Component-Oriented Radars with Probabilistic Timing Guarantees

Chin-Fu Kuo, Ya-Shu Chen, Tei-Wei Kuo, Senior Member, IEEE,
Phone Lin, Senior Member, IEEE, and Cheng Chang

Abstract—In recent years, many modern phased-array radars are built with commercial off-the-shelf components, and the functions of many hardware components are reimplemented by software modules. In such systems, radar tasks could be modeled as distributed real-time tasks which require end-to-end deadline guarantees and have precedence constraints. Different from most previous work on either algorithms with restrictions in resource utilization or heuristics without analytical ways for schedulability guarantees, the objective of this paper is to propose a joint real-time scheduling algorithm for both transmitter/receiver and signal processor workloads with an analytical framework for offline probabilistic analysis and online admission control. The strength of our approach is verified by analysis results and a series of experiments based on a real phased-array radar for air defense frigates [18].

Index Terms—Phased-array radar, real-time task scheduling, probabilistic performance guarantee, distributed systems, dwell scheduling.

1 INTRODUCTION

A multifunction phased-array radar must search and track targets or vehicles in its surveillance space in a real-time fashion. The most essential part inside a phased-array radar is an efficient scheduler implemented in a radar control computer (RCC) to manage the limited radar resource to gain the maximum performance. However, because of hardware constraints and insufficient knowledge of real-time technology, the scheduler is often designed with non-real-time resource scheduling mechanisms, such as FIFO scheduling [6]. As a result, many resources are wasted with a very limited guarantee on system performance.

In recent years, many modern phased-array radars are no longer built in a complicated hardware system with everything wired. Instead, engineers are now building phased-array radars with commercial off-the-shelf (COTS) components, and the functions of many hardware components are now reimplemented by software modules [10]. The development of component-oriented signal processors is strongly influenced by the Rapid prototyping of Application Specific Signal Processors (RASSP) program lead by the US Department of Defense [33]. The RASSP program formalized an engineering process for developing a signal processor (SP) in order to reduce the total product development time and cost by a factor of four.

With component-oriented phased-array radar architecture, radar tasks can be modeled as distributed real-time tasks which require end-to-end deadline guarantees and have precedence constraints. The task scheduling problem for component-oriented phased-array radars is often complicated by the existence of multiple processing units in some components, such as those in the SP. Real-time scheduling problems have been analyzed for different architectural assumptions (such as those for multiprocessor scheduling [3], [13], scheduling with end-to-end deadlines [12], [21], imprecise computation [26], and probabilistic performance guarantee [2], [11]), and many excellent scheduling algorithms have been proposed (such as those for independent task scheduling [27], [29], task synchronization [34], multiframe scheduling [30], rate-based scheduling [20], [28], [37]). However, little work addresses the unique problem for real-time radar task (or dwell) scheduling, especially when complicated real-time resource allocation issues are mixed with reliability and cost issues.

The task models and the work presented in [4], [18], [20] are among the few closely related to dwell scheduling at the RCC level, where a dwell is defined as the entire process from transmitting a radar beam to saving of the digital signals in SP. Although researchers and engineers have started exploring real-time dwell scheduling at RCC, the proposed algorithms are mainly variations of the Partial Template algorithm [6]. The Partial Template algorithm is the most popular one in dwell scheduling, in which a portion of each scheduling interval is reserved for the minimum operation, while the rest is open for the competition or under any priority-driven scheduling to satisfy immediate operational properties and equipment constraints. Different from most previous work on either algorithms with restrictions in resource utilization (and conservative schedulability guarantees, such as those based on Partial Template) or heuristics without analytical methodologies for schedulability guarantees (such as many priority-driven algorithms [21], [25], [35]), the objective of this paper is to propose a joint real-time scheduling...
algorithm for both transmitter and receiver (TR) and SP workloads with an analytical framework for offline probabilistic analysis and online admission control.

While nonpreemptible task scheduling with end-to-end deadlines and precedence constraints is shown to be NP-hard in the literature [15], [17], different heuristics on multistage scheduling are proposed [8], [17], [38]. Distinct from the past work, this paper aims at proposing a real-time scheduling algorithm at RCC to have joint considerations of TR and SP workloads. In addition, we propose an analytical framework for offline probabilistic analysis and online admission control to balance the hardware cost and the performance guarantee. This is motivated by observations in many radar systems implementations that the processing of dwells could miss their deadlines occasionally [7], [24]. We first present a task model for a typical phased-array radar and then an abstraction of a component-oriented phased-array radar. A priority-driven scheduling algorithm based on the given probabilistic guarantees of radar tasks is proposed for TR workloads, and an analytic method is presented to derive deadlines for workloads in the SP based on the given probabilistic guarantees of radar tasks. SP scheduling is proposed based on the well-known rate-based multiprocessor algorithm [5]; however, no task migration or preemption is allowed. The schedulability test of the proposed joint scheduling algorithm (i.e., TR and SP scheduling) is presented for offline probabilistic analysis and online admission control. The strength of our approach is verified by analysis results and a series of simulation experiments based on a real phased-array radar for air defense frigates [18]. It is shown that the proposed algorithm could significantly reduce the hardware cost and, at the same time, provide different probabilistic guarantees for radar tasks. We must point out that the proposed approach could be applied to some scheduling problems in distributed systems, not only limited to the radar system.

The rest of this paper is as follows: Section 2 defines the hardware configuration and formally defines the workload of a typical phased-array radar. Section 3 proposes our TR and SP scheduling algorithm with probabilistic guarantees. In Section 4, a series of simulation experiments are done based on a real example system. Section 5 is the conclusion.

2 HARDWARE CONFIGURATION AND WORKLOAD CHARACTERISTICS

2.1 Hardware Configuration

A component-oriented phased-array radar consists of several important modules: radar control computer (RCC), signal processor (SP), beam steering controller (BSC), receiver, antenna, and transmitter, where an SP consists of an analog signal processor (ASP), a signal processing computer (SPC), and various processing units, as shown in Fig. 1 [6], [18], [31]. RCC schedules dwell transmissions in a real-time fashion by sending the SPC commands. When the SPC receives commands from the RCC for radar beam transmissions, it issues commands to the BSC and the transmitter for radar beam transmissions in specified directions. The antenna and the receiver receive returned signals and pass them to the ASP for analog-to-digital signal processing. The digital signals are saved at the input buffer unit (IBU) for later processing. The processing units of the SP are vector signal processors (VSP), which conduct different types of data processing such as pulse compression, FFT, and other digital signal processing. The data interconnection network (DIN) is for data transmission between the processing units (VSP) and the IBU. All processing units (VSP), the IBU, and the DIN are under the command of the SPC, where the SPC assigns each processing unit a signal processing job to meet the hard deadline of each individual job. Commands in an SP go through another channel, such as VMEbus or Fibre Channel. In many real radar systems, the interconnection bus between the TR and the SP has an extremely high bandwidth such that it presents no trouble in supplying required data from the TR subtask to the corresponding SP subtask. Furthermore, each VSP of the SP retrieves its data directly from the IBU such that no shared memory architecture is needed for many SP designs. In this paper, we focus our study on the scheduling of TR and SP subtasks, where nonpreemption of TR subtasks is the most critical issue in the scheduling.

A dwell is defined as the process from the transmitting of a radar beam by the transmitter to the saving of the corresponding returned digital signals in the IBU. The entire process is nonpreemptible [6]. RCC schedules tasks in units called scheduling intervals (SI) [6], [19], where the length of a scheduling interval is determined by various factors from the system specifications and usually in tens of milliseconds. In other words, the RCC sends a sequence of commands to the SP for dwell transmissions, and retrieves results from the SP at the beginning of an SI. Besides, the SP must read the output results of the TR (in the IBU) at the beginning of an SI. With the synchronization behavior among the RCC, the TR, and the SP, radar tasks are always considered to arrive at a multiple of SI and have deadlines as multiples of SI.\(^1\)

2.2 Workload Characteristics

A typical phased-array radar can be modeled as a distributed system, where a radar task is decomposed into

---

\(^1\) Some implementations of radar systems ignore the SI constraints. However, the synchronization might be needed virtually at any time.
one TR subtask and one SP subtask. TR and SP subtasks must be executed in the TR and the SP, respectively. The execution time of a TR subtask is defined as the length of the corresponding dwell, where the *dwell length* is defined as the duration in executing a dwell. TR subtasks are nonpreemptible. Moreover, the returned signals of a dwell are processed by one of the VSPs in the SP. We say that the SP subtask corresponding to the dwell is executed on one VSP and handles the returned signals. The processing of the returned signals for each dwell is done by one of the VSPs in the SP. SP subtasks are nonpreemptible and cannot migrate among VSPs. The processing time of the returned signals for a dwell is referred to as the execution time of the corresponding SP subtask.

There exists a precedence constraint between a TR subtask and its corresponding SP subtask because the SP subtask processed the returned signals resulting from the execution of the TR subtask. An SP subtask could not start to execute until its prior TR subtask finishes. A task \( \tau_i \) in a phased-array radar system has an end-to-end deadline, which is the time for the completion of its corresponding SP subtask (from the release of its corresponding TR subtask). Moreover, the worst execution times of TR and SP subtasks of \( \tau_i \) depend on the characteristics of their corresponding dwell or radar task.

A phased-array radar could have at least three kinds of work: search, track confirmation, and track.

- **Search**: A phased-array radar must scan its surveillance space periodically for targets of interest, e.g., in terms of horizon search or long range search. This scan must be done in a hard real-time fashion. Such hard real-time searches are called *high-priority search (HS) tasks* in this paper. Low-priority search (LS) tasks, such as normal volume search, are conducted when there are free system resources available after highly critical tasks are serviced.

- **Track Confirmation**: When a target of interest is detected by RCC (because of reflected signals), to verify its presence, a track confirmation (TC) is issued in the direction of the target for the detected target to verify its presence. A track confirmation must be done in a hard real-time fashion to identify targets of interest. Operators might also choose to initiate track confirmations because of various reasons. For example, targets of interest might be discovered based on experiences of operators.

- **Track**: Once a target is identified, a sequence of “semiperiodic” normal track (NT) dwells are issued to track the target. The word *semiperiodic* means “periodic” but with dynamically changing periods. The tracking of a target might need to go into a precision track (PT) task, which is also “semiperiodic,” because the tracking of the target needs better precision. The reason for tracking being semiperiodic is that the distance between every two consecutive tracking executions for a target depends on many factors, such as target type, target position, target speed, etc. The transition of a normal track to a precision track may be requested by the operators [6]. High-precision track (HPT) tasks may be initiated by the operators for many purposes, such as missile guidance. They also need to be done in a hard real-time fashion.

Different phased-array radars have different system specifications and parameters, e.g., different search frame times for different search modes and different tracking rates for different tracking modes. A typical workload might consist of high-priority search tasks, track confirmation tasks, normal track tasks, precision track tasks, high-precision track tasks, and low-priority search, as shown in Table 1, where 1 is the highest priority and 6 is the lowest priority [18]. High-priority search is usually the most important task type because it is for the searching of targets in the most critical surveillance space. track confirmation is the second important task type because it is to confirm the appearance of a target. The priority setting of high-precision track, precision track, and normal track is defined because of their precision requirements in target tracking. Low-priority search is the less important task type because it is intended to search spaces with less importance.

High-priority and low-priority search tasks are usually referred to as *search tasks*. Normal track, precision track, and high-precision track tasks are usually referred to as *track tasks*. In this paper, the scheduling of low-priority search tasks is not discussed because they are executed only when free system resources are available. For the rest of this paper, any symbols with subscripts \( S \), \( C \), and \( T \) are those for high-priority search, track confirmation, and track tasks, respectively. As shown in Table 1, a high-priority search task must issue \( B_S \) beams every \( P_S \) time units, where \( c_{S,1} \) is the dwell length and \( c_{S,2} \) is the execution time of a corresponding SP subtask. The relative end-to-end deadline of each beam of the high-priority search task is \( D_S \). For each target of interest, a track confirmation task is issued with a relative end-to-end deadline \( D_C \), a dwell length \( c_{C,1} \), and an execution time \( c_{C,2} \) of a corresponding SP subtask. Once the target is identified, a sequence of semiperiodic track dwells is issued to track the target. Each track task issues a sequence of semiperiodic track dwells to track a target. The period of a track task is bounded by a lower bound \( P_T \) and an upper bound \( P_T^h \). Besides, a track task has a dwell length \( c_{T,1} \) and requires the processing time \( c_{T,2} \) of the returned signals in SP. The end-to-end relative deadline of a track task is \( D_T \).

### Table 1

<table>
<thead>
<tr>
<th>Task Types</th>
<th>Timing Constraints</th>
<th>Periodic</th>
<th>Deadline Type</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-Priority Search</td>
<td>((c_{S,1}, c_{S,2}, D_S, P_S))</td>
<td>periodic</td>
<td>hard deadline</td>
<td>1</td>
</tr>
<tr>
<td>Track Confirmation</td>
<td>((c_{C,1}, c_{C,2}, D_C))</td>
<td>periodic</td>
<td>hard deadline</td>
<td>2</td>
</tr>
<tr>
<td>High-Precision Track</td>
<td>((c_{T,1}, c_{T,2}, D_T, P_T^h, P_T))</td>
<td>semiperiodic</td>
<td>soft deadline</td>
<td>3</td>
</tr>
<tr>
<td>Precision Track (PT)</td>
<td>((c_{P,1}, c_{P,2}, D_P))</td>
<td>periodic</td>
<td>soft deadline</td>
<td>4</td>
</tr>
<tr>
<td>Normal Track (NT)</td>
<td>((c_{N,1}, c_{N,2}, D_N, P_N))</td>
<td>periodic</td>
<td>soft deadline</td>
<td>5</td>
</tr>
</tbody>
</table>

### 3 Probabilistic Schedulability Guarantees under Two-Stage Scheduling

#### 3.1 Overview

This research is motivated by the needs of a proper scheduling algorithm with an analysis framework for radar...
systems that have different performance guarantees for different radar tasks. It is to avoid unnecessary deployment of a large number of hardware equipments in common practice, due to some pessimistic performance analysis of radar systems. In this section, we shall propose a two-stage scheduling algorithm to provide different levels of guarantees for different radar tasks in a probabilistic fashion. An analytic framework for probabilistic performance guarantees will be derived so that the requirements of hardware equipments could be better estimated.

Component-oriented phased-array radars under consideration have two major components, as shown in Fig. 1: TR and SP. Each radar task \( \tau_i \) is modeled as a sequence of two subtasks \( \tau_{i,1} \) and \( \tau_{i,2} \), where \( \tau_{i,1} \) (referred to as a TR subtask) executes in TR for dwell transmission, and \( \tau_{i,2} \) (referred to as an SP subtask) is the processing of the returned signals in SP. The hardware configuration of a phased-array radar can be abstracted as a chain of two processor groups, as shown in Fig. 2. Because TR only has one BSC, one transmitter, and one receiver, TR must adopt nonpreemptible uniprocessor scheduling because dwell transmissions must be done one after another. We propose to schedule TR subtasks in a priority queuing model (please see Section 3.2.) on the first processor group, the TR \( p\)-group, where the TR \( p\)-group has only one processor. The returned signals of TR subtasks will be buffered at the IBU of the SP for processing. We propose to schedule SP subtasks processing returned signals in the IBU in a distributed rate-based real-time scheduling algorithm (please see Section 3.3.) on the second processor group, the SP \( p\)-group. Because signal processing on each VSP is also nonpreemptible, no migration of task execution among processors (i.e., VSPs) in the SP \( p\)-group is allowed. The number of VSPs in the SP varies, depending on the signal processing needs of a radar system.

Let each task \( \tau_i \) be given a probability threshold \( \phi_i \) such that the phased-array radar must guarantee that a specified portion of its tasks complete before their deadlines if the task is admitted. (The admission control policy will be presented later.) That is, \( \Pr[R_i \leq D_i] \geq \phi_i \), where \( R_i \) is the completion time (relative to its arrival time) of a radar task \( \tau_i \), and \( D_i \) is a given deadline (relative to its arrival time). When the processing of a dwell (including the work done by TR and SP subtasks) misses its deadline, it is simply dropped in the system. In other words, if a TR subtask misses its deadline, the corresponding SP subtask is removed. However, if the deadline-missing problem is serious, then we will lose the tracking of the corresponding track task. It would result in the deletion of the corresponding track task. However, the search tasks in a real system will locate the target again as a new one [6].

In this paper, we first assume that TR subtasks (and their corresponding radar tasks) have Poisson distributions on their arrivals (with a “control parameter” on the variance for analysis and performance evaluation). Let the inter-arrival times of each radar task \( \tau_i \) in the system be generally distributed with a density function \( f_i \) and a rate \( \lambda_i \). Each task \( \tau_i \) has a deadline \( D_i \) relative to its arrival time, referred to as the relative deadline. The completion time of a task \( \tau_i \) relative to its arrival time is referred to as the relative completion time. The technical problem in the scheduling of TR and SP subtasks is how to assign TR and SP subtasks deadlines for real-time scheduling. First, we propose an analysis framework in Section 3.2 to derive deadlines for SP subtasks. The analysis framework consists of two parts: The first part is the analysis of the completion time of TR subtasks, and the second part is the schedulability analysis of SP subtasks. The analysis framework and the RCC scheduling strategy are then further extended without the constraints on the synchronization behavior of RCC and SP. For instance, when certain hardware implementation constraints are lifted, RCC no longer needs to synchronize commands and results with SP in the beginning of each SI, and \( D_{i,1} \) is not necessarily a multiple of SI. With the removing of the constraints, we surmise that the number of VSPs needed could be further reduced. Related experiments are conducted in Section 4.

3.2 Probabilistic Guarantees for Beam-Transmission Scheduling

Given a probabilistic guarantee \( \Pr[R_i \leq D_i] \geq \phi_i \) for a task \( \tau_i \), the first thing is to derive a bound \( D_{i,1} \) on the completion time of its TR subtasks based on the priority queuing model. We shall first derive the average waiting time for a TR subtask and then do a Laplace transform and quadratic differentiation to obtain the variance of the waiting time distribution. The bound of the relative completion time of TR subtasks is obtained by looking up over the standard normal distribution table [16], based on the derived mean and variance.

3.2.1 The Average Waiting Time for a Beam Transmission

Different radar systems have different priority assignments for different radar tasks [18]. We assume that the priority of a TR subtask or an SP subtask is inherited from its corresponding radar task. When a TR subtask is under processing for a beam transmission, it is nonpreemptable. We adopt a priority queuing model for TR-subtask scheduling, where the TR \( p\)-group services TR subtasks with higher priorities first, as shown in Fig. 3. A queue with a smaller index has a higher priority for TR services. Each task is associated with a queue, where a task belonging to a
task type of a higher priority is given a queue of a higher priority (please see Table 1). Each priority queue has only TR subtasks belonging to the same task, and each queue is serviced in a first-in-first-out (FIFO) fashion. When two tasks are of the same type, e.g., High-Precision Track, their priority order could be set arbitrarily.

Given a probabilistic guarantee \( P_{TR,SP}(D_i) \geq \phi_i \) for a task \( \tau_i \) (where \( P_{TR,SP}(x) \) denotes \( \Pr[R_t \leq x] \) for the completion time of a task), the objective of this section is to derive the average waiting time for a TR subtask and \( D_i \) that satisfies \( P_{TR}(D_i,1) \geq \phi_i \) (where \( P_{TR}(x) \) denotes \( \Pr[R_t \leq x] \) for the completion time of a TR subtask) based on the priority queueing model. Note that the two functions \( P_{TR}(D_i,1) = \Pr[R_t \leq D_i] \) and \( P_{TR,SP}(D_i) = \Pr[R_t \leq D_i] \) are defined to simplify the presentation. Once \( D_i \) is determined for TR subtasks of \( \tau_i \), the deadline for SP subtasks of \( \tau_i \) is set as \((D_i - D_{i,1})\), where \( D_{i,1} = \left(\frac{1}{\phi_i}\right)SI \) for the smallest \( x \) that satisfies \( P_{TR}(x) \geq \phi_i \), where \( P_{TR}(x) = \Pr[R_t \leq x] \).

We model each radar task \( \tau_i \) in the system with a Poisson arrival pattern with the rate \( \lambda_i \). We adopt Poisson arrivals in the analysis for two reasons: 1) As shown in the literature, e.g., [16], a periodic or semiperiodic task has the memoryless property of the Poisson process. 2) In many real radar system implementations/studies, e.g., [11], [24], Poisson arrivals are observed for the arrival patterns of the modeling of TR subtasks. The service time of TR subtasks of \( \tau_i \) has a general service time distribution with the mean \( c_{i,1} \). Each task \( \tau_i \) has a fixed relative deadline \( D_i \). An M/G/1 nonpreemptible priority queueing system is adopted to model the behaviors of TR subtasks and to derive their average waiting time. Suppose that the queue with a smaller index has a higher priority, and queue \( Q_i \) is for radar task \( \tau_i \), where \( 1 \leq i \leq n \), and \( n \) is the number of radar tasks. We can derive the average waiting time for each task type, as follows, based on an M/G/1 nonpreemptible priority queueing system [22].

Since the arrival of each TR subtask is a Poisson arrival, any TR subtask arrival (of the \( n \) tasks) also forms a Poisson process with the rate \( \Lambda = \lambda_1 + \lambda_2 + \ldots + \lambda_n \). Let \( C \) be the service time of an arbitrary TR subtask (of the \( n \) tasks). The expected value of \( C \) (i.e., \( E[C] \)) and the second moment of \( C \) (i.e., \( E[C^2] \)) can be derived by the following equations:

\[
E[C] = \frac{\lambda_1}{\Lambda} c_{1,1} + \frac{\lambda_2}{\Lambda} c_{2,1} + \ldots + \frac{\lambda_n}{\Lambda} c_{n,1},
\]

\[
E[C^2] = \frac{\lambda_1}{\Lambda} c_{1,1}^2 + \frac{\lambda_2}{\Lambda} c_{2,1}^2 + \ldots + \frac{\lambda_n}{\Lambda} c_{n,1}^2.
\]

Let \( W_{\tau_i} \) be the waiting time for a TR subtask of \( \tau_i \) in the queue. We can obtain the expected value of \( W_{\tau_i} \) as follows:

\[
E[W_{\tau_i}] = \frac{E[C]}{1 - u_i},
\]

where

\[
u_i = \lambda_1 \cdot c_{1,1} + \lambda_2 \cdot c_{2,1} + \ldots + \lambda_n \cdot c_{n,1}.
\]

Consequently, the expected value of the time \( W_{\tau_i} \) spent by a TR subtask of \( \tau_i \) in the system is \( E[W] = E[W_{\tau_i}] + \frac{1}{\lambda_i} \) and the expected value of the time \( W \) spent by an arbitrary TR subtask is \( E[W] = E[W_{\tau_i}] + \frac{1}{\lambda_i} \).

3.2.2 The Deadline Determination of a Beam Transmission

The purpose of this section is to derive the deadline \( D_i \) for TR subtasks of each radar task \( \tau_i \) based on the distribution of the waiting time of \( \tau_i \). Let \( f_{q;i}(x) \) be the density function of the waiting time \( W_{\tau_i} \) for each TR subtask of radar task \( \tau_i \), and \( F_{q;i}(s) \) the Laplace transform of \( f_{q;i}(x) \). Based on the results in [22], \( f_{q;i}(s) \) could be obtained as follows:

\[
f_{q;i}(s) = \frac{(1 - \rho)[s + \lambda_H - \lambda_H G_H^*(s)] + \lambda_L[1 - B^*_i(s + \lambda_H - \lambda_H G_H^*(s))]}{s - \lambda_H + \lambda_L B^*_i(s + \lambda_H - \lambda_H G_H^*(s))},
\]

where \( B^*_i(s) \) is the Laplace transform of the density function for the service time of each TR subtask of \( \tau_i \).

\[
\lambda_H = \sum_{k=1}^{i-1} \lambda_k,
\]

\[
\lambda_L = \sum_{k=i+1}^{n} \lambda_k,
\]

\[
B^*_H(s) = \sum_{k=1}^{i-1} \lambda_H B^*_k(s),
\]

\[
B^*_L(s) = \sum_{k=i+1}^{n} \lambda_L B^*_k(s),
\]

\[
G_H^*(s) = B^*_H(s + \lambda_H - \lambda_H G_H^*(s)),
\]

and \( \rho = \sum_{k=1}^{n} \lambda_k \cdot c_{k,1} \).

Based on results in [39], we have

\[
E[W_{\phi}^k] = (-1)^i \frac{df_{q;i}(s)}{ds}|_{s=0},
\]

and the variance \( \upsilon_{\phi} \) of \( W_{\phi} \) can be obtained by the following equation:

\[\upsilon_{\phi} = E[W_{\phi}^2] - (E[W_{\phi}])^2.\]

We use the normal distribution \( N(\cdot) \) with the mean \( E[W_{\phi}] \) and the variance \( \upsilon_{\phi} \) to estimate \( D_i \) such that \( P_{TR}(D_i,1) \geq \phi \) by a table lookup [16], where \( P_{TR}(x) = \Pr[R_t \leq x] \). The normal distribution is adopted for the modeling of SP workloads for two reasons: 1) Normal distributions have been adopted by researchers in the modeling of SP workloads in radar system implementations/studies, e.g., [14]. 2) Normal distributions have been used in the modeling of natural behaviors in many domains. As pointed out in the literature, e.g., [9], [36], even though the distribution of a pattern does not fit a normal distribution, tests based on a normal distribution usually give good approximation results.

The analysis proposed in this and the last sections provides a way to derive a lower bound \( D_{i,1} \) on the relative deadline of a TR subtask of radar task \( \tau_i \). As shown in Fig. 4,
an upper bound $D_{i,2}$ on the relative deadline of a corresponding SP subtask is $D_{i,2} = (D_i - D_{j,1})$. The smaller $D_{i,1}$ is, the larger $D_{i,2}$. With a larger value for $D_{i,2}$, more slack is provided in the processing returned signals coming from the work done by the corresponding TR subtask. In other words, fewer VSPs might be needed for the SP $p$-group. Note that if a TR subtask finishes before $D_{i,1}$, then its returned signals will be left at the IBU of the SP on time for processing, and the corresponding SP subtask is ready for RCC scheduling. If a TR subtask misses its deadline, then the corresponding SP subtask is discarded, and the corresponding radar task misses its deadline. Note that the analytic framework presented in Section 3.2 has considered the probability of deadline violations for radar tasks so that enough hardware equipment is adopted for an acceptable deadline satisfaction probability for radar tasks. However, we must point out that this model does not address the cascading of deadline violations. Furthermore, when a TR subtask finishes before $D_{i,1}$, the ready time of the corresponding SP subtask is still set at the end of $D_{i,1}$. Note that the RCC sends a sequence of commands to the SP for signal processing in the beginning of an SI and also retrieves results from the SP. The SP $p$-group must 100 percent guarantee the on-time services of any SP subtasks of $\tau_i$ that have their returned signals stored in the IBU before the completion-time bound $D_{i,1}$. Note that when the probability threshold is extremely close to 100 percent, the motivation of this work will no longer retain. In other words, the number of required VSPs would not be significantly reduced. It is because the relative deadlines of TR subtasks would be large such that it becomes hard to guarantee the schedulability of SP subtasks in SP scheduling. In fact, when the probability threshold is extremely close to 100 percent, the behavior of the proposed framework would be very close to that for the Effective Deadline Algorithm (please see Fig. 6b), especially when there is no SI consideration.

### 3.3 Rate-Based Scheduling for Signal Processing

#### 3.3.1 Reservation Ratios of SP Subtasks

The idea of generalized processor sharing (GPS) was first proposed by Parekh and Gallager [32] in the context of rate-based flow and congestion control at network gateway nodes. GPS-based scheduling is a work-conserving scheduling mechanism, in which each task $\tau_i$ is given a positive real number $\theta_i$, the reservation ratio (reservation ratio) such that $\tau_i$ is guaranteed to be served at a rate of

$$g_i = \frac{\theta_i}{\sum_j g_j}.$$ 

In particular, Spuri et al. [37] proposed an effective GPS-based mechanism called TB server to service tasks under the framework of the EDF scheduling. In [5], a distributed and revised version of the TB algorithm (called multiprocessor constant-bandwidth server (M-CBS)) was proposed for multiprocessor environments, where preemption or migration is allowed.

In this section, we propose to adopt M-CBS for SP scheduling, however, under a more restricted constraint on preemption or migration. We shall determine the reservation ratio for a radar task and propose a scheduling mechanism for SP subtasks. A schedulability test of SP subtasks should also be derived on extending M-CBS for no preemption or migration. Note that results in the previous section derive an upper bound on the relative deadline of an SP subtask, i.e., $D_{i,2} = (D_i - D_{j,1})$, for a given probability to guarantee the schedulability of the corresponding radar task $\tau_i$.

The design of an SP scheduling algorithm must guarantee the processing of all of the returned signals of the TR subtasks of $\tau_i$ if they could arrive at the IBU within their relative deadline $D_{i,1}$. As discussed in Section 2, the period of each track task $\tau_i$ has a lower bound $P^{\tau_i}$ and an upper bound $P^{\tau_i}_{\text{max}}$, depending on its corresponding track type. Once a radar task is admitted, a radar system should keep track of the corresponding target with a specified degree of guarantee. The reservation ratio of each radar task is set as $\theta_i = \frac{\theta_i}{P^{\tau_i}}$, where $\theta_i$ is the execution time of the SP subtask of $\tau_i$, if $P^{\tau_i} \geq D_{i,2}$. Otherwise, the reservation ratio of $\tau_i$ is $\theta_i = \frac{\theta_i}{P^{\tau_i}}$. The rationale behind the assignment is to provide the SP subtasks of $\tau_i$ enough capacity to process its returned signals. Since the period of each search task is a fixed constant, the reservation ratio of a search task $\tau_i$ is $\theta_i = \frac{\theta_i}{P^{\tau_i}}$, if $P_S \geq D_{i,2}$, where $P_S$ is the period of the search task. Otherwise, the reservation ratio of $\tau_i$ is $\theta_i = \frac{\theta_i}{P^{\tau_i}}$. Note that since the tracking of an identified target starts with a track confirmation, then a sequence of normal tracks, and then possibly a sequence of precision tracks, the reservation ratio needed to track a target must be the maximum of the reservation ratios for the corresponding radar tasks of the target (in the confirmation, normal tracking, and precision tracking states) if a guarantee on the tracking of the target is required.

### 3.3.2 SP Scheduling

SP scheduling is based on M-CBS [5], which is a work-conserving scheduling mechanism in which the schedulability of the SP subtasks of each radar task $\tau_i$ is guaranteed with a reservation ratio $\theta_i$. We say that an SP subtask arrives if its corresponding returned signals is in the IBU of the SP. When an SP subtask arrives, its deadline is set based on M-CBS, except that no preemption or migration is allowed. Let $T = T_{i,1}, T_{i,2}, \ldots, T_{i,n}$ denote a collection of radar tasks under scheduling, where each radar task $\tau_i$ has a reservation ratio $\theta_i \leq 1$, for $1 \leq i \leq n$. Suppose that radar tasks in $T$ are indexed in a nonincreasing reservation ratio order, i.e., $\theta_i \geq \theta_{i+1}$, for all $i, 1 \leq i \leq n$, and $\theta(T) = \sum_{\tau_i \in T} \theta_i$. $T^{(\ell)}$ denotes the collection of radar tasks with the $(n - \ell + 1)$ minimum reservation ratios in $T$, i.e., $T^{(\ell)} = T_{\ell}, T_{\ell+1}, \ldots, T_n$. (According to this notation, $T = T^{(0)}$.) For $\ell < n$, $T^{(\ell)}$ denotes the smallest value of $\ell$ that satisfies the following inequality (please see Theorem 2 in the next section), where $M$ is the number of processors, i.e., VSPs, in the SP $p$-group. $Max(npb_h)$ and $Min(d_{j})$ are the maximum execution time and the minimum relative deadline of all SP subtasks, respectively:

$$M \left(1 - \frac{Max(npb_h)}{Min(d_{j})}\right) \geq \min_{\ell=1}^{n}(k-1 + \ell),$$

where $m_k = \frac{\theta(T^{(k+1)})}{\theta(T^{(k)})}$ is the number of processors needed for the collection $T^{(k)}$ of radar tasks. Radar tasks $\tau_1, \tau_2, \ldots, \tau_{n(T)}$ are called high-priority radar tasks, and the rest are called deadline-based radar tasks. Let an SP subtask of an admitted radar task $\tau_i$ arrive at time $t$, the deadline
of the SP subtask is set as $-\infty$ if $\tau_i$ is a high-priority radar task. (The admission control test will be explained in the next section.) Let the arriving SP subtask correspond to the $l$th instance of the SP subtasks of $\tau_i$ for some integer $l > 0$. If the corresponding radar task is a deadline-based radar task, then the deadline of the SP subtask is set as $d_{l,1} = \max\{d_{l,1-1} + \frac{\tau_i}{T}, d_l\}$, where $d_{l,1}$ denotes the absolute deadline of the $l$th instance of the SP subtasks of $\tau_i$ and $d_{l,0} = 0$. The scheduling of SP subtasks is based on the earliest-deadline-first algorithm [27], where the SP subtask with the earliest absolute deadline is scheduled first for any available processor. No preemption or migration is allowed. The scheduling algorithm is called M-CBS without preemption and migration (M-CBS-NPM). Note that $-\infty$ denotes the smallest possible absolute deadline such that SP subtasks of high-priority radar tasks are always scheduled first.

Since M-CBS is designed to schedule tasks with a reservation ratio $\theta_i \leq 1$, each search task $\tau_i$ that could have a reservation ratio $\theta_i$ larger than 1 is split into $n_i$ corresponding radar tasks, i.e., $\tau_{i,1}, \ldots, \tau_{i,n_i}$, where $n_i = \lceil \frac{T}{P_{SI}} \cdot \theta_i \rceil$. The reservation ratio of each corresponding radar task is $\frac{\theta_i}{n_i}$. Instances of TR subtasks of $\tau_i$ and their corresponding SP subtasks are assigned to the $n_i$ corresponding radar tasks in a round robin fashion. Note that a search task $\tau_i$ must issue $B_S$ beams in each period $P_S$. Each radar task $\tau_{i,j}$ (split from $\tau_i$) is considered independently in TR and SP scheduling. The SP of all radar tasks (regardless of whether they are split or not) is as presented in the previous paragraph.

**Example 1.** An example for the joint scheduling of TR and SP subtasks.

Consider the scheduling of radar tasks with five VSPs in SP. Let the system have one high-priority search task ($\tau_{HS}$), two high-priority track tasks ($\tau_{HPT,1}$ and $\tau_{HPT,2}$), and one precision track task ($\tau_{PT,1}$) at some time, e.g., the $k$th SI. The characteristics of the radar tasks is summarized in Table 2. Let each radar task have a probabilistic guarantee to meet 95 percent of the deadlines of its instances. Based on the characteristics, the average utilization of TR could be derived as follows (please see Section 3.2.): $U = \Sigma(\lambda_i \cdot c_{i,1}) = 0.39$. The average queue waiting time and the relative deadlines ($D_{i,1}$ and $D_{i,2}$) for all radar tasks are shown in Table 3. Since the original reservation ratio of $\tau_{HS}$ is equal to $1.685 = \frac{15}{\lambda_{HS}}$, $\tau_{HS}$ is split into 4 ($\lceil 1.125 \cdot 1.685 \rceil = 4$) new tasks $\tau_{HS,1}$, $\tau_{HS,2}$, $\tau_{HS,3}$, and $\tau_{HS,4}$ with a reservation ratio 0.422 ($\frac{1.685}{4} \approx 0.422$). The reservation ratios of the radar tasks are shown in Table 4.

Assume that one instance ($\tau_{HS,s}$) of $\tau_{HS}$, one instance ($\tau_{HPT,1,s}$) of $\tau_{HPT,1}$, and one instance ($\tau_{PT,1,s}$) of $\tau_{PT,1}$ arrive at the beginning of the $k$th SI, and one instance ($\tau_{HS,2,s}$ of $\tau_{HS}$ and one instance ($\tau_{HPT,2,s}$) of $\tau_{HPT,2}$ arrive at the beginning of the $(k+1)$th SI. Based on (1), $\tau_{HS,1}$ is a high-priority search task, and the rest are deadline-based radar tasks. The absolute deadlines of SP subtasks are shown in Table 5. The resulting schedule of SP scheduling at the $k$th SI and the $(k+1)$th SI is shown in Fig. 5.

### 3.3.3 Properties

The purpose of this section is to provide a polynomial-time schedulability test for SP scheduling. It also serves for the admission control of new radar tasks. We summarize related theorems of M-CBS in [5], [23]. Note that each M-CBS server in [5], [23] is a radar task in this paper.

**TABLE 2**

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Arrival Rate $\lambda_i$ (unit: $%$)</th>
<th>Dwell Length in TR, $c_{i,1}$ (unit: $%$)</th>
<th>Processing Time in SP, $c_{i,2}$ (unit: $%$)</th>
<th>Relative Deadline, $D_i$ (unit: $%$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-Priority Search task $\tau_{HS}$</td>
<td>1.125</td>
<td>0.24</td>
<td>1.5</td>
<td>8</td>
</tr>
<tr>
<td>High-Precision Track task $\tau_{HPT,1}$, $\tau_{HPT,2}$, and $\tau_{HPT,3}$</td>
<td>0.6</td>
<td>0.08</td>
<td>0.125</td>
<td>6</td>
</tr>
<tr>
<td>Precision Track task $\tau_{PT,1}$</td>
<td>0.15</td>
<td>0.16</td>
<td>0.25</td>
<td>6</td>
</tr>
</tbody>
</table>

**TABLE 3**

<table>
<thead>
<tr>
<th>Task</th>
<th>Average Waiting Time in the Queue</th>
<th>$D_{i,1}$</th>
<th>$D_{i,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{HS}$</td>
<td>0.050 SI</td>
<td>1 SI</td>
<td>7 SI</td>
</tr>
<tr>
<td>$\tau_{HPT,1}$</td>
<td>0.070 SI</td>
<td>1 SI</td>
<td>5 SI</td>
</tr>
<tr>
<td>$\tau_{HPT,2}$</td>
<td>0.075 SI</td>
<td>1 SI</td>
<td>5 SI</td>
</tr>
<tr>
<td>$\tau_{PT,1}$</td>
<td>0.080 SI</td>
<td>1 SI</td>
<td>5 SI</td>
</tr>
</tbody>
</table>

**TABLE 4**

<table>
<thead>
<tr>
<th>Task</th>
<th>Reservation Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{HS,1}$</td>
<td>0.422</td>
</tr>
<tr>
<td>$\tau_{HS,2}$</td>
<td>0.422</td>
</tr>
<tr>
<td>$\tau_{HS,3}$</td>
<td>0.422</td>
</tr>
<tr>
<td>$\tau_{HS,4}$</td>
<td>0.422</td>
</tr>
<tr>
<td>$\tau_{HPT,1}$</td>
<td>0.15</td>
</tr>
<tr>
<td>$\tau_{HPT,2}$</td>
<td>0.15</td>
</tr>
<tr>
<td>$\tau_{PT,1}$</td>
<td>0.05</td>
</tr>
</tbody>
</table>

**TABLE 5**

<table>
<thead>
<tr>
<th>SP Subtask</th>
<th>Absolute Deadline</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_{HS,1}$</td>
<td>$-\infty$</td>
</tr>
<tr>
<td>$\tau_{HPT,1}$</td>
<td>$SI(k+1) + 0.833 SI$</td>
</tr>
<tr>
<td>$\tau_{PT,1}$</td>
<td>$SI(k+1) + 5 SI$</td>
</tr>
<tr>
<td>$\tau_{HS,2}$</td>
<td>$SI(k+2) + 3.56 SI$</td>
</tr>
<tr>
<td>$\tau_{HPT,2}$</td>
<td>$SI(k+2) + 0.833 SI$</td>
</tr>
</tbody>
</table>
Theorem 1. [5] Given a collection \( T \) of M-CBS servers, let each server be associated with a reservation ratio \( \theta_i \). \( T \) is schedulable by M-CBS with processor migration and preemption allowed if and only if there exists a value for \( k \) that satisfies the following inequality: \( M \geq \min_{k=1}^n \{(k - 1) + m_k\} \), where \( M \) is the number of processors, and \( m_k = \frac{\theta_i}{n_i} \) is the number of the processors needed for a collection \( T^{(k)} \) of radar tasks. Each processor reserves \( \theta_k \) for the radar task \( \tau_k \).

Lemma 1. [23] When a uniprocessor system is scheduled with more than one M-CBS server on the earliest-deadline-first basis, every server meets its deadlines if the sum of the total reservation ratio of all servers is no greater than \( 1 - \frac{\max(npb_i)}{\min(d_j)} \), where \( \max(npb_i) \) and \( \min(d_j) \) are the maximum execution time of any nonpreemptable portion and the minimum of the relative deadlines for all subtasks executed by servers, respectively.

Since no M-CBS server could have a reservation ratio larger than 1, we shall first provide a schedulability test for radar tasks if no radar task has a reservation ratio larger than 1. We then show that the inequality of the schedulability test remains when we split radar tasks with a reservation ratio larger than 1 by the method presented in the previous section.

Theorem 2. Given a collection \( T \) of radar tasks with reservation ratios no larger than 1, \( T \) is schedulable by M-CBS-NPM if and only if there exists a value for \( k \) that satisfies the following inequality: \( M(1 - \frac{\max(npb_i)}{\min(d_j)}) \geq \min_{k=1}^n \{(k - 1) + m_k\} \), where \( \max(npb_i) \), \( \min(d_j) \), and \( m_k = \frac{\theta_i}{n_i} \) are the maximum execution time, the minimum of the relative deadlines of all \( SP \) subtasks, the number of the processors needed for a collection \( T^{(k)} \) of radar tasks, respectively.

Proof. The correctness of this theorem follows from Theorem 1 and Lemma 1 by considering the worst-case blocking cost.

According to M-CBS-NPM, any search task \( \tau_i \) that has a reservation ratio \( \theta_i \) larger than one must be first split into \( n_i = \lceil \frac{1}{\theta_i} \rceil \) tasks. We first show that the tasks split from a search task with \( \theta_i \leq 1 \) could provide enough processing power for the \( SP \) subtasks of the search task in meeting their deadline requirement.

Lemma 2. Given a radar task \( \tau_i \) with a reservation ratio \( \theta_i > 1 \), \( \tau_i \) could be split into \( n_i \) corresponding radar tasks, i.e., \( \tau_{i,1}, \ldots, \tau_{i,n_i} \), and the reservation ratio of each corresponding radar task is \( \frac{\theta_i}{n_i} \). If instances of TR subtasks of \( \tau_i \) and their corresponding \( SP \) subtasks are assigned to the \( n_i \) corresponding radar tasks in a round robin fashion, then the \( SP \) subtasks of the instances of \( \tau_i \) will meet their deadlines under M-CBS-NPM.

Proof. Assume that \( C_i[t_1,t_2] = \sum_{t_1 \leq t_2, d_{i,j,k} \leq t_2} C_{i,j} \) is the maximum execution time demanded by \( SP \) subtasks of a radar task \( \tau_i \) with a reservation ratio \( \theta_i > 1 \) in an interval of time \([t_1, t_2]\), where \( r_{i,j}, d_{i,j} \) and \( c_i \) are the ready time, the absolute deadline, and the execution time of the \( j \)th \( SP \) subtask instance of a radar task \( \tau_i \). Let \( x_{i,t_1,t_2} \) denote the number of \( SP \) subtask instances of radar task \( \tau_i \) that appear in the interval \([t_1, t_2]\), i.e., \( x_{i,t_1,t_2} = \sum_{t_1 \leq t_2} C_{i,j} \). Since there are \( n_i \) corresponding radar tasks for \( \tau_i \), the maximum execution time demanded by \( SP \) subtasks of a radar task \( \tau_i \) in \([t_1, t_2]\) could be written as

\[
C_i[t_1,t_2] = \sum_{k=1}^{n_i} C_{i,j} = x_{i,t_1,t_2} \cdot C_{i,j}.
\]

Since \( SP \) subtasks of \( \tau_i \) are shared evenly among the corresponding radar tasks of \( \tau_i \), the total execution time guaranteed by M-CBS-NPM for the corresponding radar tasks is no less than the demanded execution time:

\[
\sum_{j=1}^{n_i} \sum_{k=1}^{m_k} \left[ (d_{i,j,k} - r_{i,j,k}) \frac{\theta_i}{n_i} \right] \geq x_{i,t_1,t_2} \cdot C_{i,j},
\]

where \( d_{i,j,k} \) and \( r_{i,j,k} \) are the absolute deadline and the ready time of the \( k \)th \( SP \) subtask instance of the corresponding radar task \( \tau_{i,j} \) in \([t_1, t_2]\), respectively. We conclude that the \( SP \) subtasks of the instances of \( \tau_i \) will meet their deadlines under M-CBS-NPM.

Theorem 3. Given a collection \( T \) of radar tasks (where radar tasks with reservation ratios larger one are split based on M-CBS-NPM), \( T \) is schedulable by M-CBS-NPM if and only if there exists a value for \( k \) that satisfies the following inequality:

\[
M(1 - \frac{\max(npb_i)}{\min(d_j)}) \geq \min_{k=1}^n \{(k - 1) + m_k\},
\]

where \( \max(npb_i) \), \( \min(d_j) \), and \( m_k = \frac{\theta_i}{n_i} \) are the maximum execution time, the minimum of the relative deadlines of all \( SP \) subtasks, and the number of processors needed for the radar task set \( T^{(k)} \), respectively.
The correctness of this theorem follows from Theorem 2 and Lemma 2.

Proof. The correctness of this theorem follows from Theorem 2 and Lemma 2.

Theorem 3 could be used for online admission control when the system considers to admit any new radar task, given the number of VSPs in the SP. We must point out that Theorem 3 could also be used for the capacity estimation of the SP when the (maximum) workload of a radar system and its characteristics are given! Engineers could more optimistically estimate the minimum number of VSPs needed for a given workload based on the following corollary. Note that this offline-probabilistic analysis provides the relationship between the number of required VSPs and the number of track tasks that could be supported by the radar system, given the probabilistic guarantee requirements for radar tasks. Note that the workload of the search tasks is fixed in the radar system design. In other words, given a fixed number of VSPs and the probabilistic guarantee requirements for radar tasks, the offline probabilistic analysis provides a bound of radar tasks that could be supported in the system. What the online admission does is to make sure that the maximum number of radar tasks is not over the derived number in the offline-probabilistic analysis based on the number of adopted VSPs.

Corrolary 1. Given a collection \( T \) of radar tasks (where radar tasks with reservation ratios larger than 1 are split based on M-CBS-NPM), the number of VSPs needed for the radar tasks is no less than \( M' \) if \( M' \leq \sum \theta_i \), where \( \theta_i \) is the reservation ratio of a radar task \( \tau_i \).

Proof. The correctness of this corollary follows from the fact that when the number of VSPs allocated for the radar tasks is \( M' \), the maximum processing time supplied by the VSPs is \( M' \cdot (t_2 - t_1) \) in an interval of time \([t_1, t_2]\).

4 PERFORMANCE EVALUATION

The purpose of this section is to evaluate the performance of our proposed methodology, referred to as the Probabilistic Real-Time Scheduling algorithm (PRTS). A simulation model is investigated the performance of the proposed methodology. We compare the schedulability and the performance of PRTS with the Ultimate Deadline Algorithm (UD) [21], [23], the Proportional Deadline Algorithm (PD) [23], the Equal Deadline Algorithm (ED), the Equal Flexibility Algorithm (EQF) [21], the Equal Slack Algorithm (EQS) [21], and the Effective Deadline Algorithm (ED) [21], [23].

The performance of PRTS proposed in this paper is verified and simulated under different probabilistic guarantees (i.e., 91 percent, 93 percent, 95 percent, 97 percent, and 99 percent). We use PRTS_{90} to denote PRTS with a probabilistic guarantee \( \phi \) in this section.

The system parameters for the experiments are based on a real phased-array radar for air defense frigates [18].

Although the radar is not designed for the study of probabilistic performance guarantees, it is adopted for performance evaluation in this paper because it provides a typical radar system workload. Each tested task set has a search task and several track tasks. The search task issues 45 beams for every 40 SI (where 40 SI = 1 sec) and is used for horizon search. We set the dwell length, the execution time in the SP, and the relative deadline of each search task as 0.24 SI, 1.5 SI, and 8 SI, respectively. The dwell length of each track task is 0.16 SI. The execution time of the SP subtask and the deadline of a track task are 0.25 SI and 6 SI, respectively. The time interval between two consecutive instance arrivals for a track task is exponentially distributed with a mean 4 SI. The duration pattern from the arrival time to the corresponding departure time of the each track task is exponentially distributed with a mean 40 SI. The duration pattern from the last departure time to the newly arrival time of each track task is also exponentially distributed with a mean 40 SI. The two parameters are used in the experiments of admission control. The number of the track tasks in a tested task set of the simulations is from 10 to 20. Let \( N_T \) denote the number of track tasks. In other words, the total utilization of the corresponding TR subtasks including the search task varies from 0.67 to 1.07. When the total utilization of TR subtasks is over 1, TR is overloaded. The parameters are summarized in Table 6.

Ten trace files are generated based on the characteristics of each track task, as shown in Table 6. The arrivals of tasks have exponential distribution. Each trace file (of a task set) is simulated for 40,000 SI, and different scheduling algorithms are evaluated. Let \( m_{ij}(IK) \) denote the number of VSPs needed for the simulation of a trace file \( F_{ij} \) of a task set \( TS_i \) under Algorithm IK, where IK could be UD, PD, EQD, EQF, EQS, ED, or PRTS. Let \( m_i(IK) \) denote the number of needed VSPs for the task set \( TS_i \) under Algorithm IK, and \( m_{TS_i}(IK) \) is equal to \( \sum_{j=1}^{10} m_{ij}(IK) \).

There are two primary performance metrics in the experiments: 1) the schedulability of the task sets and the number of VSPs (referred to as \( N_{VSP} \)) needed for different scheduling algorithms, and 2) the number of track tasks admitted into the system. For each algorithm, the schedulability of a simulated task set has to be tested. If an algorithm under simulation can schedule a task set, then we...
The algorithms under evaluation include Max (i.e., UTR; i) dead-
lines of all TR subtasks, respectively. Otherwise, the execution
time and the minimum of the relative deadlines for each task
in the task set could not schedule the task set, regardless of
the number of adopted VSPs in the SP. Given different numbers
of VSPs, we evaluate the performance of the scheduling
algorithm in figures included in this section if the scheduling
algorithm could not schedule the tested task set, regardless
of the number of adopted VSPs in the SP. Given different
numbers of VSPs, we evaluate the performance of the
scheduling algorithm could not schedule the tested task set,
and this algorithm in figures included in this section if the
minimum number of VSPs needed for a scheduling
algorithm could not schedule the tested task set, regardless
of the number of adopted VSPs in the SP. Given different
numbers of VSPs, we evaluate the performance of the
scheduling algorithm could not schedule the tested task set,
and this algorithm in figures included in this section if the
minimum number of VSPs needed for a scheduling
algorithm. Importantly, we do not show the results
including in this section if the scheduling algorithm could not
schedule the tested task set, regardless of the number of
adopted VSPs in the SP. Given different numbers of VSPs,
we evaluate the performance of the scheduling algorithm.

4.1 Analysis Results and Experimental Results

In this section, we show the analysis results based on
Theorem 3 and the results of the simulation.

4.1.1 Analysis Results

This section first investigates the \( N_{VSP} \) values for different
algorithms based on the calculation results using
Theorem 3. The algorithms under evaluation include UD, PD, EQD, EQF, EQS, and PRTS 95%. Let
\( U_{TR}(k) \) denote the total average utilization for the task
set \( TS \), for Algorithm \( I_k \). \( TS \) could not be scheduled by
Algorithm \( I_k \), if the calculated value
\[
U_{TR}(k) > \left(1 - \frac{\text{Max}(npb)}{\text{Min}(d_j)}\right),
\]
where \( \text{Max}(npb) \) and \( \text{Min}(d_j) \) are the maximum execution
time and the minimum of the relative deadlines of all TR subtasks, respectively. Otherwise
(i.e., \( U_{TR}(k) \leq 1 - \frac{\text{Max}(npb)}{\text{Min}(d_j)}\)), we could use Theorem 3
to calculate \( N_{VSP} \) for Algorithm \( I_k \), and this algorithm
could schedule the tested task set \( TS \). Note that the
schedulability of task sets under algorithms for comparison,
i.e., UD, PD, EQD, EQF, EQS, or ED, fails if any
task misses its deadline. It is because the algorithms
could not provide any probabilistic schedulability
guarantee, and the experiments are to provide insights
for real-time task scheduling with probabilistic schedul-
ability guarantees.

As mentioned in Section 2.1, the implementation of radar
systems can ignore the SI constraints. We show the results
of the algorithms with and without SI considerations in
Fig. 6.

The Performance Comparison for Different Algorithms
with SI Considerations. In Fig. 6a, we compare \( N_{VSP} \) for
different algorithms where SI is considered. UD, PD, and
ED can not schedule any tested task set. The reason is that
\( D_{1,1} (D_{1,2}) \) of a task \( t_i \) under UD (PD or ED) is too small to
guarantee the schedule of the TR (SP) subtasks. When
\( N_T < 12 \), PRTS 95%, EQD, EQS, and EQF can schedule the
tested task sets, and the \( N_{VSP} \) values for PRTS 95% are
smaller than that for EQD, EQS, and EQF. The reason is that
\( D_{1,2} \) for PRTS 95% is larger than that for EQD, EQS, or EQF.

The Performance Comparison for Different Algorithms
without SI Considerations. In Fig. 6b, we compare \( N_{VSP} \) for
different algorithms where SI is not considered. We
observe that UD can not schedule any tested task set. EQD
EQS can schedule the tested task sets when \( N_T \leq 12 \).

EQF can schedule the tested task sets only when
\( N_T = 10 \). Although ED and PRTS 95% can schedule the
tested task sets when \( N_T \leq 18 \), PRTS 95% has smaller
\( N_{VSP} \) values. We know that PRTS 95% outperforms the other
algorithms in terms of \( N_{VSP} \) and has the highest guarantee
for schedulability. It is because the 95 percent probabilistic
guarantee results in larger relative deadlines for SP subtasks
in scheduling.

The Effects of Probabilistic Guarantees on PRTS
without SI Considerations. For PRTS, we set up different
probabilistic guarantees when scheduling a task set. In
Fig. 7, we investigate the effects of the probabilistic
guarantees on PRTS when SI is not considered. We set
probabilistic guarantees as 91 percent, 93 percent, 95 per-
cent, 97 percent, and 99 percent. For the situation when SI is
considered, we observe similar phenomena, so these results
for PRTS are not shown. Three phenomena are observed:

Phenomenon 1. When \( N_T \leq 14 \), the effects of the probabil-
istic guarantees are insignificant (i.e., the \( N_{VSP} \) values are
almost the same).

Phenomenon 2. When \( N_T = 16 \), PRTS with the higher
probabilistic guarantee needs more VSPs.

Phenomenon 3. When \( N_T = 18 \), the \( N_{VSP} \) values for PRTS
with different probabilistic guarantee setups are the
same, which are close to 20.

Phenomenon 1 occurs because TR utilization is relatively
low so that the waiting time of a TR subtask is not very
long, even under different probabilistic guarantees.
Phenomenon 1 indicates that when \( N_T \leq 14 \), we could set
the probabilistic guarantee as 99 percent (i.e., no more than
1 percent of task instances misses their deadlines), PRTS can
provide the same schedulability as the other algorithms,
and the \( N_{VSP} \) value for PRTS is the smallest among UD, PD,
EQD, EQF, EQS, ED, and PRTS. (Please see Fig. 6.) Phenomenon 2 shows that PRTS needs more VSPs to provide a higher probabilistic guarantee with the increasing of $NT$ (i.e., $NT = 16$). When $NT = 18$, the system utilization of the TR is about 99 percent. Since the TR is almost fully utilized, different probabilistic guarantees do not affect the $N_{VSP}$ values, as shown in Phenomenon 3. When $NT = 20$ (i.e., the TR utilization = 1.06), and the TR is overloaded, no scheduling algorithms guarantee the schedulability of all tasks without any deadline violation.

Note that there is a gap between analysis results and simulation results. It mainly comes from the considerations of the worst-case workloads of track tasks in the analysis. In other words, although the status of a track task might change among normal track, precision track, and high-precision track, the offline-probabilistic analysis considers the worst-case workload, i.e., the workload for high-precision track.

The Upper Bound of Track Tasks Admitted by Different Algorithms without SI Considerations. Fig. 8 shows the number of track tasks that could be admitted by different scheduling algorithms, given different numbers of VSPs. It is observed that UD does not admit any task. It is because the relative deadlines of SP subtasks under UD are always equal to 0. EQS (/EQF) and EQD (/PD) have the same performance. The number of track tasks admitted by EQS, EQF, EQD, and PD would not increase at some time point because the TR workload is already almost overloaded. PRTS$_{95\%}$ greatly outperforms other scheduling algorithms in general. The only scheduling algorithm with similar performance is ED when the number of VSPs is sufficient for radar signal processing. It is because the deadlines of SP subtasks under PRTS$_{95\%}$ would be as the same as those under ED when the number of VSPs is over 18. It is because the deadlines of SP subtasks under PRTS$_{95\%}$ would be as the same as those under ED when the number of VSPs is sufficient for radar signal processing. Although Fig. 8 shows the number of track tasks that could be admitted by different scheduling algorithms when SI is not considered, similar results are also observed when SI is considered in the analysis.

The Upper Bound of Track Tasks Admitted by Different Probabilistic Guarantees on PRTS without SI Considerations. Fig. 9 shows the number of track tasks admitted by PRTS, given different numbers of VSPs and different probabilistic guarantees. It is observed that PRTS with different probabilistic guarantees has similar performance when the number of VSPs is sufficient for the radar workload. When the number of VSPs is not sufficient, PRTS with smaller probabilistic guarantees tend to have better performance. It is as the same as our expectation. Fig. 9 shows the results without SI consideration. The results with SI consideration are similar to those shown in Fig. 9.

4.1.2 Experimental Results

In the experiments, PRTS always guarantees the schedulability of search tasks and, at the same time, reduces the number of VSPs needed for phased-array radars, compared to UD, PD, EQD, EQF, EQS, and ED. The simulation results for PRTS$_{95\%}$ without SI considerations are shown in Fig. 10. Even when the number of track tasks per SI is 20, 99.96 percent of the executions of track tasks meet their deadline requirements (even though a 100 or even 95 percent guarantee is impossible), where PRTS without SI consideration is applied. Note that the experimental results for PRTS with SI consideration are similar to those without SI consideration.

We conclude that the proposed methodology on probabilistic real-time scheduling and analysis, PRTS, guarantees the schedulability, and reduces the number of VSPs needed for phased-array radars, compared to UD, PD, EQD, EQF, EQS, and ED. PRTS improves the schedulability guarantee of SP significantly. It is shown that PRTS demonstrates a reasonable performance, compared to UD, PD, EQD, EQF, EQS, and ED, where the algorithms may not be suitable in real-time radar scheduling because they cannot guarantee the schedulability of the radar operation requirements, or more VSPs are needed.

5 Conclusion

This paper addresses two important issues on radar scheduling: 1) schedulability guarantees for radar tasks and 2) system capacity estimation. While much existing work suffers from conservative resource allocation
problems, we aim at proposing a joint real-time scheduling algorithm for TR and SP workloads with an analytical framework for offline probabilistic analysis and online admission control. A priority-driven scheduling algorithm is proposed for TR workloads, and an analytic method is presented to derive deadlines for workloads in SP based on the given probabilistic guarantees of radar tasks. SP scheduling is then proposed based on the well-known rate-based multiprocessor algorithm M-CBS, where no task migration or preemption is allowed. We provide different levels of schedulability guarantees in a probabilistic fashion for different radar tasks. The capability of the proposed scheduling algorithm is evaluated by a series of experiments based on a real phased-array radar for air defense frigates [18].

For future research, we shall further extend the results to RCC scheduling with multiple TRs, especially when they are applied for a more structured way in defense or even offense. We shall also consider a distributed system architecture for RCC scheduling for performance scaling. More research on probabilistic performance guarantees and related real-time scheduling technology might provide more efficient and economic solutions for advanced radar system designs.

ACKNOWLEDGMENTS

This paper is an extended version of a paper that appeared in Proceedings of the International Conference on Parallel Processing (ICPP), August 2004. This research was supported in part by the ROC National Science Council under grants NSC-94-2213-E-002-007, NSC-94-2752-E-002-008-PAE, NSC-94-2213-E-002-090, and NSC-94-2627-E-002-001.

REFERENCES


Chin-Fu Kuo received the BS and MS degrees in the Department of Computer Science and Information Engineering, National Chung Cheng University, in Chiayi, Taiwan, ROC, in 1998 and 2000, respectively. He received the PhD degree in computer science and information engineering from National Taiwan University, in Taipei, Taiwan, ROC, in 2005. His research interests include real-time process scheduling, resource management, and system security.

Ya-Shu Chen received the BS degree from National Chiao-Tung University, Taiwan, ROC, in 2001. She received the MS degree from National Taiwan University, Taiwan, ROC, in 2003. She has been pursuing the PhD degree in the Department of Computer Science and Information Engineering (CSIE) at National Taiwan University, ROC, since September 2003. Her current research interests include real-time systems, embedded systems, and system-on-a-chip.

Tei-Wei Kuo received the BSE degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, ROC, in 1986. He received the MS and PhD degrees in computer sciences from the University of Texas at Austin in 1990 and 1994, respectively. He is currently a professor in the Department of Computer Science and Information Engineering of the National Taiwan University, Taipei, Taiwan, ROC. He was an associate professor in the Department of Computer Science and Information Engineering of the National Chung Cheng University, Taiwan, ROC, from August 1994 to July 2000. His research interests include real-time process scheduling, real-time operating systems, embedded systems, and real-time databases. He was the program cochair of the IEEE Seventh Real-Time Technology and Applications Symposium in 2001 and an associate editor of the Journal of Real-Time Systems (SCI) since 1998. He has consulted for government and industry on problems in various real-time and embedded systems designs. Dr. Kuo is a senior member of the IEEE and has received several awards for his research achievements, including the Distinguished Research Award from the ROC National Science Council in 2003.

Phone Lin received the BSCSIE degree and PhD degree from National Chiao Tung University, Taiwan, ROC in 1996 and 2001, respectively. From August 2001 to July 2004, he was an assistant professor in the Department of Computer Science and Information Engineering at National Taiwan University, ROC. Since August 2004, he has been an associate professor. His current research interests include personal communications services, wireless Internet, and performance modeling. He is a guest editor for the IEEE Wireless Communications special issue on mobility and resource management, and a guest editor for the ACM/Springer special issue on wireless broad access. He is also an associate editorial member for the WCMI Journal. He is a senior member of the IEEE.

Cheng Chang received the BS degree in applied mathematics from Chung Cheng Institute of Technology in TaoYuan, Taiwan in 1982. He received the MS degree in computer and decision science from National Tsing Hwa University in HsingChu, Taiwan in 1987. He received the PhD degree in computer science from University of Illinois at Urbana-Champaign in 1996. He is currently an associated scientist in the System Development Center of the Chung Shan Institute of Science and Technology, TaoYuan, Taiwan, ROC and an assistant professor in the Department of Business Administration of the National Central University, TaoYuan, Taiwan, ROC. His research interests include real-time systems, software engineering, and project management.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.