead)

This is the actual sampling frequency as was used by the radarbook in the collection of rawData. It's unit is in Hz.

2.1.3 FMCW Frequency Bandwidth (fBandwidth)

The specified bandwidth of the FMCW chirp signal in Hz.

2.1.4 FMCW Centre Frequency (fCentre)

The specified centre frequency of the FMCW signal in Hz

2.1.5 FMCW Ramp up time (tRampUp)

The specified ramp-up time of the FMCW chirp signal in secs.

2.1.6 FMCW Chirp interval time (tChirpInt)

The specified time interval between chirps of the same MIMO frame in secs

2.1.7 FMCW MIMO frame interval time (tMIMOFrameInt)

The specified time interval between MIMO frames in secs

2.1.8 Inras Calibration Data (CalData)

This gives the calibration data that is read from the Inras radarbook. It has the following sizes depending on whether MIMO mode was used or not.

NON-MIMO mode
Array of 8 complex values for each of the physical Rx elements as follows

1	RX1
2	RX2
3	RX3
4	RX4
5	RX5
6	RX6
7	RX7
8	RX8

MIMO mode
Array of 32complex values for each of the virtual elements as follows

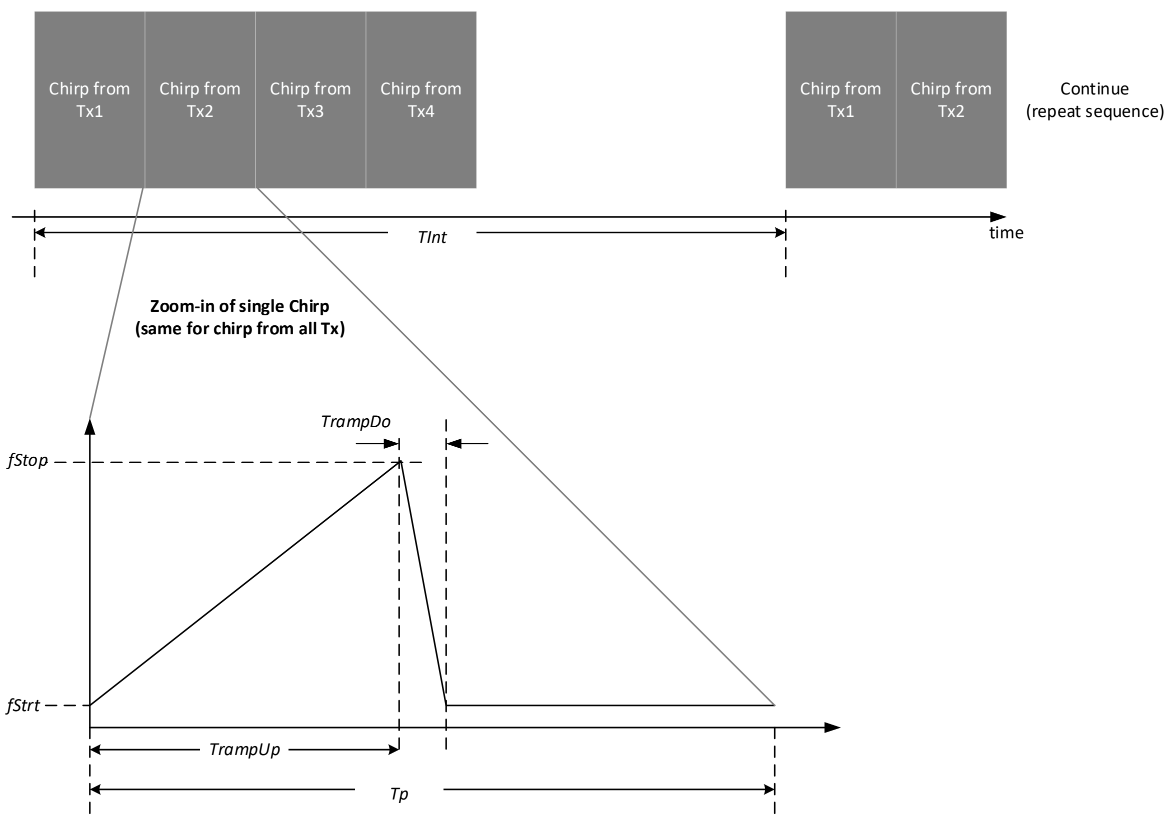
1	TX1 to RX1
2	TX1 to RX2
:	:
8	TX1 to RX8
9	TX2 to RX1
10	TX3 to RX2
:	:
32	TX4 to RX8

2.1.9 Inras Configuration (Cfg)

This gives the configuration of the Inras radar, used to collect the data within the file. This parameter is store as a struct with the following fields. Most of the values are either calculated from the FMCW parameters described previously or are set explicitly.

- fStrt: Start frequency of the FMCW signal in Hz
- fStop: Stop frequency of the FMCW signal in Hz
- TrampUp: Ramp-up time of the FMCW signal in sec
- TrampDo: Ramp-down time of the FMCW signal in secs
- TInt: Frame interval time (for MIMO mode – one MIMO frame) in secs
- Tp: Chirp interval time in secs
- N: Requested number of samples to be read per chirp
- NrFrams: Is not used by code
- TxSeq: TDMA Ordering of Tx Sequence in MIMO mode

Some of these parameters are best shown by the diagram below



2.2 Guide to using Data for MIMO Processing

2.2.1 Re-shaping Raw Data to MIMO format

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
```

2.2.2 Selecting Virtual Channels that are only $\lambda/2$ spacing

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 = [1:7, 9:15, 17:23, 25:32];

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

After this selection the rxData can then be phase calibrated by the relevant MIMO channels in CalData.

3 NXP Dolphin Data Format

The NXP Dolphin is a MIMO radar with 3 transmitters and 4 receivers. However, for CORTEX, we configure the radar with a single active transmitter and four active receivers in order to maximise Doppler performance. Beamforming is therefore performed on receive only. Timestamping is performed when data is received by the computer so there is some degree of error due to latency.

Camera data is supplied as an H.264 encoded video file in an MP4 container and is timestamped to allow for co-registration.

3.1 NXP Dolphin Parameters

For clarity, parameters will be defined as follows:

- **Chirp duration:** active duration of chirp during which time data is recorded
- **Pulse repetition interval (PRI):** time from the start of one chirp until the start of the next chirp (i.e. includes both active and non-active period). This interval determines maximum unambiguous Doppler value.
- **Chirps per Doppler interval:** number of individual chirps which are incorporated into a single coherent Doppler processing interval. This is equal to the number of Doppler bins after Doppler DFT.
- **Doppler processing interval:** active period during which a number of chirps are gathered. This interval determines Doppler resolution.
- **Doppler repetition interval:** time from the start of one Doppler interval to the start of the next Doppler interval. This includes both the Doppler Processing interval itself, and an inactive period between each Doppler interval during which data is still being transferred to the computer.
- **Doppler overlap factor:** Ratio between the Doppler processing interval and the Doppler repetition interval.

Operating parameters used for basement measurement are as follows:

Parameter	NXP Dolphin	Units
Bandwidth	2.0	GHz
Chirp duration	25.6	µs
PRI	86.6	µs
Chirps per Doppler interval	128	-
Doppler processing interval	11.1	ms
Doppler repetition interval	16	ms
Doppler overlap factor	0.69	-
Range resolution	7.5	cm
Max unambiguous range	19.2	m
Velocity resolution	0.174	ms^{-1}
Max unambiguous velocity (RX-BF)	±11.1	ms^{-1}

Operating parameters used for car park measurement are as follows:

Parameter	NXP Dolphin	Units
Bandwidth	0.5	GHz
Chirp duration	25.6	µs
PRI	86.6	µs
Chirps per Doppler interval	128	-
Doppler processing interval	11.1	ms
Doppler repetition interval	16	ms
Doppler overlap factor	0.69	-
Range resolution	30.0	cm
Max unambiguous range	76.8	m
Velocity resolution	0.174	ms^{-1}
Max unambiguous velocity (RX-BF)	±11.1	ms^{-1}

3.2 Calibration

Correct MIMO beamforming requires the application of calibration data in order to account for the fixed phase deviations which can be expected to occur due to manufacturing variations. This calibration data is based on our measured data and is provided in the form of a complex exponential which must be multiplied across the virtual phase array prior to beamforming (i.e. second matrix dimension, as described above).

Calibration data is supplied in ".mat" format, containing:

- nxp_cal: 4 element calibration for use in non-MIMO mode
- nxp_cal_mimo: 12 element calibration for use in MIMO mode (not required)

3.3 Data Files

Data for each measurement is supplied in the following files:

- **test_nxp.mat**, MAT file containing:
  - **nxpOutput:** radar data, structure described in following sections
  - **nxpTime:** timestamp, in microseconds, when each chirp is received by the computer (i.e. not entirely accurate due to latency)
- **video.mp4**, video file containing left camera image from stereo camera
- **video_timestamp.mat**, MAT file containing timestamp, in microseconds, when each video frame is received by the computer (i.e. not entirely accurate due to latency)

3.4 Frame Structure

Radar data is stored as a four-dimensional real *int16* array with the following dimensions:

- 512 time samples
- 4 receivers
- 128 chirps per frame
- n frames

In non-MIMO mode, the data is already correctly structured by dimension. Fourier processing may be performed and no reshaping of the data matrix is required.

3.5 Receive beamforming mode (non-MIMO)

In this mode, all chirps are performed with illumination by TX1 only. Transmitters TX2 and TX3 are always inactive. Therefore, this mode provides a four element λ/2 spaced receive array without any additional processing. Basic process is as follows:

1. Perform real-to-complex FFT across the first dimension for range. Since one half of the real DFT is redundant it can be discarded.
2. Perform complex-to-complex FFT across the third dimension for Doppler.
3. Apply calibration (discussed previously).
4. Perform complex-to-complex FFT across the second dimension for beamforming.

3.6 TD-MIMO beamforming mode (TD-MIMO)

In this mode, chirps are performed with illumination by a single transmitter at a time. The active transmitter, each separated by 2λ, cycles from chirp-to-chirp: TX1-TX2-TX3. Therefore, this mode provides a twelve element λ/2 spaced receive array, but a simple additional processing step must be performed in order to reshape the matrix prior to beamforming. If the number of chirps in a Doppler processing interval is not divisible by three, the remaining chirps should be discarded. Minimal MATLAB code is provided below:

```
%Some parameters
nMimoFrames = floor(size(nxpOutput, 3) / 3);
newSize = size(nxpOutput);
newSize(2:3) = [newSize(2) * 3, nMimoFrames];

%Reshape
nxpOutput = reshape(nxpOutput(:, :, 1:(nMimoFrames * 3), :), nxpSize);
```

Different calibration data is supplied for the TD-MIMO case. The basic processing should otherwise be performed as for the non-MIMO case.