2011 International Conference on Information Management and Engineering (ICIME 2011)
IPCSIT vol. 52 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V52.32

The Construction and Management of Real-time Radar Target Recognition System

Ma Liang, Cheng Xu, Li Yongzhen Wang Xuesong

School of Electronic Science and Engineering,, National University of Defense Technology, Changsha, China

Abstract—In a radar system based on the hardware-in-the-loop simulation, not all the pieces of software must communicate with the hardware devices at each simulation moment, like the target recognition software. It runs in soft real-time. The framework and workflow of the recognition software is presented. It is developed in the Windows operating system and personal computer. The practical operations show that the system has real-time performance and it is easy to development and extension.

Keywords-recognition system; management; real-time; hardware-in-the-loop;

1. Introduction

As control software used in the modern radar system, the development of radar target recognition system is greatly different from the traditional management information system (MIS). For example, the recognition software needs much complex data processing and computation instead of data querying, modifying and deleting in traditional MIS. For simulating the weapon system as realistically as possible, some hardware devices are used to form the hardware-in-the-loop (HWIL) simulation. Because these hardware devices replace the corresponding digital model, the HWIL simulation system has higher reliability compared with the all-digital simulation method. But it also has longer development cycle and higher costs [1][2].

In HWIL simulation platform, an unavoidable problem is the data transmission between hardware and software. Generally speaking, the hardware devices are running in real-time, and their work timing is hard to change. To ensure the simulation platform run normally, the software also needs running in real-time. At present, there are two solutions. The first is for the codes of software to be programmed into CPLD or FPGA, and they will be executed as fast as possible [3][4]. The second is for the codes to be executed in a real-time operating system, such as the Vxworks, Linux, RTAI and Xenomai[5][6]. No matter which kind of solution, the developers require additional energy. And It is also the constraint on the HWIL applications.

In this paper, a target recognition system used for a HWIL simulation platform is presented. The components of a radar system based on the HWIL simulation are analyzed firstly. They are divided into 3 layers: non-real-time, soft real-time and hard real-time. And the target recognition system belongs to the soft real-time layer. Secondly, the functions, workflow and architecture of the recognition software are designed to make sure of the running time. Lastly, a certain recognition software is developed in the Windows operating system to validate the methods in this paper. It is used for an actual HWIL simulation platform, and it is able to meet the requirement of real-time.

2. Radar System based on the HWIL simulation

As one of the main sensors in weapon system, radar is used for detecting, tracking and reorganizing the lethal targets. It is very important to analyse the performance of radar and many researchers pay attention to

E-mail address: aquila_free@163.com, chenxin3358@163.com, e0061@sina.com, wxs1019@vip.sina.com

the improvement of radar simulation. Recently, the HWIL technique has been applied to this field. Fig.1 shows one way of a radar system based on the HWIL simulation. There are 8 main modules. Their functions are as follows.

Parameter compilation module: It is used to help users set all the parameters required in one simulation, like the size of lethal target and decoys, the power, carrier frequency, bandwidth, gain of the radar system and so on. After all the parameters are entered correctly, the module will compile out the states of targets and radar at each simulation moment during the whole simulation according to a certain time-step. Then the results will be sent to corresponding hardware and software. Because the compilation can be carried out before the beginning of one simulation, the parameter compilation module belongs to the non-real-time layer. And it usually runs in personal computer (PC) with friendly interface.

Antenna simulation module: It is used to simulate radiating pattern, side-lobe level, gain and bandwidth of transmitting and receiving signals. By the hardware devices, the noises, uncertainty and inconsistency of channels are ensured consistent with the actual radar system as largely as possible. Therefore this module runs in hard real-time.

Target and environment module: It is used to form the receiving signals by modulating the transmitting signals with characteristics of the targets. And the characteristics are calculated according to the positions, velocities and postures of targets compiled by the parameter compilation module. This module also calculates the attenuation on receiving signals by the atmosphere and interferer by the ground and sea clutters. It runs in hard real-time, too.

Signal procession module: It is used to simulate the signal procession in actual radar system, including A/D converting, matched filtering, envelope detecting and so on. These processions are executed in hardware device. And this module also runs in hard real-time.

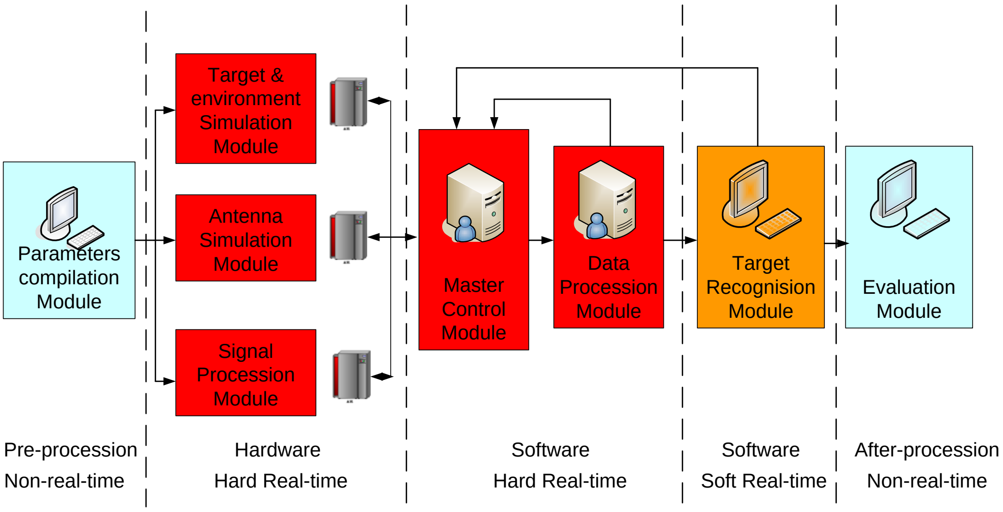


Figure 1. Components of radar system based on the HWIL simulation

Master control module: It is used to control the beam pointing and choose different transmitting signals in different tasks. The type of task is determined by the results of data procession module and the target recognition module, which may be affirming, tracking, lost-tracking, velocity measuring And the master control module should assign more resource to the high lethality targets.

Data procession module: It is used to simulate the data procession in actual radar system, including detecting and tracking. After data procession, different targets will be assigned different track numbers. Like the master control module, the data procession module is usually developed by software. But because these two modules need to communicate with the hardware devices at each simulation moment, they both run in hard real-time.

Target recognition module: It is used to extract features of targets and find out the lethal targets. The recognition results will be sent to the master control module. Unlike the results of data procession, it is utilized by the master control module every few seconds even longer. Therefore, this module runs in soft real-time, and it may be developed and executed in the Windows operating system.

Evaluation module: It is used to evaluate the outputs of one simulation, such as the distances of detection, the accuracy of tracking, the rate of recognition and so on. The needed data are saved during one simulation and the evaluation is executed after the ending of one simulation. Therefore, this module runs in non-real-time with friendly interface.

From the above, we can see that, there are 3 layers in the radar system based on the HWIL simulation: non-real-time, hard real-time and soft real-time. The software which belongs to the non-real-time layer can be developed in the Windows operating system certainly. If the software which runs in soft real-time can also be developed in the Windows, it will make the development have shorter time and lower costs.

3. Design of the target recognition system

3.1. System function

Target recognition is one of the most important parts of radar. It usually determines the performance of the whole radar system. Fig.2 shows a description of the radar target recognition. And its functions are feature extraction and recognition.

At present, an advanced radar system sends signals with different bandwidth and polarization for different tasks. For example, the narrow bandwidth signal is used for detecting and tracking targets, the middle bandwidth signal is used for distinguishing the group of targets and measuring target information such as RCS, velocity, acceleration and so on, and the wide bandwidth signal is used for imaging. For recent years, there are a lot of research achievements on feature extraction of battlefield targets by radar system. To summarize, there are six main types of radar features: kinematics feature[7], RCS feature[8], polarization feature[9],

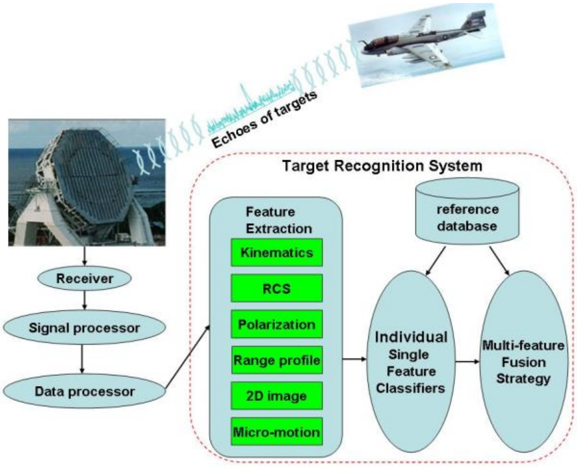


Figure 2. Description of a generic radar target recogintion system

1-dimension range profile feature[10], 2-dimension image feature[11] and micro-motion feature[12]. Each type of feature has its own superiority and inferiority in recognition. Therefore, a target recognition system should be able to extract the above six types of features.

In the field of target recognition, it is now widely accepted that combining multiple features or classifiers can provide advantages over the traditional single feature descriptor to pattern classifier design. This is based on a simple reason: decoys can imitate one or more features of the lethal target, but they cannot imitate all the features. Therefore, in the target recognition system shown in Fig.2, independent classifiers are applied in parallel to each type of features. Comparing the extracted features with the reference database, each classifier ranks the input targets according to their lethality. The multi-feature fusion strategy will be applied to take advantage of the strengths of the individual classifiers but avoid their weaknesses. And the combined decision will be sent to the master control module.

3.2. Workflow of feature extraction and recognition

Fig.3 shows the workflow of recognition system. There are 6 main steps as follows.

Step 1: Receive the tracking data calculated by the data procession module. Then assign each track an individual buffer to store the data for a period.

Step 2: If the length of data is more than a preset statistical window, the feature extraction codes will be executed. If not, wait for the tracking data at next simulation moment.

Step 3: According to the data within the statistical window, the features are calculated and averaged out. The results will be saved for the evaluation module.

Step 4: Compareing the extracted features with the reference database, each single feature classifier ranks the input targets. The more lethality a target is, the higher rank score it is.

Step 5: According to some strategies, the outputs of all individual classifiers and the historical rankings are fused to find out the lethal targets. The results are also saved for the next classification and the evaluation module.

Step 6: After one classification, the data whose locations are from head to a preset sliding window are removed from the sequence,. The length of the sliding window is shorter than the length of the statistical window.

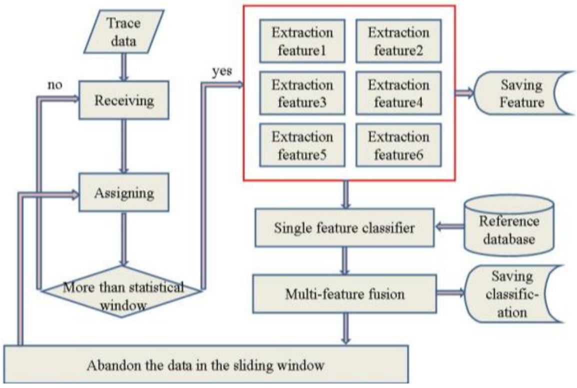


Figure 3. Processing scheme of recognition system

TABLE I. RUNNING TIME OF DIFFERENT PHASES (unit: ms)

Different phases	**Different statistical window (s)**			
	5	***10***	***15***	***20***
Receiving& Assigning	<2	3~4	<4	5~6
RCS extraction	<1	<1	<1	<1
Polarization extraction	<1	<1	<1	1~2
Kinematics extraction	2~3	6	9~10	13~15
Micro-motion extraction	1	2~3	4~5	6~8
Range porfile extraction	17~19	33~35	51~54	69~71
Image feature extraction	52~56	115~120	181~185	240~250
Single feature classification	1~2	3	4~6	7~8
Multi-feature fusion	<1	<1	<1	1
Whole running time	78~86	165~174	256~267	343~362

We can see that, the lengths of statistical window and sliding window are two important parameters in recognition system. The longer the statistical window is, the longer the data are. And the system need longer time to complete one classification. And the length of sliding window determines the output rate of classification. The longer the sliding window is, the lower the data rate is.

The running time of different phases of the recognition system is testing, shown in Table 1. The software is developed by Visual C++ 2008 and Oracal 10. The CPU of PC is Intel Core i5-760 and the clock speed is 2.80GHz, RAM is 2GB. The operating system is Windows XP 32bit. The data rate of tracking is 20Hz. It means the software should process 100 pieces of data if the statistical window is 5 seconds.

Assuming that the sliding window is 1 second, it means the output rate is 1Hz. According to Table 1, if the statistical window is 5s, the maximum target number which the recognition software can process is 11. And if the statistical window is 20s, the maximum number is only 2. Considering the operating system is non-real-time and it may respond other requests. In practice, the target number which the software can process is less than the maximum number. But a actural radar system is usually able to process several dozens input targets simultaneously. Some improvements must be made to decrease the running time.

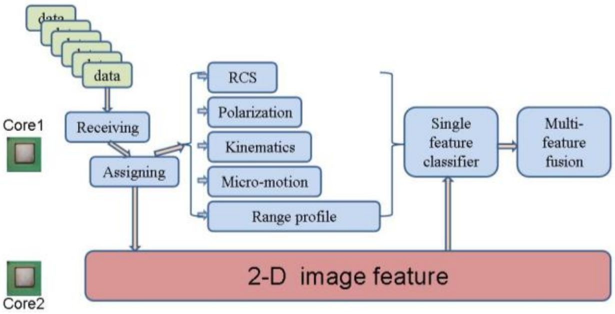


Figure 4. Flow process of feature extraction and recognition

During one simulation, the data is sent incessantly to the recognition system according to the tracking rate, and the CPU has more than one cores. Therefore, we can use the flow process technique that assigning different phases to different cores to shorten the running time of the software.

Fig.4 shows a design of flow process of the software. The phase of image feature extraction retains exclusive use of one core. And the other phases use another core inclusively. Because the running time of the image feature extraction is more than the sum of the other phases, the responding time of the software is shortened to the running time of the image feature extraction.

In addition, according to Table.1, the running time of feature extractions is much longer than other phases, and they are the bottlenecks in the real-time. But if and only if the tracking data belong to a new track, the feature extraction should calculate the whole data within the statistical window. If a track was processed before, the feature extraction can use the historical results which were saved and only need to calculate the additional data. When a target is tracked by radar continually, the running time of feature extractions is determined by the sliding window not the statistical window. If the sliding window is 1s, the running time of extractions is shortened to about 30 percent of the first column in Table. 1.

3.3. Development of recognition system

On the basis of the analysis above, the radar target recognition system is developed based on the object-oriented method. Its architecture is shown in Fig.5.

The interactivity layer is used for communicating with users and other software and hardware. The needed parameters, the expression of results and the interface protocols are defined in this layer. The main user interface is shown in Fig.6. The operational states, the feature which is extracting and the two most lethality track numbers are shown.

The management layer ensures configuration of system and documents for the software firstly. And this layer also assigns the multiple threads to different cores of CPU. It is the most important for full using the computational power and resource of PC.

The scheduling layer realizes the basic functions of the system. We design five thread classes. They manage respectively receiving, extracting, recognizing, displaying and controlling. According to the request of the management layer, these threads choose corresponding model and algorithms from the class and library layer. The scheduling layer is the kern of the software.

The class and library layer and the subclass layer are the bottom of the recognition system. Models, algorithms, database and interfaces are encapsulated into various classes. This layer is also the basis for system extension and reuse.

Some practical operations are taken as examples. The HWIL platform is developed by our group, and the structure is shown in Fig.1. The target recognition system runs in the PC with the configuration mentioned last section. Generally speaking, for almost targets, the statistical window set as 10 seconds is long enough in recognition. In the testing, the statistical window is 20 seconds, and the sliding window is 1

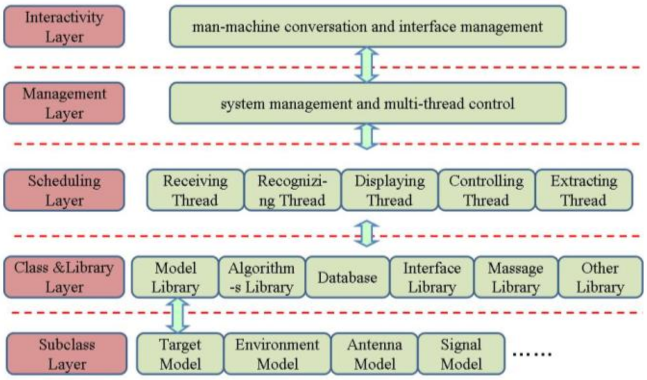
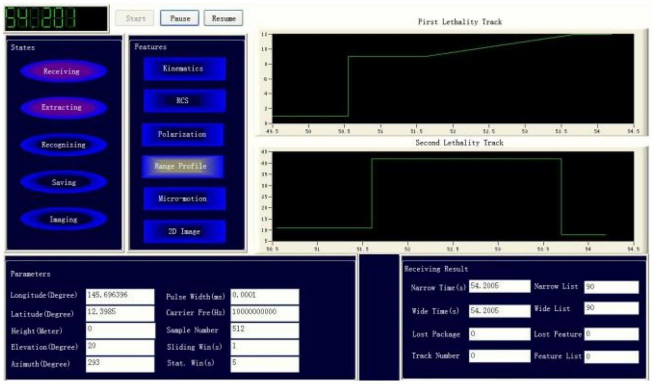


Figure 5. Architecture of the radar target recognition system



Parameters

Longitude(Degree) 145.696396
Latitude(Degree) 12.3985
Height(Meter) 0
Elevation(Degree) 20
Azimuth(Degree) 293
Pulse Width(ms) 0.0001
Carrier Fre(Hz) 10000000000
Sample Number 512
Sliding Win(s) 1
Stat. Win(s) 5

Receiving Result

Narrow Time(s) 54.2005
Narrow List 90
Wide Time(s) 54.2005
Wide List 90
Lost Package 0
Lost Feature 0
Track Number 0
Feature List 0

Figure 6. The main user interface of the radar target recognition system

seconds. The maximum track number at one time is 50. Two extracting thread objects are generated. The longest transient response time is about 5 seconds if and only if 50 new tracks are received at one time. During this period, the output rates of all the targets are unstable. Then the system can procss all the input targets within the sliding window and support the HWIL platform runs normlly.

4. Conclusion

A soft real-time radar target recognition system is presented in this paper. This software is developed in the Windows operating system. By optimizing the framework and the scheduling, the recognition system can process 50 tracks at one time to support the HWIL simulation platform. Along with the improvement of the computer, more and more complex arithmetics can be carried out by PC in a short time. By analyzing the communication in the HWIL simulation, the software that runs in soft real-time is found out. And these pieces of software can be developed easily in the Windows operating system. It will make the development of the system have lower cost and shorter time.

5. References

[1] Hui Li Steurer, M. Shi, K.L. Woodruff and S. Da Zhang, “Development of a Unified Design, Test and Research Platform for Wind Energy Systems Based on Hardware-in-the-loop Real-time Simulation,” IEEE Trans. on Industrial Electronics, vol. 53, no.4 pp.1144-1151. 2006.

[2] P. S. Shiakolas, D. Piyabongkarn, “Development of a Real-time Digital Control System with a Hardware-in-the-loop Magnetic Levitation Device for Reinforcement of Controls Educcation,” IEEE Trans. Education, vol. 46, no.1 pp.79-87. 2003.

[3] M. B. Gokhale, J. M. Stone, J. Arnold and J. Kalinowski, “Stream-oriented FPGA Computing in the Streams-C High Level Language,” IEEE Conf. Symposium on Field-Programmable Custom Computing Machines, pp. 49-56, 2000.

[4] Deming Chen Cong, J. Ercegovac and M. Zhijun Huang, “Performance-driven mapping for CPLD Architectures,” IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 22, no.10, pp.1424-1431,2003.

[5] A. Barbalace, A. Luchetta, G. Manduchi, M. Moro, A. Soppelsa and C. Taliercio, “Performance Comparison of Vxworks, Linux, RTAI and Xenomai in a Hard Real-Time Application,” IEEE Trans. Nuclear Science, vol. 55, no.1, pp. 435-439, 2008.

[6] Rahman, M. S. Atiquzzaman, M. Lvancic, W. Eddy, W.Stewart, “Implementation of SCTP in an open Source Real-time Operating System” IEEE Conf. Military Communications Conference, pp.1-7, 2008.

[7] G.Liu, H.Li, J.Li, “Moving target feature extraction with polarisation diversity in the presence of arbitrary range migration and phase errors,” IEE Proc. Radar, Sonar and Navigation, vol.147, no.4, pp.208-216, 2000.

[8] Lee S H, “Investigation of the effects of target feature variations on ballistic missile RCS,” Master thesis, Department of The Air Force Air University, 2006.

[9] Fuller D F, Terzuoli A J, Collins P J, et al., “Approach to object classification using dispersive scattering centres,” IEE Proc. Radar, Sonar and Navigation, vol.151, no.2, pp.85-90, 2004.

[10] S. P. Jacobs, et al., ”Automatic target recognition using sequencesof high resolution radar range-profiles,” IEEE Trans. Aerospace and Electronic Systems, vol.36, no.2, pp.364-381, 2000.

[11] T. Thayaparan, P. Frangos, Lj. Stankovic, et al., ”Signal processing techniques for ISAR and feature extraction,” IET Signal Processing, vol.2, no.3, pp.189-191, 2008.

[12] V.C. Chen, F. Li, S.-S. Ho and H.Wechsler, ”Micro-Doppler Effect in Radar: Phenomenon, Model, and Simulation Study,” IEEE Trans. Aerospace and Electronic Systems, vol.42, no.1, pp.2-21, 2006.