Multifunction Phased Array Radar Resource Management: Real-Time Scheduling Algorithm

Jianbin LU¹,†, Hui XIAO², Zemin XI¹, Mingmin ZHANG¹

¹Electronic Engineering College, Naval University of Engineering, Wuhan 430033, China
²Air and Space Borne Radar Research Section, Air Force Radar Academy, Wuhan 430012, China

Abstract

In this paper a real-time task model of multifunction phased array radars is built, and a novel scheduling algorithm is proposed. This algorithm takes the priority of task’s function mode and deadline into account synthetically, and can adapt well to different load conditions. Based on the task model, the algorithm can achieve the variety of scheduler’s time load in a real-time fashion and adjust the tasks’ parameters correctly when the system is over loading. The simulation results show that the proposed algorithm improves the scheduler’s performance with decreasing the missed deadline rate effectively, and the adjustment strategy is rational and effective on system overload.

Keywords: Phased Array Radar; Task Scheduling; Deadline; Priority; Time Load

1. Introduction

As a multi-function and high-performance radar system, phased array radars have the advantages of flexible beam pointing direction, versatile waveforms, controllable system parameters, as well as the effective resource allocation strategy and powerful data processing capacity. It is playing an important role in the future advanced radar systems[1]. While all the aforementioned predominance depends on its effective resource management. Resource management technique for phased array radar aims at improving radar system performance by effectively task scheduling and parameter control. There are many scheduling strategies for phased array radar such as fix-template, multi-template, partial template and adaptive scheduling, where the adaptive scheduling is the most effective and complex in the real application[2].

Many recent studies[3-6] have dealt with resource management problems for phased array radar systems using real-time dwell scheduling technology, but two limitations exist. The first one is the correlation separation of different radar dwells from the same task. For a new task, it is not clear that whether the scheduler has enough resource to allocate. When the system is overloaded, the deletion of some tasks will occur. The second limitation is that only the task’s importance is considered, regardless of its urgency attribute. This will degrade the scheduler performance. The concept of radar dwell time window is presented in [7] which denotes the effective span of dwell requests. Kuo [8,9] proposes a rate-based approach to scheduling radar dwells in a real-time fashion. It reserves radar resources for all tasks

† Corresponding author.

Email addresses: lu_jian_bin@163.com (Jianbin LU).

1553-9105/ Copyright © 2011 Binary Information Press
February, 2011
necessary for the minimal radar operation. But the real dwell time window is neglected, and the strategy to deal with system overloaded isn’t discussed.

In this paper, a novel real-time task model is built for phased array radars, in which each kind of dwell requests are combined as a radar task. This task model can be effectively applied in the radar scheduler design. Furthermore, a real-time scheduling algorithm is presented based on the proposed task model.

2. Adaptive Task Model

The proposed adaptive radar task model includes the parameters of arriving time, deadline, transmission time, dwell length and function priority of every dwell request.

2.1. Search Task Model

Based on the prior information, phased array radar can divide the whole surveillance space into some smaller regions with the optimal search theory. Then different parameters can be adopted for different search region so as to maximize the radar search performance. Assuming there are \( N \) regions in whole surveillance space, the \( i^{th} \) \((i=1,2,...,N)\) region includes \( B_i \) beam positions with dwell length \( \Delta t_i \) in each beam position. The search frame time and function priority of the region are \( P'_i \) and \( pri'_i \), respectively.

Then the task model of \( i^{th} \) search region can be expressed as

\[
T_i = \{ T_i^{j,i} \mid j = 1, 2, \cdots, B_i \}, \quad T_i^{j,i} = \{ t_{ai}^j, t_{dj}^j, t_{ej}^j, \Delta t_i, pri_i \}, \quad (i = 1, 2, \ldots, N)
\]

where \( T_i^{j,i} \) denotes the \( j^{th} \) dwell request of the \( i^{th} \) search region, \( t_{ai}^j \), \( t_{dj}^j \) and \( t_{ej}^j \) are its corresponding arriving time, deadline and transmission time. Without loss of generality, for all the requests in the same search region equal dwell length and function priority are adopted.

Before phased array radar starts to search some given region, all corresponding dwell requests can be determined based on the results of search parameter optimization and beam position arrangement. To maximize the flexibility of search task, we can assume that all requests arrive in turn at the beginning of each search frame, that is

\[
\begin{align*}
  t_{ai}^1 &= t_0  \\
  t_{ai}^j &= t_0 + (j-1) \cdot \Delta t_i^i, \quad (j = 2, 3, \cdots, B_i)
\end{align*}
\]

Similarly, the deadline of each request satisfies

\[
t_{ai}^{dj} = t_0 + P'_i - (B_i - j) \cdot \Delta t_i^i, \quad (i = 1, 2, \ldots, N; \ j = 1, 2, \cdots, B_i)
\]

where \( t_0 \) denotes some reference time, at this time all search tasks begin to execute (equivalently radar boot-strap). Equ. (2) and (3) show that in each search frame all dwell requests need be accomplished in time order. When one frame ends, the similar dwell requests are generated at the beginning of next search frame.

With the search task model proposed above, the frequency of search task \( i \) is
When the search region is fixed, according to equation (4) the frequency of search task is determined by the search frame time. The bigger the amount of \( P'_i \), the lesser of search dwell requests, the lesser of the load of the radar system and vice versa. For the real phased array radars, the search frame time is correlative with the radar function, and the minimum of which is the sum of the dwell length for the total requests.

\[
P_{i,\text{min}} = B_n \cdot \Delta t_i
\]

The selection principle for the maximum \( P'_i \) is that the radar system can detect the targets which overpass the search region with the given detection probability. For the \( i^{th} \) search region with the elevation \( \theta_i \), one target overpasses this region with the velocity \( v \) at the range \( R_T \) (as fig.1). Then the time spent for the target to overpass the region is \( \delta t = R_T \cdot \theta_i / (v \cdot \sin \alpha_T) \), and the radar should irradiate the target twice at least for the enough detection probability. That is \( \delta t \geq 2P'_i \) and the maximum \( P'_i \) can be expressed as

\[
P_{i,\text{max}} = R_T \cdot \theta_i / (2v \cdot \sin \alpha_T)
\]

In the real radar scheduling processing, the search frame time \( P'_i \in [P_{i,\text{min}}, P_{i,\text{max}}] \) can be adjusted flexibly to change the load of the corresponding search tasks, which can make the search tasks adapt to the radar system resources.

### 2.2. Track Task Model

For the multifunction phased array radar, each track task corresponds to a target, and the types of track tasks include normal track, precise track, high precise track, missile guidance, etc. When suspicious targets are detected in the search mode, a confirmation task is generated in the direction of the target to verify its presence. Once a target is identified, the corresponding confirmation task is no longer needed. Instead, a sequence of periodic-like track dwells is generated to track the target.

Suppose that there are \( K \) types of track tasks in radar system, where the \( k^{th} \) type of track task includes \( M_k \) targets \( (k=1,2,...,K) \), the track task model of the \( i^k_{th} \) target \( (i_k = 1,2,...,M_k) \) is...
\[
T^{\text{R}} = \left\{ T^{\text{R}}_j \mid j = 1, 2, \ldots, T_m \right\}, \quad T^{\text{R}}_j = \left\{ t^{\text{R}}_i, t^{\text{R}}_{\Delta i}, t^{\text{R}}_{\Delta i}, \Delta_t^{\text{R}}, \sigma_t^{\text{R}} \right\}, \quad (k = 1, 2, \ldots, K; i_k = 1, 2, \ldots, M_k)
\]

where \( T_m \) denotes the number of track dwell requests for the \( k^{th} \) target (equivalently target tracking span), but it is unknown as the capture time and disappearing time of the target can’t be determined. The other parameters are similar to the search task model. \( \Delta_{t_k}^{\text{T}} \) denotes the tracking sample interval for the target \( i_k \), and without loss of generality, assuming one confirmation dwell before target tracking, then

\[
T_{\text{cap}} = \left\lceil \left( t_{\text{cap}}^{i_k} - t_{\text{cap}}^{i_k} \right) / \Delta_{t_k}^{\text{T}} \right\rceil
\]

\[
\begin{align*}
\left( t^{\text{R}}_i, t^{\text{R}}_{\Delta i}, t^{\text{R}}_{\Delta i}, \Delta_t^{\text{R}}, \sigma_t^{\text{R}} \right) & = \left( t^{\text{R}}_{i_k}, t^{\text{R}}_{\Delta i}, t^{\text{R}}_{\Delta i}, \Delta_t^{\text{R}}, \sigma_t^{\text{R}} \right), (k = 1, 2, \ldots, K; i_k = 1, 2, \ldots, M_k) \\
\left( t^{\text{R}}_i, t^{\text{R}}_{\Delta i}, t^{\text{R}}_{\Delta i}, \Delta_t^{\text{R}}, \sigma_t^{\text{R}} \right) & = \left( t^{\text{R}}_{i_k}, t^{\text{R}}_{\Delta i}, t^{\text{R}}_{\Delta i}, \Delta_t^{\text{R}}, \sigma_t^{\text{R}} \right), (k = 1, 2, \ldots, K; i_k = 1, 2, \ldots, M_k)
\end{align*}
\]

where \( \left\lceil x \right\rceil \) denotes the maximal integer which is less than \( x \), \( t^{\text{R}}_{\text{cap}}^{i_k} \) and \( t^{\text{R}}_{\text{dist}}^{i_k} \) are capture and disappearing time for target \( i_k \), \( \Delta_{t_k}^{\text{cap}} \) and \( \Delta_{t_k}^{\text{dist}} \) denote the time windows of confirmation task and the \( k^{th} \) type track task.

With the track task model above, the frequency of track task \( i_k \) is

\[
\begin{align*}
T_{\text{cap}}^{i_k} & = \left( \frac{1}{\Delta_{t_k}^{\text{T}}} \right) \\
\end{align*}
\]

where \( T_{\text{cap}}^{i_k} \) denotes the tracking sample frequency and equal to the frequency of the track dwell requests for the target \( i_k \). The tracking sample interval \( \Delta_{t_k}^{\text{T}} \) can be selected within the rational range, the minimum of which is determined by the hardware and software of the radar system, for example \( \Delta_{t_k}^{\text{T}, \text{min}} \geq SI \) where \( SI \) denotes the scheduling interval for phased array radars. And its maximum is correlated with the character of tracking task. For example, the maximum can be 2s more or less for the track maintenance of normal targets, while for the target with high threaten priority the maximum must be restricted to a shorter time range as 0.5s.

### 2.3. Other Task Model

This type of task mainly includes track loss task, self-examination task, calibration task, special experiment task, etc. Usually, those tasks only consume one or several continuous beam dwell time, which are less than search task or track task in quantity and undetermined for the radar scheduler. So an individual model should be established for each task. Assuming \( H \) types of those tasks, the \( i^{th} \) task model is

\[
T_{\text{R}}^{i} = \left\{ T_{\text{R}}^{i} \right\}, \quad T_{\text{R}}^{i} = \left\{ t_{\text{R}}^{i}, t_{\text{R}}^{i}, t_{\text{R}}^{i}, \Delta_t^{i}, \sigma_t^{i} \right\}, \quad (i = 1, 2, \ldots, H)
\]

The parameters of the task model in equ.(12) are similar to those of the search or track task. The difference is that this task has only one request. In equ.(12), \( t^{\text{R}}_{\text{cap}}^{i_k} \) and \( t^{\text{R}}_{\text{dist}}^{i_k} \) are unknown, while \( \Delta_t^{i} \) and \( \sigma_t^{i} \) vary with the different task.
Above three task models cover most phased array radar function. While the third type of task is stochastic and consumes only a little of system resources, the first two types of tasks are the main factor affecting the scheduler efficiency.

3. Adaptive Scheduling Algorithm

According to above radar task model, we know that the most tasks in phased array radar system are aperiodic and non-preempt (one task during executed course can’t be interrupted by others). In a word phased array radar task scheduling belongs to non-preempt hard real-time scheduling problem. This problem is NP-Hard, and its solution may be not existed or unique. So the heuristic method is usually used to obtain its suboptimal solution[10-11].

3.1. Design Method of Task Priority

In this paper two characteristic parameters of tasks, relative deadline and function priority, are used to design its integrated priority. The basic principles of task scheduling are: (1) the higher of function priority of dwell request is, the higher of the final priority is; (2) the earlier of relative deadline is, the higher of the final priority is.

Assuming that there are totally \( Q \) dwell requests in the scheduler currently, denoted as \( q = \{q_1, q_2, ..., q_Q\} \) which satisfy: (1) the arriving time of all dwell requests is no more than the current time; (2) the relative deadline of each request is more than its dwell length. Sort the above dwell requests by function priority from high to low and by relative deadline from early to late, and we can get two request chains, function priority chain and relative deadline chain, respectively. The sequence numbers of each request in the two chains can be obtained, which are function priority sequence number \( N_{pd} \) and relative deadline sequence number \( N_{hd} \). Obviously these two sequence numbers satisfy \( N_{pd}, N_{hd} \in [1, Q] \). Then, the integrated priority of each task can be obtained through

\[
p_i = f(N_{pd}, N_{hd})
\]

Above function \( f \) can be selected according to some special intention, while the simplest form is the linear function adopted here.

\[
p_i = [\eta \cdot N_{pd} + (Q + 2 - \eta) \cdot N_{hd}] / (Q + 1)
\]

where the factor \( \eta \) lies between 1 and \( Q+1 \). From equ.(14) some tradeoff is made between function priority and relative deadline. And for the special case \( \eta = Q/2 + 1 \), equal impact of these two factors on the final priority, we call this scheduling strategy as HPEDF.

One point to mention here that, the final priority calculated with equ.(14) may not correspond to each request one for one. When multiple requests have the same final priority, FIFO (First Come First Out) rule can be introduced to schedule them.

3.2. Realization of Adaptive Scheduling Algorithm

Different from the general real-time system, the task scheduling in phased array radar system is executed according to some fixed time interval, which is called scheduling interval. The dwell requests in the next scheduling interval are analyzed in the current scheduling interval. Suppose totally \( L \) dwell requests
q = \{q_1, q_2, ..., q_r\} in the next scheduling interval arrive at radar scheduler orderly, the realization of the current analysis in radar scheduler is as follows:

**Step 1.** Obtain the start time index \( t_p(p \geq t_q) \) and set \( i = 1 \);

**Step 2.** Delete those dwell requests whose relative deadline are less than the dwell length. Assume the number of those requests is \( n_i \), and set \( i = i + n_i \);

**Step 3.** Find out all the requests whose arriving time is less than the current time index \( t_p \), denoted as \( q_i = \{q_{1,i}, q_{2,i}, ..., q_{|q_i|}\} \). Then calculate the final priorities of these requests;

**Step 4.** Select the request \( q_{j,i} \) with the maximal final priority from \( q_i \), and schedule this dwell request successfully;

**Step 5.** set \( t_p \) equal to the finishing time of request \( q_{j,i} \), and set \( i = i + 1 \). If \( t_p > t_q + SI \) or \( i > L \), go to step 6, else return to step 2;

**Step 6.** scheduling analysis ends, get the whole scheduled dwell requests and time index \( t_p \).

In the scheduling procedure above, the last dwell scheduled maybe occupy some span of next scheduling interval, because of the non-preempt characteristic of dwell request. So the time index \( t_p \) need to be reserved for the analysis in the next scheduling interval.

4. Scheduler Load Analysis

4.1. Time Load of Scheduler

In this section, only search and track tasks are analyzed for scheduler workload, as the third kind of task is so few that its workload can be omitted. According to the task model in section 2, we can get the ratio of the \( i^{th} \) search task occupying radar system time is

\[
\zeta_i^T = \frac{B_{i_0} \cdot \Delta t_i^T}{P_{i_0}^T} \quad (i = 1, 2, ..., N)
\]  

(15)

where the parameter \( \zeta_i^T \) denotes the search task consumes system resources ratio averagely. Because the phased array radar can track multiple targets and search the given region at the same time with the TAS(Track And Search) mode, the occupy ratio of the \( i_k \) target track task can be expressed as

\[
\zeta_{i_k}^T = \frac{\Delta t_{i_k}^T}{\Delta T_{ik}^T} \quad (k = 1, 2, ..., K; i_k = 1, 2, ..., M_k)
\]  

(16)

Thus, the total time ratio of all search and track task, called as the scheduler time load is

\[
\zeta = \sum_{j=1}^{N} \zeta_j^T + \sum_{k=1}^{K} \sum_{i_k=1}^{M_k} \zeta_{i_k}^T
\]  

(17)

Inequality \( \zeta \leq 1 \) denotes the scheduler load is rational, but this is only a necessary condition for scheduling all requests. Even if \( \zeta \leq 1 \), some requests still may be deleted for the confliction in time. But \( \zeta > 1 \) denotes scheduler overloaded, which indicates some requests deleted definitely in the process of task schedule.
4.2. Adjustment Method for Scheduler Overload

Assuming current scheduler time load is $\zeta < 1$, a new task (search or track), which load is $\zeta_0$, adds in the scheduler. If $(\zeta + \zeta_0) \leq 1$, this task can be permitted to be scheduled; otherwise this task will result in overload. In this instance, we need some adjustment strategy to tradeoff between this task and others, such as reducing the time load of this task or others.

Assuming the time load of the appending task after adjustment is $\zeta^\prime$, other target track sample ratio after adjustment is $\Delta T_{iT}^\prime$, each search frame time is $P_i^\prime$, those parameters must satisfy

$$\sum_{i=1}^{N} B_n \cdot \Delta t_i^\prime / P_i^\prime + \sum_{k=1}^{N} \sum_{i=1}^{M_k} \Delta t_i^\prime / \Delta T_{iT} + \zeta_0^\prime \leq 1$$

Equation (18) indicates the scheduler load after adjustment still need be less than 1. With above strategy, the time load of radar system can be kept in a rational scope. Note that, besides search frame and track sample ratio, the other task parameters can also be adjusted with this method.

5. Simulation and Analysis

5.1. Simulation Parameters

According to the radar task models proposed, we select the main four work modes of phased array radar, which are search, track, track confirmation and track loss tasks. Furthermore there are two types for search tasks, which are volume search and horizon search. For the track tasks, there are also two types adopted, normal track and precise track. And the ratio of target number with above two types of track tasks is 4:1. The other parameters are shown in Table I, where the period denotes the search frame time for the search tasks or the tracking sample interval for the tracking tasks. Simulation time is 12 second, and the scheduling interval is set to be 50 ms. The track confirmation tasks arise while a new target is detected, and the track loss tasks are produced randomly with a given probability 1%.

In our simulation, capture time of each target is chosen randomly between start time of simulation and the first track sample period, and the disappearing time is assumed to be the end of simulation. The results below are obtained through 100 times simulations.

5.2. Simulation Results and Analysis

We select the missed deadline ratio (MDR) as the main evaluation parameter for the scheduling algorithm. The MDR is defined as the ratio between the number of all deleted dwell requests and the total number of all requests. It is inversely proportional to the scheduling success ratio (SSR). The lower the MDR is, the higher the SSR is. The scheduling algorithm proposed in this paper is compared with the traditional method in which all dwell requests are scheduled by the function priority, that is the dwell request with the highest function priority is processed first.

Firstly, the fixed task parameters are selected for the scheduling processing, where the frame time of volume search and horizon search are 4s and 2s respectively, the sample interval of normal track and
precise track are 1s and 0.5s. Then fig.2(a) shows the time load of scheduler varies with the target number tracked. When the target number equals to 50, the workload of radar scheduler is saturated.

### Table 1 Phased Array Radar Task Parameters

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Types</th>
<th>Function Priority</th>
<th>Dwell Length</th>
<th>Time Window</th>
<th>Beam Number</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Track Confirmation</td>
<td>6</td>
<td>4 ms</td>
<td>30 ms</td>
<td>1</td>
<td>—</td>
</tr>
<tr>
<td>2</td>
<td>Precise Track</td>
<td>5</td>
<td>2 ms</td>
<td>30 ms</td>
<td>1</td>
<td>0.5~1s</td>
</tr>
<tr>
<td>3</td>
<td>Normal Track</td>
<td>4</td>
<td>4 ms</td>
<td>50 ms</td>
<td>1</td>
<td>1~2s</td>
</tr>
<tr>
<td>4</td>
<td>Track Loss</td>
<td>3</td>
<td>4 ms</td>
<td>50 ms</td>
<td>4</td>
<td>—</td>
</tr>
<tr>
<td>5</td>
<td>Horizon Search</td>
<td>2</td>
<td>8 ms</td>
<td>—</td>
<td>100</td>
<td>2~2.5s</td>
</tr>
<tr>
<td>6</td>
<td>Volume Search</td>
<td>1</td>
<td>4 ms</td>
<td>—</td>
<td>400</td>
<td>4~5s</td>
</tr>
</tbody>
</table>

Furthermore, fig.2(b)(c) gives the scheduling results without adjustment when the system is overloaded. It is shown that the traditional method can keep the high function priority tasks being scheduled effectively. But large numbers of low function priority tasks are deleted. The proposed HPEDF algorithm can keep almost all track tasks scheduled, while deleting fewer search tasks.

Secondly, the parameters of the search and track tasks can be changed adaptively from table 1 when the target number is more than 50, showed in fig.3(a). According to eq.(18), the search frame time of volume search is adjusted firstly when the radar system is overloaded. As the search frame time is increased to 5s, the scheduler can work normally with the no more than 70 targets. Furthermore, the frame time of horizon search can be adjusted when the target number is between 70 and 90. Last, the frame time of normal track and precise track can be increased for the more target number, and the time load of the system doesn’t exceed its upper limit 1 all the time.

According to aforementioned parameters adjustment scheme, scheduling results are shown in fig.3(b)(c). Comparing fig.3 with fig.2, an improvement can be seen obviously. The proposed algorithm can schedule all requests, while the traditional method still needs to delete a few search tasks.

From above simulation results and analysis, some conclusions can be drawn: (1) the adjustment scheme can effectively improve the performance of radar scheduler; (2) the proposed algorithm can synthetically consider the function priority and the relative deadline, so its performance is better than the traditional
method.

![Graphs showing scheduling results with parameter adjustment, MDR of search task, and MDR of track task.](image)

Fig. 3 Scheduling Results with Parameter Adjustment

6. Summary and Conclusion

Many existing phased array radar systems still adopt inefficient or even non-real-time resource management technique, such as FIFO-like or cyclic-executive-like scheduling algorithms. As a result, much radar resource is wasted without significant performance improvement. Based on real-time theory, this paper aims at the essential issue for the design of modern phased array radar resource management - task scheduling. A novel real-time task model is built and corresponding scheduling algorithm is proposed for phased array radar systems. Simulation results show that the proposed algorithm can effectively schedule radar tasks in a real-time fashion with optimal performance.

References