Simulation of Real Time Multi Radar Data With a Non-Real Time Simulator

Atakan Simsek, Dr. Ahmet Murat Ozdemiray, Dr. Alper Yildirim
TUBITAK BILGEM Advanced Technologies Research Institute, Ankara, Turkey
Email: {atakan.simsek,murat.ozdemiray,alper.yildirim}@tubitak.gov.tr

***Abstract*—Radar Warning Receiver is a type of Military Airborne Radar system, which works based on a predefined scheduling algorithm that uses Radar Scan Table. In real life, these tables' time measurement unit is milliseconds (could be less than one millisecond for specific radar types). When Radar Warning Receiver System is part of a Radar Defense System, determinism and time accuracy become vital properties to have. To satisfy these properties, a real-time simulation environment is necessary. Since real-time simulator development process is challenging and time-consuming, we propose a technique which simulates real-time multi-radar data using non-real time radar simulator. Our technique achieves this by managing simulation network.**

***Keywords*–Simulation Network; Radar Simulator; Network Management**

I. INTRODUCTION

Radar Warning Receiver (RWR) systems are widely used for detecting signals from other radar systems [1][2][4]. These systems warn the pilot if there are emitters in the range of the RWR. These warnings are handled by, manually or automatically based on the usage type. Military RWR systems are more complicated than the commercial RWR systems, since the latter only use specific radar bands [5]. More sophisticated systems can classify emitter type by using signal strength, phase, waveform and other signal properties.

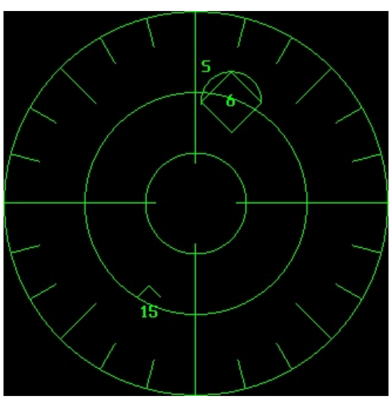

Figure 1. Threat Warning Indicator [8]

In military domain, RWR systems generally have a Threat Warning Indicator (TWI) display, depicted in Figure 1 [8], in the cockpit. Pilot uses this display to track friend or enemy emitters (threats). Airborne RWR system consists of different band antennas placed around the air platform. A receiver uses these antennas for scanning different frequency bands periodically [4].

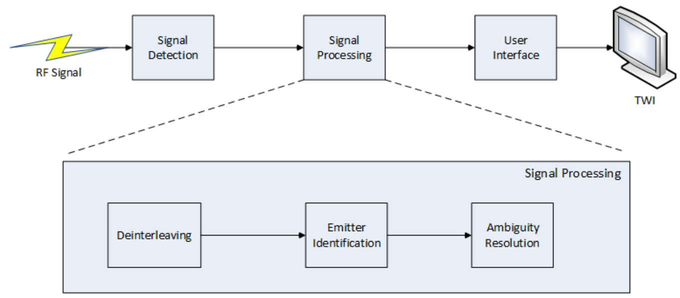

Figure 2. RWR Flow

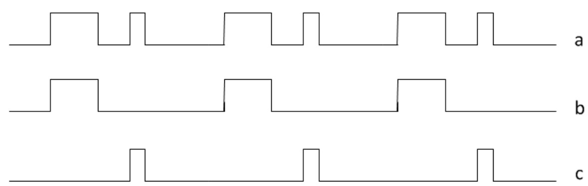

Figure 3. Deinterleaving

An RWR system can be defined by three major modules: Signal Detection, Signal Processing and User Interface. At Signal Detection Module, receivers sense the emitter signals and determine emitter parameters such as frequency, time of arrival, direction of arrival, pulse repetition frequency, pulse repetition interval, etc. These parameters are combined to define Pulse Descriptor Word (PDW). PDWs are analyzed and deinterleaved to infer emitter radar characteristics. Figure 3 'a' illustrates combined pulses of 2 emitters. Figure 3 'b' and 'c' depicts separated pulses after deinterleaving operation. Deinterleaved PDWs are used for Threat Identification process. PDWs are matched with previously known emitter records. If matching process results in only one emitter, the emitter is identified. If it results in more than one emitter, the ambiguity problem is solved by intelligent algorithms. The identified emitters are shown on the TWI and the pilot is warned with predefined icons on the display.

RWR systems can be defined as passive systems because they only receive frequency signals from emitters and do not emit any signals. RWR systems can be used for tracking the threats in order to take some counter measures [3].

The functionality of RWR Systems can be summarized as:

- Radio Frequency signal detection
- Radar type signal identification

 ISBN: 978-1-61208-482-4

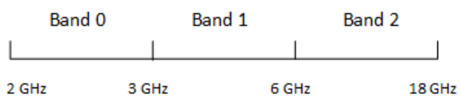

Figure 4. Signal Bands

TABLE I. SIMPLE SCAN SCHEDULE

Band ID	Dwell Time (millisecond)
0	10
1	50
2	60

- Detected signal management
- Visual and audio warning generation

The rest of this paper is organized as follows. Section II will explain real time requirements in RWR domain. Section III will present problem definition and our approach to solve this problem. Section IV will explain network flow of our solution and Section V will conclude the paper.

II. Real-Time Conditions

An RWR system needs look-up tables in order to prepare scan schedule algorithm and identify emitter types. Radar Scan Table (RST) is used for defining scan steps and Mode Table is used for defining emitter specific parameters. These tables are defined in the Mission Data File (MDF) and loaded into RWR before the operation. Therefore, RWR is a configurable system according to mission specific needs. The "Dwell Time" field of the RST represents 'sensing time of a specific target on that frequency band'. One important requirement of RWR systems is to produce enough number of PDWs in an exact Dwell time (these PDWs will be used to analyze and identify the emitter type correctly). Dwell time can be changeable according to mission requirements but its time measurement unit is milliseconds for military RWR systems. RWR management software (RWR-MS) should accomplish these dwells in millisecond time resolution. Moreover, in some cases this value may be decreased to microseconds.

In Figure 4 there are three different signal bands (2GHz - 3GHz, 3GHz - 6GHz, 6GHz - 18GHz). Basic scan schedule can be constructed as seen in Table I. A more realistic scan schedule can be seen in Table II.

It should be emphasized that, producing enough number of PDWs, which will be used for emitter identification, in exact dwell time is the most important requirement of an RWR. Therefore, RWR Management Software should be developed with a simulator or an environment, which ensures that at specific dwell time an emitter emits within a specified frequency band and the emitter should stop emitting exactly at the end of the dwell time. Non real time operating systems cannot supply this requirement, because they cannot guarantee the determinism. Radar Scan Tables are constructed based on tight scheduling and determinism assumption. Therefore, a real time operating systems should be used in RWR systems. "PDW production count in exact dwell time" requirement can be tested only by intersecting the emitter and receiver schedules in exact time accuracy. Therefore, a real time simulator should be used to test the RWR system capability. In our study, we prepared a network environment, which satisfies real time scheduling requirement and employs standard emitter simulator instead of a real time emitter simulator. It is achieved by looking the problem from a different perspective. In our solution, emitter and receiver schedules are not intersected. Instead, emitter schedule covers the receiver schedule for each dwell.

TABLE II. MORE REALISTIC SCAN SCHEDULE

Band ID	Dwell Time (millisecond)
0	5
1	25
2	30
1	10
0	5
2	15
1	15
2	15

III. Proposed Solution

In this section, we will define the problem and propose our solution. Moreover, we will discuss the concepts and components needed to understand our approach.

A. Problem Definition

In military research and development projects, commercial products usually cannot be used for security reasons. Therefore when a new project starts, its development environment also should be developed parallel with main project (especially in the projects requiring state of the art technologies). If simulators or other development environment elements are complex products, their developments may last nearly half or full time of the main project development period. At that situation, main projects cannot use these elements in development period. In our case, same situation happens and we cannot use real time radar simulator in development phase of main project. Therefore, we simulates real time radar simulator with existing simple radar simulator. In this paper, we introduce a solution which solves an industry problem by managing network components and network flow. The novel approach of this paper is setting up a complex development environment by using existing products and network components. The purpose of this solution is achieving the functionality of real time simulator. Performance comparison with other real time simulators is beyond the scope of this paper because these simulators are military specific products and their technical specifications are not publicly reachable.

B. Our Solution

The proposed system can be divided into two sub units: RWR Unit and Simulator Unit. They are deployed into different hardwares (Figure 5).

- *Threat Warning Indicator*: This component represents the graphical components of RWR. The pilot can see the emitters and send commands by using this component.

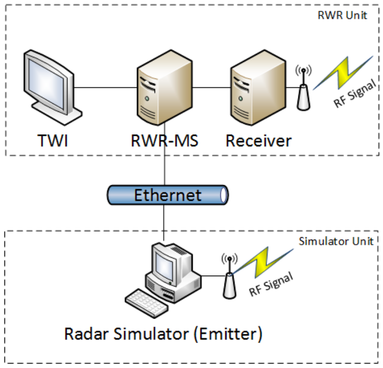

Figure 5. System Components

TABLE III. RST

RST ID	Dwell Period	Dwell Time	Freq Band Min	Freq Band Max
0	150	10	2000	3000
1	1200	50	3000	6000
2	500	60	6000	18000

- *RWR Management Software*: It is the main software which, manages RWR states and operations. It interacts with all of the RWR system to send commands and receive responses. RWR-MS gets PDW data from the Receiver and detects the emitter identity. "ScanControl" is a class which manages scan operations among TWI, Receiver and Radar Simulator. It works on a Single Board Computer. In order to supply determinism, time accuracy and other real time conditions, VxWorks 6.9 Real Time Operating System is used.
- *Receiver*: This component receives emitter signals and converts them into PDWs. It consists from the Radar Receiver Antenna and a Curtiss Wright FPGA hardware board.
- *Radar Simulator (RadSim)*: This is an existing Multi-emitter simulator. It can simulate up-to 4 different emitter simultaneously. It consists from Intel architecture single board computer and FPGA hardware unit. Radar parameters can be entered via graphical user interfaces. Radar parameters are deployed to its hardware unit via previously developed "Windows Forms Application". This unit has signal generating and emitting property. In this solution, its application was modified to communicate with RWR-MS software.

RWR-MS "MDF-Reader" class reads Mission Data File from hard disc and serves RST and Mode Tables to the related classes (Table III and Table IV):

Table III fields are as follows:

TABLE IV. MODE TABLE

Mode ID	Emit ID	Freq Min	Freq Max	PRI Min	PRI Max	PW Min	PW Max
1	1	2500	2600	5000000	5600000	1200	1400
2	1	3300	3400	4500000	4900000	1400	1700
3	2	2700	2750	7500000	7700000	3500	3550

- *Dwell Period*: RWR receiver should sense this Frequency Band at least one "Dwell Time" in this period.
- *Dwell Time*: Exact time required to scan on that specific band.
- *Freq Band Min-Max*: Frequency range minimum/maximum value.

Table IV fields are as follows:

- *Emitter ID*: This id identifies Mode belongs to which emitter.
- *Freq Min-Max*: These fields identify frequency range.
- *PRI Min-Max*: These fields identify Pulse Repetition Interval (PRI) range. PRI is the elapsed time from beginning of one pulse to beginning of the next pulse.
- *PW Min-Max*: These fields identify Pulse Width (PW) range. PW is the length of time between the rise and decay of a pulse.

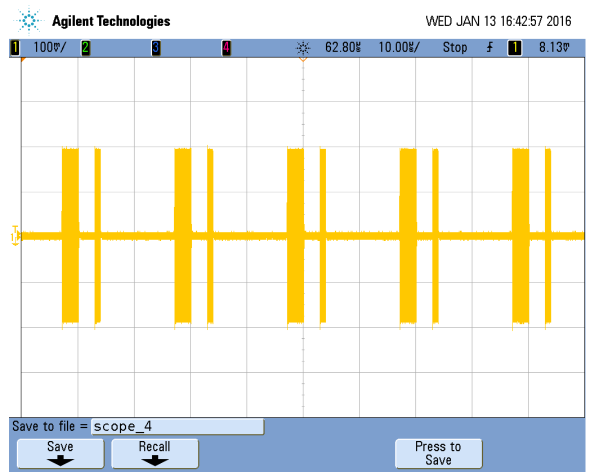

Figure 6. Radar Simulator Output (Oscilloscope Screen)

IV. Simulation Network

This section describes the network flow step by step. Figure 7 represents the data flow diagram between components and Figure 8 represents the sequence diagram. At the initialization period of the RWR-MS, ScanControl module starts as a new process and this process runs as a loop until the RWR system is shut down. The "Reader" class reads MDF file into memory; RST and Mode tables are created. At the beginning of each ScanControl loop, dwell data is read from RST and indexes of related Mode (Radar running mode) Table entries are calculated. Next, Ethernet communication class sends indexes of these modes to RadSim. Simultaneously, same MDF file is loaded into RadSim before simulation starts. Using the same configuration files allows us to send only related modes indexes instead of all data in mode table; in order to manage data traffic between RadSim and ScanControl efficiently. As an example, "1,3,-1,-1" is a valid message from ScanControl to RadSim, which represents two valid Radar modes in this Dwell with the mode indexes 1 and 3 respectively. Since RadSim can simulate up-to four different radars at the same time, -1 is inserted in the message to represent the empty radar slot.

 ISBN: 978-1-61208-482-4

TABLE V. DWELL TIME ACCURACY

Experiment Number	Simulated Radar	RST Dwell Time (millisecond)	Look Dwell Time (millisecond)	Extracted PDW Count	Correct Emitter Identification	Ambiguous Property
1	Search Radar 1	5	3	6	No	Unknown PRI Type
2	Search Radar 1	5	4	6	No	Unknown PRI Type
3	Search Radar 1	5	5	23	Yes	Correct
4	Search Radar 2	10	8	37	No	MDF Match Ambiguity
5	Search Radar 2	10	9	51	Yes	Correct
6	Search Radar 2	10	10	54	Yes	Correct

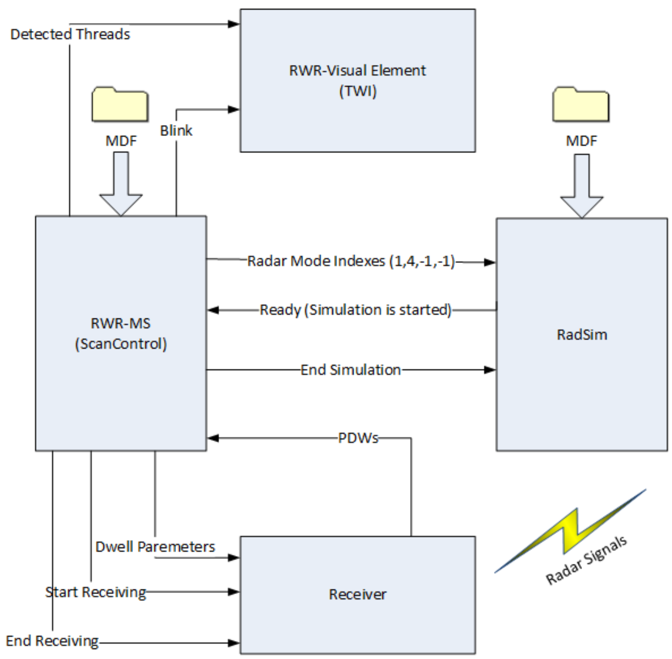

Figure 7. Network Flow

At the start of the simulation, RadSim reads Mode table in MDF and loads radar parameters. After this step is completed, RadSim starts to generate radar signals (Figure 6 shows the Oscilloscope screen of RadSim output) and “Simulation is started” message is sent to ScanContol as an acknowledgment message. At the same time, RWR-MS reads RST to adjust receiver parameters. ScanControl waits for this RadSim acknowledge message when the receiver is ready. When RadSim “Simulation is started” message is received, RWR-MS starts the timer and opens the receiver antennas for the specified dwell time period. The receiver stops when the dwell time is completed. As the sequence diagram shows in Figure 8, RWR-MS opens the Receiver antennas after RadSim starts simulation. Also, the Receiver antennas are closed after exact dwell time while RadSim continues to generate radar signal. This mechanism ensures that Radar Scan Table time accuracy constraint is exactly applied.

Subsequently, the system passes to the next dwell and same operations are repeated. Simultaneously, another process in RWR-MS reads the produced PDWs from the Receiver. These PDWs are used to identify the emitter type and other parameters by related modules. When emitters are detected, they are sent to TWI to warn the pilot (Figure 10). Total full scan time cannot exceed one second, independent from the Dwell count. When a full scan is completed; a blink message is sent to display to warn the pilot.

If we use real time simulator, there will be no interaction between RWR-MS and Simulator modules. Before simulation, both RWR and Real Time Simulator are programmed with predefined Mission Data Files. Simulator only emits data according to its MDF and RWR only sense data according to its MDF. We prepare this network environment and handshake protocol to intersect their schedule exactly for each frequency band. In our solution, we use Gigabit Ethernet and TCP for network communications.

We carried out some experiments to show importance of dwell time accuracy and to validate our simulator (Table V). In this table, our Radar Simulator simulates two different Search Radars. In the 1th and 2nd experiment, receiver should sense 5 milliseconds according to RST but for experimental purpose it senses less than 5 milliseconds between frequency band 2000-3000 MHz. In such situations, search radar frequency band and our receiver frequency band are not intersected. Therefore less than necessary PDWs are extracted from receiver module for 1th and 2nd experiment. The RWR-MS module performs emitter identification operation but the emitter is not correctly identified because of inadequate data. Same situation occurs for 4th experiment. These results show that dwell time accuracy is very important for RWR and it should be developed/tested with a precise simulator. For 3rd and 6th experiment, correct accuracy is supplied and emitter identification is done successfully. Note that, 5th experiment is correctly finished but this situation is undeterministic.

In another experiment, our simulator is tested with a more complex scenario. In this experiment, our simulator simulates two moving emitters. In the first dwell, Emitter A is emitting at frequency 3400 MHz and Emitter B is emitting at 3700 MHz. In the second dwell, our simulator changes frequencies. Emitter A’s new frequency is 2600 MHz and Emitter B’s new frequency is 2200 MHz. Figure 9 shows the timeline of this experiment. In this figure:

- T0: First dwell, simulator starts simulation
- T1: First dwell, RWR starts sensing
- T2: First dwell, RWR ends sensing
- T3: First dwell, simulator ends simulation
- T4: Second dwell, simulator starts simulation
- T5: Second dwell, RWR starts sensing

Our solution guarantees that, $T_0 < T_1$ and $T_3 > T_2$ intuitively owing to its design characteristic. For a single dwell,

 ISBN: 978-1-61208-482-4

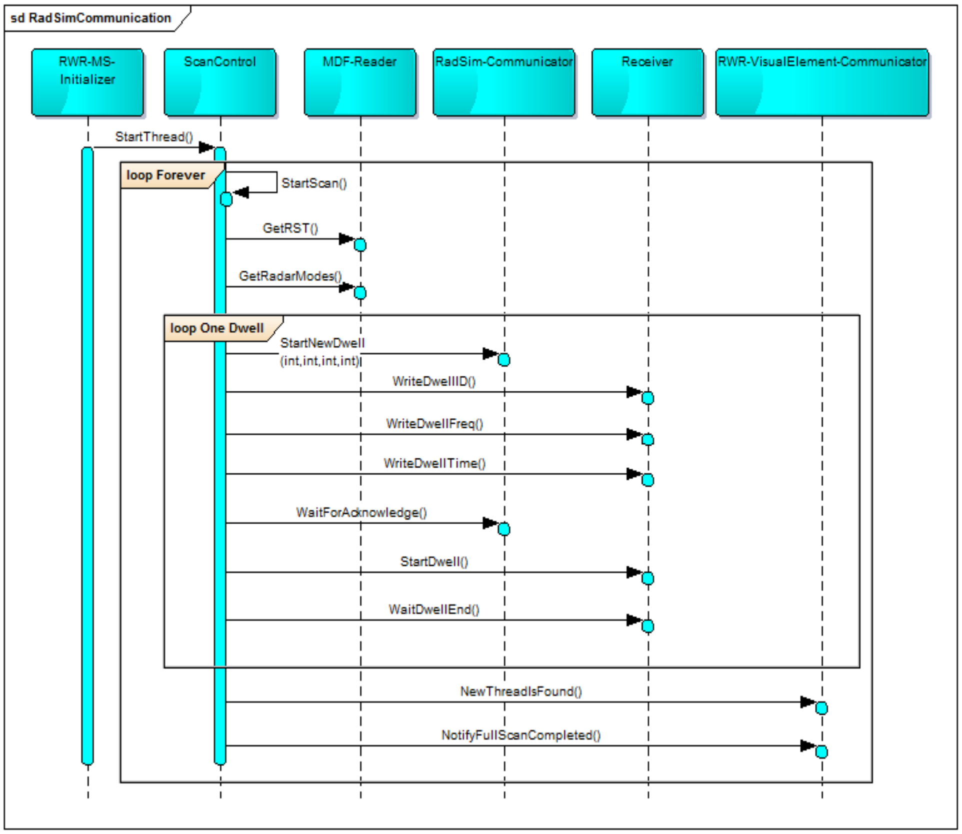

Figure 8. Flow Sequence Diagram

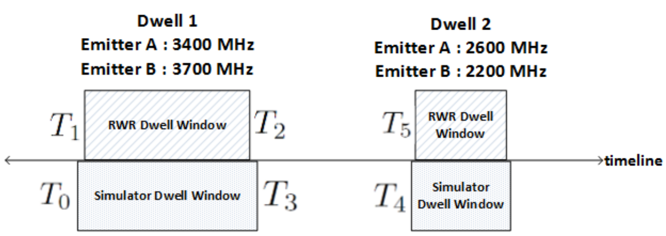

Figure 9. Simulation Timeline

emitting requirement is tested in Table V. Therefore, in this experiment we investigate its dwell switching capability. We have $(T_3 - T_2) + (T_5 - T_4)$ amount of extra latency in dwell switching time. In this experiment we find this value as 0.24 millisecond (average value for 10 runs). Main reason for this loss is TCP overhead and simulator load time. We prepare an experiment to compare Ethernet protocols efficiencies and undeterminism. Table VI shows the result of our experiment. In this experiment 19200 byte data is send from sender to receiver. This experiment repeated 10000 times to find min,max and average values. Result shows that Raw Ethernet protocol can decrease latency but we plan to use 'Discrete Wires' to decrease latency minimum.

In this solution we use an existing simulator that its emitting capability, signal count and signal strength has already been validated in previous projects. For this new configuration, only its scheduling accuracy can be arguable but we do not

TABLE VI. ETHERNET PROTOCOLS

Protocol Name	Min Time (msec)	Average Time (msec)	Max Time (msec)
Raw Ethernet	0.085	0.1271	0.1562
UDP	0.131	0.151	0.234
TCP	0.215	0.284	0.445

change its characteristics. RWR-MS sends only start simulation and end simulation commands to this simulator. However the results in the Table V verify simulator scheduling accuracy because this results match up with our theoretical knowledge in RWR systems. In this experiment, we verify that our simulator works well in the selected single frequency band. Moreover, Figure 9 shows that, our solution has a latency value at dwell switching but our design compensates this. We know this handicap and we plan to solve this issue by changing network protocol as future work.

V. Conclusion and Future Work

In this paper, we present a simulator environment to simulate radar signals. We prepared a set-up which ensures that simulator signal is ready when receiver starts to scan. At the completion of the exact dwell time, receiver stops to collect data. We have shown that receiver collects radar signal in the exact dwell time period as required for successful operation. This enables us to solve RWR real time schedule requirement and simulate in a real time like environment. In our solution,

 ISBN: 978-1-61208-482-4

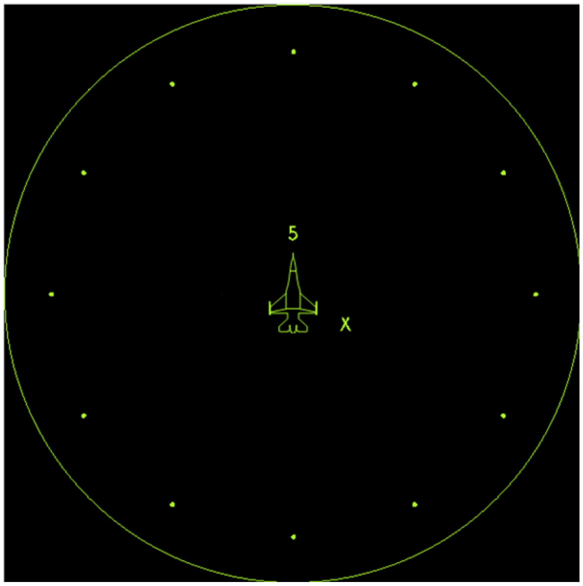

Figure 10. TWI Screen after Emitter Identification

the simulator also compensates the latency while switching is being performed between two dwells. As a future work, we plan to replace Ethernet connection with "Discrete Wires" to decrease network latency and undeterminism. We plan to change the design of RadSim to support Remote Procedure Call (RPC). We expect to decrease system jitter time and to increase determinism by using discrete wires and adding RPC functionality.

REFERENCES

[1] Peinecke, Niklas, Hans-Ullrich Doehler, and B. R. Korn, "Real-time millimeter wave radar simulation." Journal of Aerospace Information Systems 10.7 (2013): 337-347.

[2] C. B. Boettcher and R. Poster, "Object-Oriented Design Of Radar Warning Receiver Application Software", Digital Avionics Systems Conference, 1991. Proceedings., IEEE/AIAA 10th. IEEE

[3] F. Neri, "Introduction to Electronic Defense Systems", 2nd ed, MA, USA:Artech House, 2006

[4] Wikipedia.com, Radar Warning Receiver, 2015 [Online], Available:https://en.wikipedia.org/wiki/Radar_warning_receiver [retrieved 05,2016]

[5] Wikipedia.com, Radar Configurations and types, 2015 [Online], Available:https://en.wikipedia.org/wiki/Radar_configurations_and_types [retrieved 05,2016]

[6] J. W. S. Lui, "Real-Time Systems", Prentice Hall, 2000

[7] M. Dursun and O. Kizilay, "Zaman Tetikli Almac Planlayici Yazilim Bileseni Tasarimi", *UYMS*, 2014, pp.165-176

[8] Wikipedia.com, RWR Example, 2015 [Online], Available:https://en.wikipedia.org/wiki/File:Rwr_example.gif [retrieved 05,2016]