International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:7, 2009

# Radar Task Schedulers based on Multiple Queue

María I. Jiménez, Alberto Izquierdo, Juan J. Villacorta, Lara del Val, and Mariano Raboso

***Abstract*—**There are very complex communication systems, as the multifunction radar, MFAR (Multi-Function Array Radar), where functions are integrated all together, and simultaneously are performed the classic functions of tracking and surveillance, as all the functions related to the communication, countermeasures, and calibration. All these functions are divided into the tasks to execute. The task scheduler is a key element of the radar, since it does the planning and distribution of energy and time resources to be shared and used by all tasks. This paper presents schedulers based on the use of multiple queue. Several schedulers have been designed and studied, and it has been made a comparative analysis of different performed schedulers. The tests and experiments have been done by means of system software simulation. Finally a suitable set of radar characteristics has been selected to evaluate the behavior of the task scheduler working.

***Keywords*—**Queue Theory, Radar, Scheduler, Task.

## I. Introduction

RADAR system is used for detection and tracking of targets. Specifically, a multifunction radar or MFAR (Multi-Function Array Radar), is based on phased arrays, and it is able to execute multiple functions integrated all together. The main functions are: tracking, surveillance, communication, counter measures and calibration. Further analysis can be found in [1] and [2].

For this kind of radar, a wide area of coverage and high number of targets to be tracked, are key characteristics of MFAR that require both complex design and development. Analytical methodologies can't be used, so it is necessary to use simulation techniques based on Monte-Carlo method. It is discussed in [3].

Figure 1 shows the typical MFAR block diagram. A deep analysis of these blocks can be found in [4]-[6]:

- Transmitter/Receiver: it generates the signal to be sent and also receives the reflected signals.
- Beamformer: It makes spatial signal processing, and generating beams for the antennas that are electronically positioned.
- Signal processing: It makes the signal processing relative to detection, correlation and filtering.
- Tasks scheduler: Resource allocation for each task.

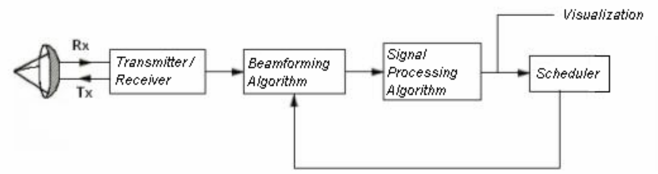


Fig. 1 Scheme of multifunction radar

Main tasks schedulers can be classified in four groups, on the basis of the used strategies for tasks scheduling, as it can be seen in [7]-[11]:

- Task interleaving: Every task is planned and mixed with others to be executed in order to minimize the radar time used.
- Queue scheduling: The tasks are planned in multiple queue of tasks that are executed in time intervals.
- Pre-emptive of tasks: The planning of tasks is previously done, but it is allowed urgent tasks, (which are unexpected), to be executed first.
- Pre-planned time templates: The tasks are planned depending on a set of fixed templates to use every radar time interval.

This paper focuses on schedulers that use task queues specifically those that use two different queues: priority and normal. It uses one queue for priority tasks and the other one for the tasks that are not so urgent.

The scheduler selects, from the input list of tasks, urgent ones and put them into the priority queue, assigning the others to the normal queue.

The scheduler first takes the tasks from the priority queue, and when this queue is empty, tasks from the other one. Once the tasks are classified, next phase deals with task arrangement along the time interval.

There is not a previous study about the parameters that influence on the behaviour of schedulers based on multiple queue. This work presents a complete analysis about it and the results and conclusions to consider for these schedulers.

The following sections describe the used methodology, the results obtained, the comparative analysis, and finally, the conclusions.

Alberto Izquierdo is with University of Valladolid, Spain.
e-mail: alberto.izquierdo@tel.uva.es

## II. METHODOLOGY

A set of families of schedulers with multiple queue are going to be analyzed. There have been considered all the parameters associated to the tasks that the scheduler manages.

The radar time is divided into time intervals called scheduling intervals, assigning to them a reference value of 100 milliseconds. This idea is proposed in [12].

The elapsed time for simulation that has been chosen is 1000 seconds. As it is used in Monte-Carlo method, a value of 100 executions is chosen for each experiment. A further analysis can be seen in [13].

A scenario with 200 targets uniformly distributed has been selected. The scanned air space is divided in sectors with range up to 70 km, variation in azimuth of 120º, and 60º for elevation.

The parameters that have been chosen for the radar implementation are detection probability $P_D = 0.99$, and false alarm probability $P_{FA} = 10^{-4}$. The air space is scanned every 20 seconds and the targets are tracked every second.

For the modelling and definition of the tasks scheduler, the following time parameters have been considered:

- Time of the beginning of the task.
- Deadline time of the task.
- Length of the task or duration of its execution.

The functions of the radar that are implemented have been divided into tasks. The types of task that have been used are:

- Surveillance.
- Confirmation of false alarms.
- Target tracking.
- Backscanning to find a lost target.
- Reacquisition when there is not success after the execution of a backscanning task.

All the tasks that must be executed immediately are considered urgent, and they are called so. It can occur when there are false alarms or when a target under tracking is not found. The types of tasks considered urgent are: confirmation, backscanning and reacquisition.

The task schedulers that have been studied along this work use two types of queue for planning every scheduling interval. The queues that the scheduler manages are:

- Priority queue: this queue includes all the tasks considered as priority.
- Normal queue: this one includes all the tasks that are not considered priority.

These schedulers use two queues, but the classification of them is based on the policy that is applied to make the priority queue. According to the specific scheduler, the priority queue consists of different number and type of tasks. This can be seen in [14].

The multiple queue task schedulers that have been performed are six. Each scheduler has its own scheduling policy. Three of them have a scheduling policy based on one only type of task for their priority queue, and the type of task that is chosen belongs to the urgent group. The other three scheduling policies are based on two types of tasks (tracking and one type that belongs to urgent group). The scheduling policies are:

- Scheduling policy 1: Where confirmation tasks are selected and assigned to the priority queue.
- Scheduling policy 2: Where backscanning tasks are selected and assigned to the priority queue.
- Scheduling policy 3: Where reacquisition tasks are selected and assigned to the priority queue.
- Scheduling policy 4: Where confirmation and tracking tasks are selected and assigned to the priority queue.
- Scheduling policy 5: Where backscanning and tracking tasks are selected and assigned to the priority queue.
- Scheduling policy 6: Where reacquisition and tracking tasks are selected and assigned to the priority queue.

Figure 2 shows the scheme of a multiple queue task scheduler. First of all, there is a list that contains all the tasks to execute. At the beginning, this list is formed only by surveillance tasks, because the space has not been scanned yet. When the radar system is working, tasks of any of the five types appear and are included into that list to be planned to execute.

When the scheduler applies its policy to the list of tasks, two different queues are obtained. The priority queue is formed by all the tasks that are considered of high priority with the scheduling policy applied. The normal queue is formed with the others that are not so priority.

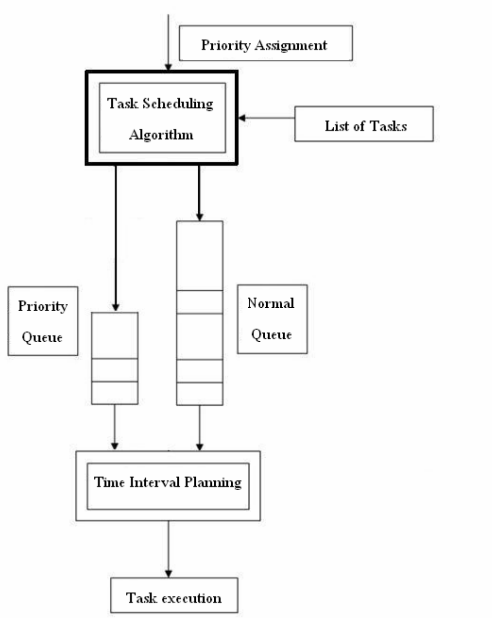


Fig. 2 Scheme of a multiple queue task scheduler

Once the tasks are distributed in the queues by the specific requirements, the second step of the policy is applied, selecting the tasks to be executed for each scheduling interval.

International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:7, 2009

The scheduling policy is described as follows:

a) Therefore, at first, the scheduler takes the tasks from the priority queue, and only when this queue is empty, it begins to take tasks from the other queue.
b) The tasks in the priority queue need to be executed urgently, because the priority policy applied by the task scheduler proposes that, so those tasks are the first to be planned in the scheduling interval.
c) The tasks in the normal queue are not considered urgent, so they can be executed after the tasks from the priority queue.
d) When the priority queue is empty because all of its tasks have been planned, the tasks from the normal queue can be planned to execute.
e) When the task scheduler completes the scheduling interval planning, it sends it to execute.
f) The execution of the tasks during each scheduling interval generates new tasks, (from the execution results), that are included into the list of the tasks of the scheduler.

## III. Obtained Results

The obtained results and its analysis are based on the study of both surveillance and tracking.

The set of characteristics of the radar that are measured to study the kindliness of every scheduler has been chosen on the basis of radar execution performance, so it is characterized by:

- The total radar time missed.
- The number of executed tasks, of every type of task, versus the theoretical one.
- The average delay that the executed tasks suffer, for every type of task.

Table I shows the radar time that is not used during the execution of the tasks, therefore, missed during the work of each task scheduler. It is observed that the scheduling policies 5 and 6 are the best using the radar time, because the radar time missed is between 20% and 30%. Although these results are good for this type of task scheduler, it can be seen that these schedulers do not give a very good performance at all.

The worst cases have been found with the scheduling policies 2 and 3 that miss between 42% and 45% of radar.

TABLE I
FRACTION OF FADAR TIME (%) MISSED FOR EVERY SCHEDULER

| **Scheduling Policy** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| $RT_{missed}$ | 27.82 | 41.23 | 41.05 | 29.21 | 11.15 | 13.41 |

Table II shows the percentage of tasks of surveillance and tracking that are executed, versus the theoretical value that is calculated in an analytical way, for every type of task, and for scheduling policy. The results show that the surveillance tasks are executed for almost all of them.

The percentage of executed tracking tasks is lower than surveillance for this type of scheduler. The scheduling policies 5 and 6 offer better results because all of them allow executing more than the 90% of the tracking tasks.

TABLE II
NUMBER OF EXECUTED SURVEILLANCE AND TRACKING TASKS (PERCENTAGE) FOR EVERY SCHEDULER, VS. THE THEORETICAL ONES

| **Scheduling Policy** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| *Surveillance* | 100.0 | 100.0 | 100.0 | 99.99 | 99.66 | 99.36 |
| *Tracking* | 67.11 | 48.30 | 48.57 | 66.22 | 95.31 | 91.89 |

Table III shows for every scheduler, the average delay suffered by surveillance and tracking tasks executed during the time the radar is working.

Although it is necessary to take into account both average delays, for surveillance and tracking tasks, always the result obtained for tracking is the most important one. The reason is that the frequency of execution for tracking is higher than for surveillance, and also, because their different relevance. It is necessary to consider this criterion to observe and analyze the results in a good way.

Therefore, the better performance is given by the scheduler that has the best results in both average delays, for surveillance and tracking tasks.

It is observed that the best behaviour is obtained for the scheduling policies 5 and 6, since both values in the table are the lowest ones, although for tracking are still lowest than for surveillance.

The scheduling policies 1, 2 and 3, offer results that can be considered as acceptable. All of them have very low average delays for surveillance tasks and acceptable average delays for tracking ones.

TABLE III
AVERAGE DELAY (MILLISECONDS) IN SURVEILLANCE AND TRACKING FOR EVERY SCHEDULER

| **Scheduling Policy** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| $\bar{\tau}_{Sur}$ | 0.01 | 0.01 | 0.00 | 0.72 | 66.87 | 129.37 |
| $\bar{\tau}_{Tra}$ | 72.79 | 60.90 | 60.89 | 498.46 | 0.40 | 0.84 |

## IV. Analysis Results

Now it is presented the detailed analysis that has been made from the obtained results for the selected characteristics to measure the behaviour of the tasks scheduler.

### *A. Radar Time Utilization*

The best radar utilization is obtained using the scheduling policy 5 that uses a priority policy based on backscanning and tracking tasks. This occurs because these types of tasks are the most important in radar environment for tracking the targets without losing them. So, the criterion is to execute them as soon as possible and leaving the rest of the time scheduling interval to execute the other types of tasks, in the normal queue. Scheduling policy 6 provides acceptable results as it assigns priority for tracking and reacquisition tasks.

The radar time used for executing surveillance tasks is the expected as the scheduler planned. For all the schedulers this value overcomes 99% of the theoretical calculated value.

With scheduling policies 5 and 6 the radar time used for tracking is over 90% of the theoretical calculated value. Scheduling policies 1 and 4 can be accepted because 66% of the theoretical value is considered high enough.

*B. Number of Executed Tasks*

All the schedulers allow the execution of at least 99% of the planned surveillance tasks. The scheduling policies that offer the best proved are those that allow executing almost the whole planned tracking tasks. These scheduling policies are 5 and 6. Anyhow, the percentage of executed tracking tasks, versus to the corresponding theoretical value, overcomes 90% for both scheduling policies.

*C. Average Delay of Executed Tasks*

The scheduling policy that offers best results complies with low average delays for both types of tasks (surveillance and tracking).

Some scheduling policies have only low delay for executed surveillance tasks. They are scheduling policies 1-4.

It can be found that the scheduling policies 5 and 6 have only low delay for executed tracking tasks. The scheduling policy 4 has the highest value for this delay that is nearly 500 milliseconds, which is half the time dedicated to tracking tasks.

Table IV shows the set of results of the evaluation of each scheduler for all measured radar characteristics. There have been established some levels (good-passable-bad) to determinate the behaviour of each scheduling policy for each radar characteristic. These levels have been established for each type of characteristic based on the study and analysis realized and exposed before.

TABLE IV
EVALUATED RADAR CHARACTERISTCS FOR EVERY SCHEDULER

| Algorit. | Measured Radar Characteristics | | | | |
|---|---|---|---|---|---|
| | *Radar Time missed* | *Surveillance Executed* | *Tracking Executed* | *Average Delay Surveillance* | *Average Delay Tracking* |
| **1** | Passable | Good | Passable | Good | Passable |
| **2** | Bad | Good | Bad | Good | Passable |
| **3** | Bad | Good | Bad | Good | Passable |
| **4** | Passable | Good | Passable | Good | Bad |
| **5** | Good | Good | Good | Passable | Good |
| **6** | Good | Good | Good | Passable | Good |

As it can be observed each performance of the task scheduler offers different results. These results are evaluated:

- Scheduling policy 1: It offers good results for surveillance tasks. The radar time missed and tracking results are acceptable. So this scheduling policy is considered as passable for this task scheduler.
- Scheduling policy 2: This one gives bad results for tracking tasks and radar time missed. Although the results for surveillance tasks are good, the other characteristics are important enough to show that this scheduling policy is one of the worst one for this task scheduler.
- Scheduling policy 3: This scheduling policy, as the previous one, also gives bad results for tracking tasks and radar time missed. Although the results for surveillance tasks are good, as it is said before, radar time missed and tracking tasks are quite important characteristics that shows this scheduling policy is also another of the worst one for this task scheduler.
- Scheduling policy 4: It offers good results for surveillance tasks, and the radar time missed is not bad. But the tracking results are bad, so this scheduling policy is considered passable for this task scheduler.
- Scheduling policy 5: This one offers good results for all the radar characteristics. Only the average delay for surveillance tasks could be improved, but its global results show this scheduling policy is one of the best for this task scheduler.
- Scheduling policy 6: This scheduling policy, as the one before, also offers good results for all the radar characteristics, in general. Only the average delay for surveillance tasks could be improved, but its global results show this scheduling policy is another of the best for this task scheduler.

As it has been shown before, there have been measured and evaluated five radar characteristics. But their influence in the multifunction radar behaviour occurs in different way. So it is necessary to take into account the importance of the effect that every characteristic has on task scheduler behaviour.

Tracking and not losing the targets in the scenario are both considered really important for radar performance. Because of that, both characteristics about tracking are the most important ones joined to the radar time missed, that indicates the task scheduler efficiency.

Meanwhile, about surveillance, the number of executed tasks is more important result than its average delay. But both of them are not as important as the tracking ones, with regard to task scheduler performance.

So, the best multiple queue task schedulers are those whose task schedulers take tracking tasks and another type of task considered as urgent (backscanning or reacquisition) to make their priority queue. The best scheduler behaviour is really good as radar tracking of the targets and as radar scheduling efficiency.

After this, it can be observed that scheduling policies 5 and 6 are what present the best results, because their delays are low or acceptable, an also, according to the results of radar time used and executed tasks.

## V. Conclusions

A detailed analysis of families of schedulers with multiple queue and the influence of their parameters in the behaviour of the task schedulers have been presented in this work.

This paper shows that for radar task schedulers based on multiple queue, the scheduling policies that works with both types of tasks (tracking and only one of the urgent group) give better results and more efficient performance than scheduling policies that work only with one type of tasks.

This is because the task scheduler must be effective to execute urgent tasks, but it also must follow executing tracking tasks for all the detected targets in the scenario.

Also it is verified that specifically two of those schedulers are really the best ones, those that work with backscanning or reacquisition, joined to tracking, in their scheduling policy for making the priority queue. They provide the best performance of the tasks scheduler, because maximize the radar time utilization and minimize the delay of the executed tasks.

## References

[1] S. Kingsley, *Understanding radar systems*, Sci Tech Publishing, 1998.
[2] M. I. Skolnik, *Radar handbook*, 2nd Ed., McGraw-Hill, 1990.
[3] D. L. Billeter, *Multifunction array radar*, Artech House, 1989.
[4] S. Sabatini, M. Tarantino, *Multifunction array radar: System design and analysis*, Artech House, 1994.
[5] D. K. Barton, *Radar technology encyclopedia*, Artech House, 1997.
[6] D. K. Barton, *Radar system analysis and modeling*, Artech House, 2005.
[7] A. J. Orman, A. K. Shahani, A. R. Moore, “Modelling for the control of a complex radar system”, *Computers & Operations Research, Vol 25, Nº 3,* pp. 239-249, Mar 1998.
[8] J. Holloway, “Design considerations for adaptive active phased-array multifunction radars”, *Electronics and communication engineering journal,* pp. 277-288, Dec 2001.
[9] F. Bolderheij, F. G. J. Absil, P. van Genderen, “A Risk-Based Object-Oriented Approach to Sensor Management”, *IEEE Proceedings of the 7th International Conference on Information Fusion* 2005.
[10] E. R. Billam, “The problem of time in phased array radar”, *Proceedings of Radar (Conf. Publ. nº 449),* pp. 563-575, Oct 1997.
[11] C. Cheng, C. Chien-Chung, C. Yao-Liang, H. Fu-Shin, “Real-time scheduling in a programmable radar signal processor”, *Proceedings of the 4th International Workshop on Real-Time Computing Systems and Applications,* pp. 206-213, Oct 1997.
[12] G. R. Curry, *Radar system performance modeling*, Artech House, 2001.
[13] A. Izquierdo, J.R. Casar, “A simulation methodology for multifunction radar evaluation”, *Proceedings of the IASTED International Conference Modeling and Simulation, Pittsburgh, PA, USA,* pp. 103-106, May 1993.
[14] A. G. Huizing, A. A. F. Bloemen, “An efficient scheduling algorithm for a multifunction radar”, *IEEE Proceedings of the International Symposium on Phased Array Systems and Technology* 1996.