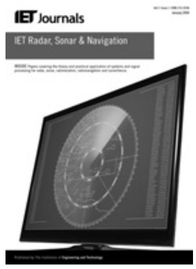
www.ietdl.org

Published in IET Radar, Sonar and Navigation
Received on 9th September 2011
Revised on 12th December 2011
doi: 10.1049/iet-rsn.2011.0309



ISSN 1751-8784

Design of task scheduling process for a multifunction radar

M.I. Jiménez[1] *L. del Val*[1] *J.J. Villacorta*[1] *A. Izquierdo*[1] *Mariano Raboso Mateos*[2]

[1]*Department of Signal Theory and Communications and Telematic Engineering, University of Valladolid, Spain*
[2]*Universidad Pontificia de Salamanca, España*
E-mail: marjim@tel.uva.es

Abstract: During the operation of the multifunction radar system, all the tasks related to the functions of the radar must be launched. The key element of the radar responsible for managing all these tasks is the task scheduler. Many scheduling techniques give good results at the expense of too complex and expensive designs. This study presents the results of a model for a radar task scheduler to achieve both a simple design and a good performance. The scheduling process consists of three stages in which the whole scheduling is divided into: task priorisation, scheduling algorithm and temporal planning. A task priority method is established to be applied to the tasks and the scheduling algorithms that have been tested based on this criterion for the priority task queue building. The authors have developed a software platform for testing all scheduling algorithms. The evaluation of the schedulers was made based on a set of features of the radar to measure the system's performance from the timing and the tasks execution. The authors offer a model to test the global radar system focusing on the task scheduler. This way allows us to analyse different scheduling algorithms and policies, and applying specifically scheduling policies that give priority to the most important and critical tasks.

1 Introduction

Radar system is used for detection and tracking of targets. Specifically, a multifunction radar or multifunction array radar (MFAR) 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 [1, 2]. All these functions are divided into tasks that must be managed and executed during the radar system operation.

The antenna consists of an electronic scanning array that can focus on the desired angle in a short time as it is electronic-based. This type of antenna allows running multi-target tracking systems. Therefore this kind of radar can carry out surveillance of a wide area of scanning and tracking of a high number of targets simultaneously. These are key characteristics of MFAR that require complex design and development. Nowadays, most of the advanced navigation systems employ this type of radar. Analytical methodologies cannot be used, so it is necessary to use simulation techniques based on the Monte-Carlo method.

Fig. 1 shows the typical MFAR block diagram [3–5]:

- *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, energy and time, for each task.

The main resources of a radar system, time and energy, are limited. The task scheduler is a key element of the radar, since it carries out the planning and distribution of the energy and time resources to be shared and used by all those tasks in an efficient form [6].

That is why task scheduling is a very essential problem of radar resource management that has been the focus of intensive research [7]. This work focuses on the task scheduler performance and the analysis of its behaviour to improve the scheduling process for these devices.

The functions of the radar implemented are divided into specific tasks. The types of task determined are:

- *Surveillance task:* Scanning of the space looking for targets.
- *Confirmation task:* When a surveillance task finds a new target, the confirmation task rules out if it is a false alarm to confirm the new target or decide it was a false alarm.
- *Tracking task:* If the confirmation is made, a new track is started to follow the new target found.
- *Backscanning task:* Sometimes a target in tracking can be lost, and this type of task makes a set of several tracking tasks to find the target.
- *Reacquisition task:* If the backscanning task fails, this type of task allows finding the lost target transmitting several beams in time over the area where the target is expected to be.

All the tasks to be executed immediately are called in this work as immediate tasks, for instance, false alarms or lost targets in tracking. Therefore the types of tasks confirmation, backscanning and reacquisition are defined and grouped as immediate tasks.

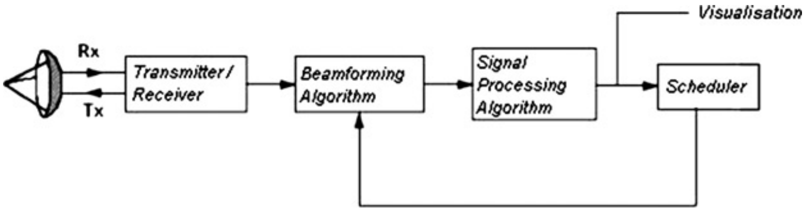


Fig. 1 *Multifunction radar diagram*

The radar task scheduler manages all the tasks to be executed, and in order to achieve an adequate performance of the system, the tasks must be executed perfectly and before the deadline.

Before the radar system starts to work there are only surveillance tasks to be executed. All these tasks are included on a task list, as the scheduler picks the tasks from this list. The following tasks generated while the radar is working are also included on that list before the system starts the scheduling process.

The execution of the tasks during each scheduling interval generates new tasks, from the execution results, which are included on the task list. At the same time, all the accomplished tasks are eliminated from the list.

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

- *Initial time:* When the task can start to be executed.
- *Deadline:* When the task execution must finish.
- *Length:* Duration of the task execution.
- *Priority:* Order of importance of the specific task execution.

In Fig. 2, the temporal parameters of the task, initial time, deadline and length, can be shown.

The tests and experiments have been performed using software simulation of the designed system, developed from a previous model [8, 9]. This work focuses on the modelling and simulation of schedulers that use queues of tasks as a scheduling technique. This type of scheduler builds the task queues to make the specific planning for every scheduling interval. From the built queues, the task scheduler takes the tasks to plan them as they are going to be executed in due time [10].

Queue-based task schedulers provide good results for not too complex dynamic scenarios with simple system designs [11]. The queue-based schedulers that are more basic and simple are those with a single queue, and whose queue is built by task ordering. This ordination can be done according to different classification criteria. This type of scheduler is easy to implement and provides a good performance for classical dynamic scenarios [12–15].

Many of these schedulers do the planning for the execution of the tasks as soon as possible, based on a sequential execution. To do this, the built of the queue is only made looking at the length of the tasks, their starting and ending times, and the task priority [16].

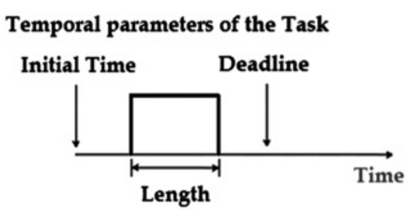


Fig. 2 *Temporal parameters of the task*

Among the task scheduler algorithms based on queue, there are some of a more complex type. These algorithms use two or more task queues. They construct their queues not only based on ordinations but also taking into account the types of tasks, the values of the parameters that characterise them and the policy for building such queues [17]. Also, they may take into account issues such as the sectors of airspace scanning and different importance or danger of the targets.

Classic schedulers based on two queues construct two different types of queues depending on the types of work they put into them. Tasks are classified as priority and non-priority, and that is how the queues are processed [16]. It also considers the time parameters of tasks to run on time. That is why this type of scheduler, that uses two queues, has been the selected one to test the radar task scheduler model designed because it is related to the scheduling process that has been developed [17].

Heuristic criteria based on the analysis of the experimental results offer the ability to make decisions and choose appropriate scheduling strategies according to the objectives that arise [18]. The scheduling techniques that are based on heuristics include task scheduling based on system status at all times, the requirements and response of the system at any time. These scheduling strategies provide reliable and useful tools to analyse the behaviour of task schedulers on the basis of experimentation in order to verify theoretical approaches [19, 20].

It is also essential to analyse the feasibility of implementing a real scheduler, from the point of view of design and affordable cost. For this reason, under these criteria, the performance assessment scheduler can provide useful information for the final implementation decision.

In the literature there are few studies regarding task schedulers, so it is necessary to provide one. There is a study that shows two classic types of schedulers for different loads of the system that says that the performance response varies depending on the different load situations [21]. Also, there is another work about task prioritising and a study for task schedulers based on queue. This one presents that fixed prioritising offers simple and good results compared with fuzzy logic techniques [9].

Also, previous studies have analysed different prioritisation methods applied to the tasks. They have demonstrated that employing fixed priorities for the tasks results in good performance with a simple design. This prioritisation method is based on the priority order typically used in radar scheduling, where tracking tasks have a higher priority than surveillance [7].

Therefore although the queue-based task schedulers are known and used, there is no study that deals with performance of a model from the point of view of the aims of the task scheduler. In this work, we have tested a set of queue-based task schedulers. We used a single software

doi: 10.1049/iet-rsn.2011.0309

platform, which was designed and developed with the aim of analysing the behaviour of such operators for the same scenario [22].

Section 2 presents the task scheduler model design and details the task schedulers that have been researched. In Section 3 the obtained results are shown, and also the analysis of those results. Finally, we present the conclusions of our research.

2 Task scheduler design

Based on the queue a task scheduler selects, from the list of tasks that it has to execute, a set of tasks to build the specific queues. The initial task list is defined as the whole group of tasks that have to be executed. The new tasks that are obtained after the execution of the first tasks are included into this list before the starting of the next scheduling process.

Time, as a resource, is shared among all the tasks to be executed, so the whole radar time available is divided into small and equal time intervals called scheduling intervals, which can be seen in Fig. 3.

Once the tasks are in the queues, for every scheduling interval (of the radar time), the tasks are chosen and planned to be executed in the above-mentioned time interval [10]. This process is followed for every scheduling interval, and also, the complete scheduling process is made step by step for every scheduling interval.

After analysing the schedulers' performance and in order to simplify the scheduling process, the scheduling was broken up into stages to handle it appropriately, and address all the steps involved in the said scheduling.

In this work, we came up with a common functional structure based on these three stages, so we have designed our own scheduling scheme that defines and applies three scheduling stages to perform the whole task scheduling process that is shown in Fig. 4. The first stage consists of the priorisation of the tasks, the second stage applies the specific scheduling algorithm based on queue, and finally, the third stage assigns the time planning to execute the tasks.

All of the schedulers tested in this work use this three-stage process to set up the task scheduling for the radar performance. The task scheduler applies its scheduling policies through out these three stages for every scheduling interval. A set based on queue task scheduling algorithms

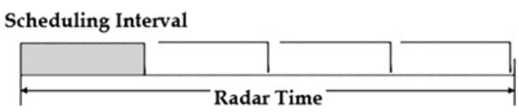


Fig. 3 *Scheduling interval*

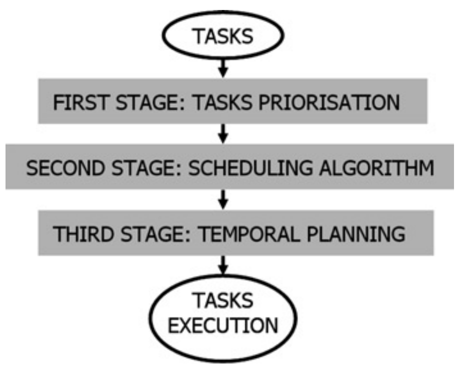


Fig. 4 *Task scheduler process*

was analysed. All the parameters associated with the tasks that the scheduler manages were considered.

2.1 *Three stage scheduling process*

2.1.1 First stage: tasks priorisation: The first stage of the task scheduling process consists in the priorisation of the tasks. First of all, it is necessary to define what priority means for the radar task scheduler, and which set of levels the scheduler will establish [7]. After this, the scheduler has to assign a specific numeric value of priority for every pre-established level. Then, the task scheduler must carry out the assignment of the priority value for every task. Once this stage is completed, the scheduler can go on to the next stage.

Multiple definitions and assignments of priority were essayed. At first, five priority levels were established, assigning one of them for every type of task: surveillance, confirmation, tracking, backscanning and reacquisition. However, it was observed that for this kind of scheduler the best behaviour was obtained if the tasks were grouped as follows:

- Surveillance
- Tracking
- Immediate

There are then three levels of priority, and the assignment of numeric values was done allotting the highest priority to immediate tasks, then to tracking tasks and finally to surveillance tasks.

2.1.2 Second stage: scheduling algorithm: This stage is very important, because this is where the queues are built up. A queue is a structure made up of tasks. The task scheduler uses a queue policy that consists in applying a set of rules to determine how to build every queue and which tasks are going to be included [23].

The task schedulers designed and developed in this work can use different types and number of queues. The second stage applies the scheduling algorithm that includes two different policies: the queue policy and the scheduling policy. The queue policy refers to how the queues are built according to the list of tasks, whereas the scheduling policy is related to the management of the tasks in the different queues and how the scheduler is going to pick up the tasks from the different queues, before the third stage. In this work, both policies have been considered as the complete scheduling algorithm that is used by the radar task scheduler.

2.1.3 Third stage: time planning: The last stage is when the time scheduling is planned and applied on the task execution. The task scheduler has to select which task must be executed at every moment for each scheduling interval. The selection and planning of tasks for their execution in the correct scheduling interval is carried out by means of a time planning policy. This time planning policy tries to plan the execution of every task as close as possible to its initial time, and using the radar time without wasting it. Good scheduler behaviour offers good radar time use, without gaps between the sequentially executed tasks, and reducing possible time wasted.

2.2 *Two queues task scheduler scheme*

The task scheduler that has been chosen works according to the three scheduling stages explained above. The second stage is the main stage of task scheduling, and shows the technique based on queue scheduler.

First of all, the tasks are included on a list called the task list, where both the generated and new tasks are saved when they appear in the radar system. The scheduler picks up the tasks from this list, and carries out the priority assignment for those tasks.

After that, the queues are made functions of the algorithms, including the tasks in the corresponding queue. Then, the scheduler picks up the tasks from the queues for planning every specific scheduling interval. But only when the task scheduler completes the scheduling interval planning, it sends the tasks to be executed.

The most characteristic algorithms were chosen on the basis of the design and test of this type of task scheduler. The algorithms selection was made according to the definition and own properties.

Fig. 5 shows the scheme of the queue task scheduler. First of all, there is a list that contains all the tasks to execute. This task scheduler works with two different types of queue [17] on two priority levels [11]:

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

First of all, the priority queue is built, and afterwards a normal queue is made with the rest of the tasks [16]. The priority queue is formed by all the tasks that are considered high priority with the scheduling policy applied. The normal queue is formed with the ones that are not so much a priority.

The tasks in the priority queue need to be executed urgently, because the priority policy applied by the task scheduling algorithm proposes so, thus, those tasks are the first to be planned in the scheduling interval. The tasks in the normal queue are not considered urgent, so they can be executed after the tasks in the priority queue.

The scheduling process consists of three stages, so the queue implementation is related to the second and third stages. In the second stage, the two queues are built from the tasks that are in the list, by means of the criterion of the scheduling algorithm (queue policy) that is applied in each case. This way, the queue implementation (scheduling policy) is an ordination of the tasks looking at their deadline, the tasks whose deadline is sooner, comes first. When some tasks have the same deadline, it puts the shortest one first.

After the building of the queues, in the third stage, the tasks from the priority task queue are selected to be executed, and when that queue is empty, the tasks from the other queue are chosen. So, the execution of the tasks is planned as they are going out of the queue that they belong to, considering an earliest deadline first (EDF) structure. The EDF algorithm consists in the ordination of the tasks based on their deadline for being executed.

2.3 Scheduling algorithms

These scheduling algorithms use two queues, but their classification is based on the policy to build and manage the priority queue. So, every specific scheduling algorithm describes a different number and type of tasks that are used to make the priority queue.

These are the task scheduling algorithms:

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

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.

The task scheduler selects the tasks for the next stage picking the tasks from the priority queue. Once the priority queue is empty, the scheduler takes the tasks from the normal queue as it was exposed before [17].

3 Three-stage scheduler results

3.1 Methodology and evaluation

The followed methodology to realise this work is based on simulation tools that have been programming for designing, modelling and developing the performance of the surveillance and tracking of targets in a specific scenario for the typical multifunction radar parameters as we will explain later.

The simulation techniques are based on radar performance, but under the point of view of the scheduler of the

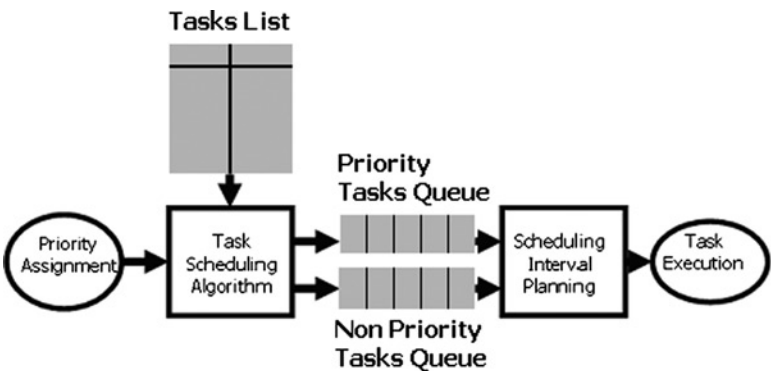


Fig. 5 *Task scheduler scheme*

doi: 10.1049/iet-rsn.2011.0309

multifunction radar. This means that the main blocks of the radar have been developed as functionalities of the system, whereas the task scheduler has been designed in detail to observe all the events and radar characteristics involved in it [22].

Radar time is divided into scheduling intervals with a reference value of 100 ms [18]. The elapsed time for simulation that has been chosen is 1000 s. As in the Monte-Carlo method, a value of 100 executions is chosen for each experiment [24]. A typical scenario with 200 targets uniformly distributed was selected, where the range reaches up to 70 km, being the sectors of variation in azimuth of 120° and in elevation of 60°.

The parameters chosen for the radar implementation are: detection probability $P_D = 0.99$ and false alarm probability $P_{FA} = 10^{-4}$. It is assumed that the air space is scanned every 20 s and the targets are tracked every second, as surveillance and tracking intervals, respectively.

In fact, as a target is tracked every second while a surveillance task is launched every 20 s, it was decided that the tracking average delay should be less than 5 ms and surveillance average delay should be around 100 ms, values that correspond to 0.5% of the values used for each type of task. These values were chosen after analysing the times allowed for both types of tasks to be executed in time. The results obtained in these cases and their analyses are based on the study of surveillance and tracking modes.

The aim of the designed system is to minimise the resources consumed for each target, as this automatically implies maximising the ability of the system. However, minimising resource consumption is not simply minimising each of the resources, since they are interrelated and have a system with a high degree of integration. Therefore it is always necessary to balance the use of all resources involved [16].

The set of characteristics of the radar that has been chosen to study the kindness of the scheduling algorithms and to measure the task scheduler behaviour was established based on radar performance, after a previous study [23]:

- The radar time used for the execution of every type of task.
- The number of executed tasks, of every type of task, against the theoretical one.
- The average delay that the executed tasks experience, for surveillance and tracking tasks.

Radar research requires simulation methodologies because of the difficulty and complexity of obtaining a valid and quick performance. Analytical methodologies cannot be used, so it is necessary to use simulation techniques based on the Monte-Carlo method. This method consists in executing each experiment during a defined time interval that must be long enough to verify the system behaviour. Also, the method requires establishing the specific number of times to repeat each experiment in order to offer valid results [25].

The simulations' implementation has been made a number of times and is enough to make the results valuable. It means that every experiment to test the task scheduler in the radar system has been made 100 times to check every algorithm response for different conditions of task priority assignment.

As the number of targets is fixed and they appear on the scenario as the system starts to run, the radar begins working in a transitory way. But afterwards, when all the targets have been detected, the radar works as usual till it reaches a stationary behaviour. That is why it has not been necessary to repeat every experiment too many times.

3.2 Obtained results

The results obtained for the task scheduler are shown, indicating all the values of the measured radar characteristics for every scheduling algorithm studied. The theoretical values are calculated according to the periods of surveillance and tracking modes, in an analytical way. For every radar characteristic measured and for every scheduling algorithm, the typical statistics (mean, deviation etc.) values were calculated to obtain results that can be taken into account in radar performance.

The best scheduling algorithms are those that allow the execution of most of the tasks with the least average delays and without wasting radar time (in gaps) or, at least, that this wasted radar time is as short as possible.

Although it is important to take into account average delays for both surveillance and tracking tasks, the result obtained for the tracking ones is always more critical. This is due to the fact that the frequency of execution of a tracking task is higher than the surveillance task itself, and also, because of their different nature. It is necessary to consider these criteria in order to see and analyse the results in the best way.

The criteria used for the scheduler were chosen to carry out the analysis and evaluation of all the algorithms correctly. In order to do so, radar time utilisation should be the highest possible, at least 85%. The percentage of executed tasks over 95%, the average delay for surveillance tasks should be around 100 ms and finally, the average delay for tracking tasks should be below 1 ms.

Table 1 shows the radar time used for the execution of the tasks, for each task scheduling algorithm. As we can observe algorithms 3, 5 and 6 give the best radar time, between 85 and 90%.

The worst case was algorithm 2, as it misses around 41% of radar time.

Table 2 shows the percentage of surveillance and tracking tasks that are executed, against the theoretical value. The results show that the surveillance tasks are totally executed for the six algorithms.

The percentage of executed tracking tasks is lower, as it usually happens with this type of algorithm. The algorithms 3, 5 and 6 are again the ones that offer better results, because all of them allow the execution of more than 90% of the tracking tasks. These three algorithms work especially well to retrieve the lost targets and continue their tracking.

In Table 3 we can observe, for every algorithm, the average delay of the surveillance and tracking tasks.

Therefore the best performance is given by the algorithm that shows a tracking average delay result not greater than 1 ms. Likewise, the surveillance average delay offers values around 100 ms.

Table 1 Radar time (%) used

Scheduling algorithm	1	2	3	4	5	6
RT_{used}	72.20	58.90	86.71	70.79	88.85	86.59

Table 2 Number of executed surveillance and tracking tasks (%) against theoretical ones

Scheduling algorithm	1	2	3	4	5	6
surveillance	100.0	100.0	100.0	99.99	99.66	99.36
tracking	67.11	48.57	91.39	66.20	95.31	91.89

Table 3 Average delay (ms) of surveillance and tracking

Scheduling algorithm	1	2	3	4	5	6
$\bar{\tau}_{Sur}$	0.01	0.01	0.03	0.72	66.87	129.37
$\bar{\tau}_{Tra}$	72.80	60.90	91.20	498.46	0.40	0.84

It is observed that the best behaviour is obtained by algorithms 5 and 6, since they offer both the lowest tracking delay and an acceptable surveillance delay, which are the desirable results for good task scheduler behaviour.

3.3 Results analysis

The best behaviour for this task scheduler type is achieved by algorithm 5. The percentage of executed tasks is adequate, the average delays are satisfactory and the radar time wasted is reasonable. This occurs because this algorithm builds the priority queue from the tracking and backscanning tasks, which are very important for radar performance due to the nature of the tracking mode and also because, if a target is lost, the backscanning task is the type of task that can retrieve it as fast as possible.

Thus, the best algorithm is the one whose strategies take up tracking tasks and other type of tasks considered as immediate to build up their priority queue.

In the model we work with, five radar characteristics were measured and evaluated. However, their influence in the multifunction radar behaviour occurs in a different way. Therefore it is necessary to take into account the importance of the effect that each characteristic has on task scheduler behaviour.

Tracking targets in a given scenario without losing them is considered of great relevance for radar performance. Owing to this, the number of executed tasks and average delay together with radar time lost indicate task scheduler efficiency.

On the other hand, as far as surveillance, the number of executed tasks is a more important result than its average delay. But, with regard to task scheduler performance, both of them are not as important as tracking tasks.

Therefore based on the results obtained and their relevant analysis, it appears that the best algorithms are those that accomplish the maximum number of tasks with a minimum time delay and make optimum use of the available radar time.

The analysis of the algorithms behaviour is made based on established thresholds that have been decided in function of the regular radar performance, for the different radar characteristics that are measured.

The best result is obtained by the algorithm that uses 89%, approximately. This confirms that, if the scheduler complexity is higher, radar time wasted is lower. This is because of the decisions and comparisons between the tasks from several queues that allow making the time planning, using radar time in the best way.

The percentages of executed tasks are satisfactory for the scheduler, because it shows the best results with 99% for surveillance tasks and upper 91% for tracking tasks, as the highest priority is given to immediate tasks, and the next priority level to tracking tasks.

The task scheduler presents good results for average delays, the same for tracking as for surveillance. The tracking mode is always more important than surveillance mode in radar performance, therefore this fact is taken into account in the analysis of the results.

For this reason, the scheduling algorithm that offers the best results requires a complex enough design as a scheduler based on queue to balance the results and the design.

A task scheduler offers satisfactory behaviour if it executes most of its tasks (over 85% against theoretical tasks) and the average delays of the executed tasks are reasonable, if it, as well, uses radar time as much as possible, then the task scheduler has a very good behaviour.

The designed and implemented model has allowed testing a complex system by means of the focusing on the radar key element we are interested in. That is why the valid results obtained demonstrate that the model can be used as a tool to test all the algorithms developed for the radar task scheduler.

The three-stage scheduling process designed and modelled simplifies the scheduling, because it decreases the complexity, and allows consideration of more parameters in the scheduling, doing independently the priority tasks assignment and the queue building, which improves the scheduler working, and hence the radar too.

The building of two different queues based on the tasks priority adds a relevant parameter for the scheduling process that improves the radar because the radar time is then used in a better way to function with the temporal radar system necessities. For this, the dual queue task scheduler is a good reference scheduler to test this scheduling process because it is a simple scheduler but is valid enough to obtain results that can be taken into account in this kind of research analysis.

The radar time resource is planned in the best possible way for the execution of the tasks, and all the task features, temporal and priority, are used for the scheduling, which means that all the information of the tasks is employed in favour of improving the radar's working.

4 Conclusions

A model of a radar task scheduler has been designed and tested to analyse the performance of this type of system. The aim of improving radar task scheduling and making it more efficient has achieved a three-stage scheduling process in this work. This scheduling process simplifies the scheduling and separates the priority task assignment of the rest of the scheduling actions about tasks managing and time planning.

The task schedulers based on queue provide a good performance with simple designs, which facilitates deployment and reduces computational load, but there was no analysis based on the design of the radar task scheduler. The selection of a dual queue task scheduler for testing the scheduling process has been very remarkable because the differentiation of priority and non-priority queues has been able to prove how the three stages work independently.

The scheduling algorithms of this scheduler perform complex enough schemes as comparisons and relationships between tasks belonging to different queues, to use the radar resources in the best way, executing all tasks on time and without wasting radar time.

The tests and experiments have been realised by means of software simulation of the designed system, which is developed from the heuristic model decided from the beginning of the work.

Guidelines have been provided for the design of scheduling algorithms based on queue by means of simple strategies that achieve a good performance for multifunction radar systems,

doi: 10.1049/iet-rsn.2011.0309

and the use of scheduling algorithms based on tasks priority, especially for the most important and most critical tasks, are the most recommended.

The main functions of the multifunction radar, surveillance and tracking, indicate the most important aims of radar working, but also it is very important to use the radar time in the best way to obtain a valid and useful radar system. In this work the scheduling process developed and tested is based on these objectives, and the results analysis takes into account this issue.

Finally, based on the obtained results in this work and considering the value of this scheduling process, it can be applied for any type of more complex queue task scheduler, and even other types of scheduler.

5 References

1 Skolnik, M.I.: 'Radar handbook' (McGraw-Hill, 1990, 2nd edn.)

2 Kingsley, S.: 'Understanding radar systems' (Sci Tech Publishing, 1998)

3 Sabatini, S., Tarantino, M.: 'Multifunction array radar: system design and analysis' (Artech House, 1994)

4 Barton, D.K.: 'Radar system analysis and modeling' (Artech House, 2005)

5 Barton, D.K.: 'Radar technology encyclopedia' (Artech House, 1997)

6 Blackman, S., Popoli, R.: 'Design and analysis of modern tracking systems' (Artech House, 1999)

7 Miranda, S.L.C., Baker, C.J., Woodbridge, K., Griffiths, H.D.: 'Phased array radar resource management: a comparison of scheduling algorithms'. Proc. IEEE Radar Conf., 2004

8 Barbaresco, F., Gaillard, P.O., Guenais, T., *et al.*: 'Intelligent radar management by advanced beamscheduling algorithms based on short-time planning and constraints relaxation strategies'. Proc. 2003 IEEE Int. Symp. on Phased Array Systems and Technology, October 2003, pp. 283–288

9 Miranda, S.L.C., Baker, C.J., Woodbridge, K., Griffiths, H.D.: 'Simulation methods for prioritising tasks and sectors of surveillance in phased array radars', *Int. J. Simul.*, 2004, **5**, (1–2), pp. 18–25

10 Barbato, A., Giustiniani, P.: 'An improved scheduling algorithm for a naval phased array radar'. Proc. 1992 IEEE Int. Radar Conf., October 1992, pp. 42–45

11 Thaens, R.: 'Sensor scheduling using intelligent agents'. Proc. Seventh Int. Conf. on Information Fusion, July 2004, pp. 190–197

12 Orman, A.J., Shahani, A.K., Moore, A.R.: 'Modelling for the control of a complex radar system', *Comput. Oper. Res.*, 1998, **25**, (3), pp. 239–249

13 Holloway, J.: 'Design considerations for adaptive active phased-array multifunction radars', *Electron. Commun. Eng. J.*, 2001, **13**, (6), pp. 277–288

14 Bolderheij, F., Absil, F.G.J., van Genderen, P.: 'A risk-based object-oriented approach to sensor management'. IEEE Proc. Seventh Int. Conf. on Information Fusion, 2005

15 Billam, E.R.: 'The problem of time in phased array radar'. Proc. Radar (Conf. Publ. no. 449), October 1997, pp. 563–575

16 Bar-Shalom, Y., Blair, W.D.: 'Multitarget-multisensor tracking: applications and advances' (Artech House, 2000, vol. III)

17 Huizing, A.G., Bloemen, A.A.F.: 'An efficient scheduling algorithm for a multifunction radar'. IEEE Proc. Int. Symp. on Phased Array Systems and Technology, 1996

18 Curry, G.R.: 'Radar system performance modeling' (Artech House, 2001)

19 Orman, A.J., Potts, C.N., Shahani, A.K., Moore, A.R.: 'Scheduling for a multifunction phased array radar system', *Eur. J. Oper. Res.*, 1996, **90**, (1), pp. 13–25

20 de Jong, J.L., van Norden, W.L.: 'Application of hybrid metaheuristics in sensor management', *Aerosp. Sci. Technol.*, 2007, **11**, pp. 295–302

21 Miranda, S.L.C., Baker, C.J., Woodbridge, K., Griffiths, H.D.: 'Comparison of scheduling algorithms for multifunction radar', *IET Radar Sonar Navig.*, 2007, **1**, (6), pp. 414–424

22 Jiménez, M.I., Izquierdo, A., Villacorta, J.J., del Val, L., Raboso, M.: 'Evaluation of radar task scheduling algorithms based on simulation methodology'. Proc. 18th Int. Conf. on Applied Simulation and Modelling, September 2009, pp. 175–181

23 Jiménez, M.I., Izquierdo, A., Villacorta, J.J., del Val, L., Raboso, M.: 'Analysis and design of multifunction radar task schedulers based on queue'. Proc. 28th Digital Avionics Systems Conf., October 2009, pp. 295–302

24 Izquierdo, A., Casar, J.R.: 'A simulation methodology for multifunction radar evaluation'. Proc. IASTED Int. Conf. on Modeling and Simulation, Pittsburgh, PA, USA, May 1993, pp. 103–106

25 Billeter, D.L.: 'Multifunction array radar' (Artech House, 1989)