Mission Research Corporation

RADAR STIMULATOR INTERFACE PROTOCOL, PRELIMINARY INTERFACE DESIGN AND DOPPLER BEAM SHARPENING IMPLEMENTATION

DISCLAIMER

Charles H. Aldrich

June 1993

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Prepared for:

Los Alamos National Laboratory

Los Alamos, New Mexico

Under Subcontract No.:

9-XF9-7334W-1 / Task Order 11

Prepared by:

MISSION RESEARCH CORPORATION

127 Eastgate

Los Alamos, New Mexico

(505)662-0070

MASTER

1

The following pages comprise the final report for Subcontract Number 9-XF9-7344W-, Tack Order 11. They describe implementation of the Doppler Beam Sharpening model on the Cray supercomputers, the interface protocol between the stimulator prototype and the Edward's Air Force Base Radar and a preliminary study of the stimulator interface for the final specifications.

DBS Mode Implementation

We have added an implementation of the DBS mode to our radar stimulation code (RBGM mode) running on the Cray super computer. The radar processing part of the code is reasonably close to that in fielded aircraft radar systems. Most of the details of the code (generating the radar return, etc.) are the same reported in previous progress reports. In this report we will only include details which differ from the RBGM code or are important factors in understanding the current DBS implementation.

In RBGM mode all targets are blurred on the radar screen by the width of the antenna pattern which we have chosen as 3.0 degrees. DBS mode sharpens this blurring to a small fraction of the antenna pattern ($1/N_{\text{sharpening}}$). In this implementation we have chosen two sharpening values $N_{\text{sharpening}} = 8$ and 64, consistent with the APG-68.

To simplify the implementation we assume a constant ownship velocity, a single look and no azimuth presum processing.

Details

To achieve this sharpening DBS mode the radar processes return signals differently than in the RBGM mode. In RBGM mode the phase of the return signal is ignored and processed incoherently a single return at a time. In DBS mode, however the returns are processed in blocks. Figure xxx diagrammatically shows a typical antenna scan. Between the beginning and end of bar the transmitted pulses are collected in coherent processing intervals of N_{coherent} returns. Processing of the returns is done in batches with one being processed while the next is being collected. Each

range bin in the returns is processed with that range bin for all other returns in a coherent manner described below. The resulting sharpened image is then displayed similar to the RBGM image.

We use a gaussian shape for the radiated antenna power pattern $\exp(-\Theta^2 \ln(\frac{1}{2})/\Theta_{hw}^2)$, where Θ is the angle from the antenna center and Θ_{hw} is the half width at half maximum power point. The phases of all targets included in the return calculation are calculated and summed coherently in each range bin allowing targets within the antenna pattern in each range bin to constructively and destructively interfere with each other. This interference pattern will vary in time as the antenna sweeps over a coherent processing interval since targets with slightly different azimuths will have different doppler frequencies.

Figure xxx shows an example of how the doppler frequency varies across the antenna footprint. For small antenna footprints this varies approximately linear with angle. This information is used to set the Pulse Repetition Frequency (PRF). The PRF determines the maximum doppler frequency that can be unambiguously resolved. If the PRF is set too low only part of the antenna footprint will be sampled unambiguously. In figure xxx the PRF is set to sample dopplers within the full width of the gaussian at half maximum. Any targets outside of the full width of the gaussian will have higher frequencies than can be properly sampled and will alias to lower frequencies. If the PRF is set to high time is wasted sampling targets way out on the side lobes of the antenna pattern where there is very little return power. For this implementation we set the PRF to sample twice the full width at half maximum of the gaussian beam .

$$PRF = f_{doppler}(\Theta_{bore} + 2\Theta_{hw}) - f_{doppler}(\Theta_{bore} - 2\Theta_{hw})$$

The extent that DBS processing can sharpen the display depends on the number of pulses included in a coherent processing interval ($N_{coherent}$). With $N_{coherent}$ samples we can determine the doppler frequency with a resolution of $(PRF)/N_{coherent}$. Since the PRF is set to sample

twice the full width of the gaussian to sharpen the beam by a factor $N_{sharpening}$ requires twice as many samples so we set.

$$N_{coherent} = 2N_{sharpening}$$

The last variable to set is the antenna scan rate. This needs to be set to a value that adequately samples each target in the antenna footprint during a coherent dwell. If the scan rate is too high targets will not be illuminated by the beam long enough to be sampled. If the scan rate is too slow the antenna will scan through too small a fraction of the antenna footprint, wasting time. In this implementation the antenna scans through one full width at half maximum of the gaussian beam in one coherent dwell, or.

$$\dot{\Theta} = \frac{\Theta_{hw}}{N_{coherent}} \times PRF$$

To sample the different doppler frequencies in each range bin corresponding to different targets we use a correlation technique. To do this phase histories are accumulated for each antenna position. For constant ownship velocity the doppler frequency at the center of the antenna footprint is constant in time. A series of $N_{coherent}$ history values are calculated for each antenna position in the coherent processing interval from the following formula, where $f_{doppler}$ is the doppler

$$\Psi_{history}(t_{return}) = \exp(2\pi i f_{doppler} t_{return})$$

frequency at the center of the antenna footprint.

For each range bin a series of $N_{coherent}$ received signals $\Psi_{received}$ are stored and preprocessed in a batch. These signals are correlated against each of the phase histories for each antenna position. The correlations are performed by performing a summation over pulses of the product of the

phase histories and the stored received signals, as follows,. Signals from targets near the center of

$$\Psi_{processed}(t_{pulse}) = \sum_{t_{pulse}} \phi_{history}(t_{pulse}) \Psi_{received}(t_{pulse})$$

each antenna position will correlate well with the time history for that antenna position producing a large sum, while targets further from the antenna center will correlate less well and produce smaller sums. This results in a series of $N_{coherent}$ values, $\Psi_{processed}$, one for each antenna position which represents the sharpened image which we display in the same way the RBGM image was displayed.

Coherent Processing Interval for DBS

Example of Doppler

Doppler Aliasing

Variable Doppler Implies Variable PRF

Synchronizing the Prototype Stimulator with the Radar Under Test

To synchronize the prototype radar stimulator to the radar under test (RUT) requires that the stimulator hardware have advance knowledge of what the radar hardware is doing. Edwards is developing a device (Edwards' PC) to eavesdrop on the Avionics D-Mux 1553 bus on the RUT to supply this information. Figure xxx shows a diagrammatic representation of how this device will interface to the prototype stimulator.

Approximately 50 times a second data is available on the 1553 bus reporting the current antenna azimuth and elevation. This data is passed unchanged over a serial connection to the hardware controller. The hardware controller uses this signal to determine when the beginning of a scan occurs and resets the frame buffer at the correct time. This data is also passed on to the compute server. By a process of interpolation the compute server will be able to use this data to calculate the scan rate and position of the antenna at any time during the antenna scan.

Whenever the pilot commands a mode change from the MFD a command is sent over the 1553 bus. Edward's PC will intercept this message. Along with current cursor position on the MFD and other data available on the 1553 bus Edwards' PC will calculate the minimum and maximum range and the minimum and maximum azimuth angles that the radar will illuminate during the next scan and send them to the hardware controller. This data is passed on to the compute server. On receipt of mode change the hardware controller will send a reset signal to the Frame Buffer clearing its memory in readiness for the computed signals for the new mode.

The hardware controller is also responsible for synchronizing the Frame Buffer. The compute server calculates the returns for each scan of the antenna and transfers it to the Frame Buffer memory over a HIPPI bus. The I/Q data is computed and stored as a series of records one for each transmitted pulse from the radar antenna. When the transmits a pulse the Xmit Sync signal is generated. When the Frame Buffer receives this signal it waits for a programmable amount of time (chosen consistent with the radars minimum display range) then starts sending out a set of analog I/Q signals corresponding to one record stored in the Frame Buffers memory. The digital data is

clocked out of memory based on an internal clock with a programmable repetition rate (chosen consistent with the expected radar range resolution.) This data is sent to digital to analog converters and sent out to the remainder of the radar stimulator. This is done for each transmitted pulse.

The Frame Buffer is built with two memory buffers. This allows the Frame Buffer to be sending the I/Q data while the compute server is calculating and transferring the I/Q data for future returns.

The Frame Buffer has an additional control signal that is generated by the hardware controller. This is the pulse hold off mask. It is used to prevent the Frame Buffer from responding to the Xmit Sync signal and clocking out I/Q signals until the hardware controller determines that the current antenna scan is at an antenna position where valid data is present in the Frame Buffer. The hardware controller knows the minimum and maximum azimuth angles for the antenna scan in the current mode. It also knows the current antenna position at any time. This allows it to determine if the antenna is within the azimuth limits of the next scan and whether it is retracing for the scan or turning around. Between scans the hardware controller will use the hold off mask to prevent sending data to the radar until the next scan begins.

Figure xxx shows how the hold off mask might be used in a typical case. Here we have set up the Frame Buffer to hold data for 128 returns. The radar mode, however, requires that we send 320 returns during an antenna scan. Thus one scan of the antenna will require sending data from two and one half buffers. To begin the stimulation the compute server will calculate and load the first 128 pulses into buffer #1. As the Frame Buffer plays out the first 128 pulses from buffer #1 the compute server will calculate and load the second 128 pulses into buffer #2. When the Frame Buffer finishes playing out the first 128 pulses (buffer #1) it will automatically switch to other buffer (#2.) At this point the compute server will compute the last 64 pulses in the current scan and the first 64 pulses in the next scan and load them into the frame buffer. When the Frame Buffer plays out the second set of 128 pulses it will switch back to the buffer #1 and play out the last 64 pulses of the scan. At this point the hardware controller will assert the hold off mask signal

and the Frame Buffer will ignore any transmitted pulses coming from the radar until the first valid pulse of the next scan. Just before this pulse arrives the hardware controller will release the hold off mask signal and let the Frame Buffer play out the first 128 pulses of the second scan. While this is happening the compute server continues to calculate pulses ahead of the Frame Buffer and the hardware controller will assert the hold off mask signal as necessary to keep the computation in synch with the antenna scan.

During a change in radar mode where the radar signal has to be recalculated the hardware controller will reset the Frame Buffer memory and send data to the compute server asking it to calculate data for the next mode. It will assert the pulse hold off mask until valid data has been calculated by the server and loaded into the Frame Buffer. For some modes this may result in a blank display on the radar display for one antenna scan.

Encoding the Messages from the 1553 Bus.

Two types of messages are transferred from Edward's PC to the hardware controller. One message is sent every time the radar is commanded to change modes giving the hardware controller the information it needs to update the calculation of the radar returns. The second message is sent approximately 50 times a second reporting the current antenna position. The structure of these messages is shown in figure xxx. Both messages begin with a single word which includes a byte uniquely identifying the data contained in the message. The most significant byte contains the message id, (0 for the Current Antenna Position and 1 for the Current Mode). The least significant byte contains a number from 0 to 255 representing the message number. As each message is sent this number will be increased by one wrapping back to zero when 256 is reached. This allows the receiving hardware controller to check if it has received all the messages in the proper order.

The Current Antenna Position message consists of the Message Id word plus two data words for a total of three words (6 bytes.) The first data word contains the current antenna azimuth. The second data word contains the current antenna elevation. Both of these angles are encoded as a

signed 16 bit integer with the least significant bit representing $180 / 2^{15}$ degrees or approximately 0.0055 degrees.

The Current Mode message consists of the Message Id word plus five data words for a total of six words(12 bytes.) The first data word contains the current radar mode (0 for SPAM, 1 for Expanded, 2 for DBS1, 3 for DBS2). The second and third data words contain the minimum and maximum azimuth angles. These two angles are encoded as signed 16 bit integers with the least significant bit representing approximately 0.0055 degrees, the same encoding used for the minimum and maximum azimuth angles in the Current Antenna Position message. The fourth and fifth data words contain the minimum and maximum range. These two ranges are encoded as unsigned 16 bit words with the least significant bit representing ten meters. This allows the minimum and maximum ranges to be specified somewhere between 0 and 655 kilometers with a resolution of ten meters.

HIPPI Hold Off Timing Diagram

Current Antenna Position

0	0	0	0	0	0	0	0	Msg7	Msg6	Msg5	Msg4	Msg3	Msg2	Msg1	Msg0	Message Id
Az15	Az14	Az13	Az12	Az11	Az10	Az9	Az8	Az7	Az6	Az5	Az4	Az3	Az2	Az1	Az0	Azimuth
El15	El14	El13	El12	El11	El10	El9	El8	El7	El6	El5	El4	El3	El2	El1	El0	Elevation

Azimuth and Elevation encoded in the range -180.0 to 180.0 degrees with a resolution of approximately 0.0055 degrees.

Current Mode

0	0	0	0	0	0	0	1	Msg7	Msg6	Msg5	Msg4	Msg3	Msg2	Msg1	Msg0	Message Id
0	0	0	0	0	0	0	0	0	0	0	0	0	0	Mod1	Mod0	Mode
AzB15	AzB14	AzB13	AzB12	AzB11	AzB10	AzB9	AzB8	AzB7	AzB6	AzB5	AzB4	AzB3	AzB2	AzB1	AzB0	Azimuth Begin
AzE15	AzE14	AzE13	AzE12	AzE11	AzE10	AzE9	AzE8	AzE7	AzE6	AzE5	AzE4	AzE3	AzE2	AzE1	AzE0	Azimuth End
RgB15	RgB14	RgB13	RgB12	RgB11	RgB10	RgB9	RgB8	RgB7	RgB6	RgB5	RgB4	RgB3	RgB2	RgB1	RgB0	Range Begin
RgE15	RgE14	RgE13	RgE12	RgE11	RgE10	RgE9	RgE8	RgE7	RgE6	RgE5	RgE4	RgE3	RgE2	RgE1	RgE0	Range End

Azimuth and Elevation encoded in the range -180.0 to 180.0 degrees with a resolution of approximately 0.0055 degrees. Range encoded in the range 0.0 to 655 kilometers with a resolution of ten meters.

Avionics D-Mux 1553

ISSUES

- 1.) How to keep PPI in synch with Radar
 - A.) Signal from Synchronizer, PRF signal.
 PRF Signal allows accurate prediction of next Pulse.
 Synchronizer signal allows us to be allways in synch, even for long periods of no Pulses.
 - B.) PRF signal.
 PRF Signal allows accurate prediction of next Pulse.
 Oscilator, Counter locked to LO may keep synchronization.
 - C.) Signal from Synchronizer. PRF obtained by measuring PPI.
 PRF changes need to be scripted vs. Pulse Number.
 - D.) Only Transmitted Pulses.
 Lose synch in dead periods, out of synch on first pulse.
- 2.) How to predict First Pulse in Look?
- 3.) How to predict Frequency changes?
- 4.) How to predict Aircraft and Antenna positions, etc. accurately?

Questions

- 1.) Squint
 - Random? YES
 - Frame Buffer Required for each Squint Position tested? How many?
- 2.) Frequency Agility
 - Deterministic? Predictable?
 - Can Frequency be scripted versus Pulse Number?
 - Synthesize from high Resolution Time Series?
- 3.) PPI synchronization
 - Is PRF command available?
 - How predictable?
 - Slowly Varying, predict next PPI from last? (DBS)
 - Sudden changes (GMTI)
 - dead period predictable? Scripted vs. Pulse Number?
 - Synch pulse from Radar Synchronizer available?
- 4.) Is first Pulse Transmitted the First Pulse Processed?
- 5.) Pulses from Non Ground Mapping Modes also Present?
 - How know which Pulses to respond to?
- 6.) Is there a signal to Identify First Pulse in Look?
 - PreLook Interrupt?
7. How far ahead is prediction needed? One or Two PPIs?
- 8.) Do we want to compensate SAR mode for non scripted acceleration, that is correct for range migration and interpolate between dwells?
- 9.) Run INU ahead and delay input to INU by small amount equivalent to a few PPI's?

Avionics Bus

Pilot Cmds

INERTIAL
NAVIGATION
UNIT (INU)

Scripted
Trajectory

CONTROL
COMPUTER

DISPLAY

KEYBOARD

TERRAIN
CLUTTER
COMPUTER

Avionics Bus

END

DATE

FILMED

12/23/93