

Probabilistic Real-Time Guarantees for Component-Oriented Phased Array Radars *

Chin-Fu Kuo, Ya-Shu Chen, Tei-Wei Kuo, Phone Lin and Cheng Chang[†]

Department of Computer Science and Information Engineering,

National Taiwan University, Taipei, Taiwan 106, ROC

[†]System Development Center, Chung Shan Institute of Science and Technology,

TaoYuan, Taiwan 325, ROC

{d89005,ktw}@csie.ntu.edu.tw

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 also re-implemented by software modules. In such systems, radar tasks could be modelled 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 off-line probabilistic analysis and on-line 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 [6].

Keywords: Phased Array Radar, Real-Time Task Scheduling, Probabilistic Performance Guarantee, Distributed Systems, Dwell Scheduling

1 Introduction

How to efficiently schedule radar resource to achieve the maximum performance is one of the most essential and challenging problems in the design of a multi-function phased array radar. Due to hardware constraints, many traditional radar systems are designed with non-real-time resource scheduling mechanisms, such as FIFO scheduling [3]. As a result, much resource is 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 every-

thing 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 re-implemented by software modules [4]. The development of component-oriented signal processors is strongly influenced by the Rapid prototyping of Application Specific Signal Processors (RASSP) program lead by the Department of Defense, United States of America [11]. 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 the component-oriented phased array radar architecture, radar tasks could be modelled 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 SP. Although real-time scheduling problems have been analyzed for different architectural assumptions, 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 [1, 6] are among the few closely related to dwell scheduling at the Radar Control Computer (RCC) level. Although researchers and engineers have started exploring real-time dwell scheduling at RCC, the proposed algorithms are mainly variations of the Partial Template algorithm [3]. 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 and Receiver (TR) and SP workloads with an analytical framework for off-line probabilistic analysis and on-line admission control.

While nonpreemptible task scheduling with end-to-end deadlines and precedence constraints is shown being NP-

*This research was supported in part by the National Science Council under grants NSC92-2213-E-002-091 and NSC-92-2213-E-002-093, and NSC92-2213-E-002-094.

hard in the literature [5], different heuristics on multi-stage scheduling (e.g., [13]) are proposed. Distinct from the past work, this paper aims at the proposing of a real-time scheduling algorithm at RCC to have joint considerations of TR and SP workloads. In addition to that, we would propose an analytical framework for off-line probabilistic analysis and on-line admission control to balance the hardware cost and the performance guarantee. 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 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 proposed based on the well-known rate-based multi-processor algorithm [2], provided that 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 off-line probabilistic analysis and on-line 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 [6].

The rest of this paper is as follows: Section 2 defines the system architecture 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, the performance evaluation results based on a real example system are presented. Section 5 is the conclusion.

2 System Architecture and Workload Characteristics

2.1 System Architecture

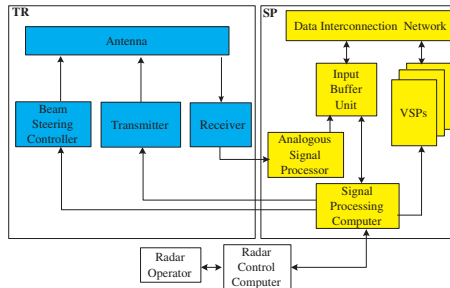


Figure 1. The hardware architecture of a phased array radar

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 Figure 1 [3, 6]. RCC schedules dwell transmissions in a real-time fashion by sending SPC commands. When SPC receives

commands from RCC for radar beam transmissions, it issues commands to BSC and Transmitter for radar beam transmissions in specified directions. Antenna and Receiver receive returned signals and pass them to ASP for analog-to-digital signal processing. The digital signals are saved at the input buffer unit (IBU) for later processing. Processing units of the SP are Vector Signal Processors (VSP) which conducts different types of data processing such as pulse compression, FFT, and other digital signal processing. Data Interconnection Network (DIN) is for data transmission between VSP's and IBU.

A *dwell* is defined as the process from the transmitting of a radar beam by Transmitter to the saving of the corresponding returned digital signals in IBU. RCC schedules tasks in units called *scheduling interval* (SI) [3]. In other words, RCC sends a sequence of commands to SP for dwell transmissions, and retrieves results from SP at the beginning of an SI. Besides, SP must read the output results of TR (in IBU) at the beginning of an SI. With the synchronization behavior among RCC, TR, and SP, radar tasks always consider to arrive at a multiple of SI and have deadlines as multiples of SI¹.

2.2 Workload Characteristics

A typical phased array radar could be modelled as a distributed system, where a radar task is decomposed into one TR subtask and one SP subtask. TR and SP subtasks must be executed in TR and 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 VSP's in SP. We say that the SP subtask corresponding to the dwell runs on one VSP and handles the returned signals. SP subtasks are nonpreemptible and could not be migrated among VSP's. 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 resulted from the execution of the TR subtask. An SP subtask could not start to execute until its precedent TR subtask finishes. A task τ_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 TR and SP subtasks of τ_i have their worst execution times, depending on the characteristics of their corresponding dwell or radar task.

A phased array radar could have at least three kinds of works: Search, Track Confirmation, and Track.

- *Search*: A phased array radar must scan its surveillance space periodically for suspicious targets, e.g., in terms of

¹Some implementations of radar systems ignore the SI constraints. However, the synchronization among RCC, TR, and SP will become complicated because synchronization might be needed virtually at any time.

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 the servicing of highly critical tasks.

- **Track Confirmation:** When suspicious targets are detected at RCC (because of reflected signals), a Track Confirmation is issued for each detected target in the direction of the target to verify its presence. A Track Confirmation (TC) task must be done in a hard real-time fashion to *identify* suspicious targets.
- **Track:** Once a target is identified, a sequence of “semi-periodic” Normal Track (NT) dwells are issued to track the target. The word *semi-periodic* means “periodic” but with dynamically changing periods. The reason for tracking being semi-periodic is because 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 tracking of a target might need to go into a Precision Track (PT) task, which is also “semi-periodic”, because the tracking of the target needs better precision. 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.

Task Types	Timing Constraints	Periodic	Deadline Type	Priority
High-Priority Search (HS)	$\{B_i^S, c_{S,1}, c_{S,2}, D_i^S, P_i^S\}$	periodic	hard deadline	1
Track Confirmation (TC)	$\{c_{C,1}, c_{C,2}, D_C\}$	aperiodic	hard deadline	2
High-Precision Track (HPT) Precision Track (PT) Normal Track (NT)	$\{c_{T,1}, c_{T,2}, P_T^H, P_T^L, D_T\}$	semi-periodic	hard deadline	3

Table 1. Timing parameters of a typical phased array radar

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 consists of High-Priority Search tasks, High Precision Track tasks, Precision Track tasks, Normal Track tasks, and Track Confirmation tasks, as shown in Table 1, where 1 is the highest priority, and 3 is the lowest priority [6].

An HS task must issue B_i^S beams every P_i^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 the HS task is D_i^S . HS or LS tasks are referred to as *search tasks* although we will focus our discussions on HS tasks in this paper (where the timing constraints of HS and LS tasks are similar). For each suspicious target, a TC 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 semi-periodic Track dwells are issued to track the target. For the rest of this

paper, NT, PT, or HPT tasks are referred to as *track tasks* for the simplicity of discussions for the rest of this paper. Each track task issues a sequence of semi-periodic Track dwells to track a target. The period of a track task is bounded by a lower bound P_T^L and an upper bound P_T^H . Besides, a track task has a dwell length $c_{T,1}$ and requires the execution time $c_{T,2}$ of the returned signals in SP. The end-to-end relative deadline of track task is D_T .

3 Probabilistic Schedulability Guarantees under Two-Stage Scheduling

3.1 Overview

The purpose of this section is to propose a two-stage scheduling algorithm to provide different levels of guarantees for different radar tasks in a probabilistic fashion. It is to avoid unnecessary deployment of a large number of hardware equipments, due to some pessimistic performance analysis of radar systems.

The component-oriented phase-array radars under considerations have two major components, as shown in Figure 1: TR and SP. Each radar task τ_i is modelled 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*) represents the processing of the returned signals in SP. The system architecture of a phased-array radar could be abstracted as a chain of two processor groups. TR must adopt nonpreemptible uniprocessor scheduling because dwell transmissions must be done one after another. We propose to schedule TR subtasks in a priority queueing model (Please see Section 3.2) on the first processor group (referred to as 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 SP for processing. We propose to schedule SP subtasks with returned signals in the IBU in a distributed rate-based real-time scheduling algorithm (Please see Section 3.3) on the second processor group (referred to as the *SP p-group*). Because signal processing on each VSP is also non-preemptible, no migration of any task execution among processors (i.e., VSP's) in the SP p-group is allowed. The number of VSP's in SP varies, depending on the needs of a radar system for signal processing.

Let each task τ_i be given a probability threshold ϕ_i such that the phased-array radar must guarantee that a specified portion of its tasks will always 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 τ_i , and D_i is a given deadline (relative to its arrival time). In this paper, we first assume that TR subtasks (and its 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 τ_i in the system be generally distributed with

a density function f_i and a rate λ_i . Each task τ_i has a deadline D_i relative to its arrival time, referred to as the *relative deadline*. The completion time of a task τ_i relative to its arrival time is referred to as the *relative completion time*. The technical problem on the scheduling of TR and SP subtasks is how to assign TR and SP subtasks deadlines for real-time scheduling.

We first 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 on the analysis of the completion time of TR subtasks, and the second part is on the schedulability analysis of SP subtasks.

3.2 Probabilistic Guarantees for Beam-Transmission Scheduling

Given a probabilistic guarantee $\Pr[R_i \leq D_i] \geq \phi_i$ for a task τ_i , the first thing is to derive a bound $D_{i,1}$ on the completion time of its TR subtask based on the priority queueing 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 a table lookup over the standard normal distribution table based on the derived mean and variance.

Different radar systems have different priority assignments for different radar tasks [6]. We assume that the priority of a TR subtask or an SP subtask inherits that of its corresponding radar task. When a TR subtask is under processing for a beam transmission, it is non-preemptable. We adopt a priority queueing model for TR-subtask scheduling, where the TR p-group services TR subtasks with higher priorities first. We assume that 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.

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

We model each radar task τ_i in the system with a Poisson arrival pattern with the rate λ_i . The service time of the TR subtask of τ_i has a general service time distribution with the mean $c_{i,1}$. Each task τ_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 τ_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 [8]:

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 $\Lambda = \lambda_1 + \lambda_2 + \dots + \lambda_N$. 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: $E[C] = \frac{\lambda_1}{\Lambda} c_{1,1} + \frac{\lambda_2}{\Lambda} c_{2,1} + \dots + \frac{\lambda_N}{\Lambda} c_{N,1}$ and $E[C^2] = \frac{\lambda_1}{\Lambda} c_{1,1}^2 + \frac{\lambda_2}{\Lambda} c_{2,1}^2 + \dots + \frac{\lambda_N}{\Lambda} c_{N,1}^2$. Let W_{q_i} be the waiting time for a TR subtask of τ_i in the queue. We can obtain the expected value of W_{q_i} as follows: $E[W_{q_i}] = \frac{\lambda E[C^2]}{2(1-u_{i-1})(1-u_i)}$ where $u_i = \lambda_1 \cdot c_{1,1} + \lambda_2 \cdot c_{2,1} + \dots + \lambda_i \cdot c_{i,1}$. Consequently, the expected value of the time W_i spent by a TR subtask of τ_i in the system is $E[W_i] = E[W_{q_i}] + \frac{1}{c_{i,1}}$, and the expected value of the time W spent by an arbitrary TR subtask is $E[W] = E[W_q] + \frac{1}{E[C]}$.

The purpose of this section is to derive the deadline $D_{i,1}$ for TR subtasks of each radar task τ_i based on the distribution of the waiting time of τ_i . Let $f_{q,i}(x)$ be the density function of the waiting time W_{q_i} for each TR subtask of radar task τ_i , and $f_{q,i}^*(s)$ the Laplace transform of $f_{q,i}(x)$. Based on the results in [8], $f_{q,i}^*(s)$ could be obtained as follows:

$$f_{q,i}^*(s) = \frac{(1-u_n)[s + \lambda_H - \lambda_H G_H^*(s)] + \lambda_L[1 - B_L^*(s + \lambda_H - \lambda_H G_H^*(s))]}{s - \lambda_j + \lambda_j B_j^*(s + \lambda_H - \lambda_H G_H^*(s))} \quad (1)$$

where $B_i^*(s)$ is the Laplace transform of the density function for the service time of each TR subtask of τ_i , $\lambda_H = \sum_{i=1}^{j-1} \lambda_i$, $\lambda_L = \sum_{i=j+1}^n \lambda_i$, $B_H^*(s) = \sum_{i=1}^{j-1} \frac{\lambda_i}{\lambda_H} B_i^*(s)$, $B_L^*(s) = \sum_{i=j+1}^n \frac{\lambda_i}{\lambda_L} B_i^*(s)$, $G_H^*(s) = B_H^*(s + \lambda_H - \lambda_H G_H^*(s))$, and $u_i = \lambda_1 \cdot c_{1,1} + \lambda_2 \cdot c_{2,1} + \dots + \lambda_i \cdot c_{i,1}$. Based on results in [14], we have $E[W_{q_i}^n] = (-1)^n \frac{d^n f_{q,i}^*(s)}{ds^n} \Big|_{s=0}$ and the variance v_{q_i} of W_{q_i} can be obtained by the following equation: $v_{q_i} = E[W_{q_i}^2] - (E[W_{q_i}])^2$. We use the normal distribution $N(\cdot)$ with the mean $E[W_{q_i}]$ and the variance v_{q_i} to estimate $D_{i,1}$ such that $P_{TR}(D_{i,1}) \geq \phi_i$ by a table lookup, where $P_{TR}(x) = \Pr[R_i \leq x]$.

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 [10] 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 τ_i is given a positive real number θ_i (called the *reservation ratio*) such that τ_i is guaranteed to be served at a rate of $g_i = \frac{\theta_i}{\sum_j \theta_j}$. In particular, Spuri, et al. [12] proposed an effective GPS-based mechanism called *TB server* to service tasks under the framework of the EDF scheduling. In [2], a distributed and revised version of the TB

algorithm (called Multiprocessor Constant-Bandwidth Server (M-CBS)) was proposed for multi-processor 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., $D_{i,2} = (D_i - D_{i,1})$, for a given probability to guarantee the schedulability of the corresponding radar task τ_i .

The design of an SP scheduling algorithm must guarantee the processing of all of the returned signals of the TR subtasks of τ_i if they could arrive at IBU within their relative deadline $D_{i,1}$. As discussed in Section 2, the period of each track task τ_i has a lower bound P_i^L and an upper bound P_i^H , depending on its corresponding track type. Once a radar task is admitted, a radar system should keep tracks of the corresponding target with a specified degree of guarantee. The reservation ratio of each radar task is set as $\theta_i = \frac{c_{i,2}}{D_{i,2}}$, where $c_{i,2}$ is the execution time of the SP subtask of τ_i , if $P_i^L \geq D_{i,2}$. Otherwise, the reservation ratio of τ_i is $\theta_i = \frac{c_{i,2}}{P_i^L}$. The rationale behind the assignment is to provide the SP subtasks of τ_i an 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 τ_i is $\theta_i = \frac{c_{i,2}}{D_{i,2}}$ if $P_i^S \geq D_{i,2}$, where P_i^S is the period of the search task. Otherwise, the reservation ratio of τ_i is $\theta_i = \frac{c_{i,2}}{P_i^S}$. Note that since the tracking of an identified target starts with a Track Confirmation, then a sequence of Normal Tracks, and 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 [2], that is a work-conserving scheduling mechanism in which the schedulability of the SP subtasks of each radar task τ_i is guaranteed with a reservation ratio θ_i . We say that an SP subtask arrives if its corresponding returned signals is in the IBU of SP. When an SP subtask arrives, its deadline is set based on M-CBS, except that no preemption or migration is allowed: Let $T = \tau_1, \tau_2, \dots, \tau_n$ denote a collection of radar tasks under scheduling, where each radar task τ_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 non-increasing reservation ratio order, i.e., $\theta_i \geq \theta_{i+1}$, for all $i, 1 \leq i \leq n$, and $\theta(T) = \sum_{\tau_i \in T} \theta_i$. $T^{(i)}$ denotes the collection of radar tasks with the $(n-i+1)$ minimum reservation ratios in T , i.e., $T^{(i)} = \tau_i, \tau_{i+1}, \dots, \tau_n$. (According to this notation, $T \equiv T^{(1)}$.)

Let $\kappa(T)$ denote the smallest value of k that satisfies the following inequality (Please see Theorem 2 in the next section), where M is the number of processors, i.e., VSP's, in the SP p-group. $Max(npb_i)$ and $Min(d_j)$ are the maximum execution time and the minimum relative deadline of all SP subtasks, respectively:

$$M(1 - \frac{Max(npb_i)}{Min(d_j)}) \geq \min_{k=1}^n \{(k-1) + \frac{\theta(T^{(k+1)})}{1 - \theta_k}\} \quad (2)$$

Radar tasks $\tau_1, \dots, \tau_{\kappa(T)-1}$ are called *high-priority radar tasks*, and the rest are called *deadline-based radar tasks*. Let an SP subtask of an admitted radar task τ_i arrive at time t , the deadline of the SP subtask is set as $-\infty$ if τ_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 τ_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 $d_{i,l} = \max\{t, d_{i,l-1}\} + \frac{c_{i,2}}{\theta_i}$, where $d_{i,l}$ denotes the absolute deadline of the l_{th} instance of the SP subtasks of τ_i , and $d_{i,0} = 0$. The scheduling of SP subtasks is based on the earliest-deadline-first algorithm [9], where the SP subtask with the earliest absolute deadline is scheduled first for any available processor. No preemption or migration is allowed. The SP 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 τ_i that could have a reservation ratio θ_i larger than one is split into $\lceil \frac{1}{P_i^S/B_i^S} \rceil \cdot \theta_i$ corresponding radar tasks, i.e., $\tau_{i,1}, \dots, \tau_{i, \lceil \frac{1}{P_i^S/B_i^S} \rceil \cdot \theta_i}$.² The reservation ratio of each corresponding radar task is $\frac{\theta_i}{\lceil \frac{1}{P_i^S/B_i^S} \rceil \cdot \theta_i}$.

Instances of TR subtasks of τ_i and their corresponding SP subtasks are assigned to the $\lceil \frac{1}{P_i^S/B_i^S} \rceil \cdot \theta_i$ corresponding radar tasks in a round robin fashion. Note that a search task τ_i must issue B_i^S beams in each period P_i^S . Each radar task $\tau_{i,j}$ (split from τ_i) is considered independently in TR and SP scheduling. The SP scheduling of all radar tasks (regardless of whether they are split or not) is as presented in the previous paragraph.

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 shall first summarize related theorems of M-CBS in [2, 9]: Note that each M-CBS server in [2, 9] is a radar task in this paper.

Theorem 1 [2] *Given a collection T of M-CBS servers, let each server be associated with a reservation ratio θ_i . T is*

²Search tasks are radar tasks that could have a reservation ratio $\theta_i = \frac{c_{i,2}}{D_{i,2}}$ (or $\frac{c_{i,2}}{P_i^S/B_i^S}$ if $(P_i^S/B_i^S) < D_{i,2}$) larger than one.

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, where M is the number of processors: $M \geq \min_{k=1}^n \{(k-1) + \frac{\theta(T^{(k+1)})}{1-\theta_k}\}$.

Lemma 1 [9] When a uni-processor system is scheduled with more than one M-CBS servers 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 one, we shall first provide a schedulability test for radar tasks if no radar task has a reservation ratio larger than one. We then show that the inequality of the schedulability test remains when we split radar tasks with a reservation ratio larger than one in a way 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:

$$M(1 - \frac{Max(npb_i)}{Min(d_j)}) \geq \min_{k=1}^n \{(k-1) + \frac{\theta(T^{(k+1)})}{1-\theta_k}\},$$

where $Max(npb_i)$ and $Min(d_j)$ are the maximum execution time and the minimum of the relative deadlines of all SP subtasks, respectively.

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

According to M-CBS-NPM, any search task τ_i that has a reservation ratio θ_i larger than one must be first split into $\lceil \frac{1}{P_i^S/B_i^S} \rceil \cdot \theta_i$ corresponding radar 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 τ_i with a reservation ratio $\theta_i > 1$, τ_i could be split into $\lceil \frac{1}{P_i^S/B_i^S} \rceil \cdot \theta_i$ corresponding radar tasks, i.e., $\tau_{i,1}, \dots, \tau_{i,\lceil \frac{1}{P_i^S/B_i^S} \rceil \cdot \theta_i}$, and the reservation ratio of each corresponding radar task is $\frac{\theta_i}{\lceil \frac{1}{P_i^S/B_i^S} \rceil \cdot \theta_i}$. If instances of TR subtasks of τ_i and their corresponding SP subtasks are assigned to the $\lceil \frac{1}{P_i^S/B_i^S} \rceil \cdot \theta_i$ corresponding radar tasks in a round robin fashion, then the SP subtasks of the instances of τ_i will meet their deadlines under M-CBS-NPM.

Proof. The correctness of the proof follows from the fact that the total execution time guaranteed by M-CBS-NPM for the corresponding radar tasks is no less than the maximum demanded execution time. \square

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: $M(1 - \frac{Max(npb_i)}{Min(d_j)}) \geq \min_{k=1}^n \{(k-1) + \frac{\theta(T^{(k+1)})}{1-\theta_k}\}$, where $Max(npb_i)$ and $Min(d_j)$ are the maximum execution time and the minimum of the relative deadlines of all SP subtasks, respectively.

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

Theorem 3 could be used for on-line admission control when the system considers to admit any new radar task, given the number of VSP's in SP. We must point out that Theorem 3 could also be used for the capacity estimation of SP when the (maximum) workload of a radar system and its characteristics are given! Engineers could estimate the minimum number of VSP's needed for a given workload in a more optimistic based on the following corollary:

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

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

4 Performance Evaluation

The purpose of this section is to evaluate the performance of the proposed methodology, referred to as the Probabilistic Real-Time Scheduling algorithm (PRTS) in this section. A simulation model was constructed to investigate the performance of the proposed methodology. We compared the schedulability and the performance of PRTS with the Ultimate Deadline Algorithm (UD) [7, 9], the Proportional Deadline Algorithm (PD) [9], the Equal Deadline Algorithm (EQD), the Equal Flexibility Algorithm (EQF) [7], the Equal Slack Algorithm (EQS) [7], and the Effective Deadline Algorithm (ED) [7, 9]³. The performance of PRTS proposed in this paper was verified and simulated under different probabilistic guarantees (i.e., 91%, 93%, 95%, 97%, and 99%). We used PRTS _{ϕ} to denote PRTS with a probabilistic guarantee ϕ in this section.

The system parameters for the experiments were based on a real phased array radar for air defense frigates [6]: Each tested task set had a search task and several track tasks. The search task issued 45 beams for every 40 SI (where 40 SI = 1 sec). We set the dwell length, the execution time in SP, and the relative deadline of each search (/track) task as

³Under UD, $D_{i,1} = D_i$ and $D_{i,2} = 0$. Under PD, $D_{i,1} = D_i \cdot \frac{c_{i,1}}{c_{i,1}+c_{i,2}}$ and $D_{i,2} = D_i \cdot \frac{c_{i,2}}{c_{i,1}+c_{i,2}}$. Under EQD, $D_{i,1} = D_i/2$ and $D_{i,2} = D_i/2$. Under EQF, $D_{i,1} = (D_i - c_{i,1} - c_{i,2}) \cdot \frac{c_{i,1}}{c_{i,1}+c_{i,2}} + c_{i,1}$ and $D_{i,2} = (D_i - c_{i,1} - c_{i,2}) \cdot \frac{c_{i,2}}{c_{i,1}+c_{i,2}} + c_{i,2}$. Under EQS, $D_{i,1} = \frac{D_i - c_{i,1} - c_{i,2}}{2} + c_{i,1}$ and $D_{i,2} = \frac{D_i - c_{i,1} - c_{i,2}}{2} + c_{i,2}$. Under ED, $D_{i,1} = D_i - c_{i,2}$ and $D_{i,2} = c_{i,2}$.

0.24 (/0.16) SI, 1.5 (/0.25) SI, and 8 (/6) SI, respectively. The time interval between two consecutive instance arrivals for a track task was exponentially distributed with a mean 4 SI. The number of the track tasks in a tested task set of the simulations was from 10 to 20. In other words, the total utilization of the corresponding TR subtasks (including the search task) varied from 0.67 to 1.07. When the total utilization of TR subtasks was over 1, TR was overloaded. Let N_T denote the number of track tasks per SI. Each task set was simulated for 40,000 SI.

The primary performance metrics were the schedulability of the task sets and the number of VSP's (labelled as N_{VSP}) needed for different scheduling algorithms. The schedulability of a simulated task set for each algorithm had to be achieved. If an algorithm under simulation could schedule a task set, then we re-did the simulation with a smaller number N_{VSP} of VSP's until some task in the task set missed its deadline. We should derive the minimum number of VSP's needed for a scheduling algorithm.⁴ We did not show the minimum number of VSP's needed for a scheduling algorithm in figures if the scheduling algorithm could not schedule the task set, regardless of the number of adopted VSP's in SP.

4.1 Analysis Results and Experimental Results

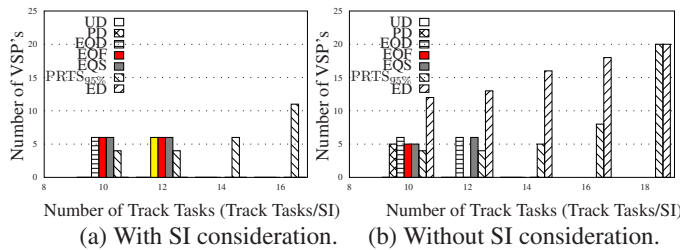


Figure 2. The number of VSP's needed for the algorithms with and without SI considerations.

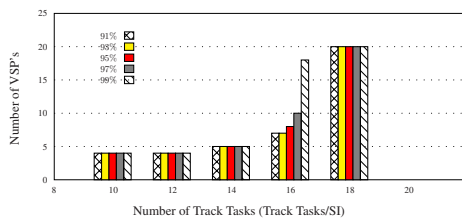


Figure 3. The number of VSP's needed for PRTS without SI considerations.

In this section, we showed the analysis results based on Theorem 3 and the results of the simulation. We first investigated the N_{VSP} values for different algorithms based on

⁴Note that a smaller number of VSP's means a better system performance.

the calculation results by using Theorem 3. The algorithms under evaluation included UD, PD, EQD, EQF, EQS, ED, and PRTS_{95%}. Let $U_{TR,i}(\mathbb{K})$ denote the total average utilization for the task set TS_i (with i track tasks) for Algorithm \mathbb{K} , where \mathbb{K} can be UD, PD, EQD, EQF, EQS, ED, or PRTS. We first calculated $U_{TR,i}(\mathbb{K})$ for the algorithms. TS_i could not be scheduled by Algorithm \mathbb{K} , if the calculated value $U_{TR,i}(\mathbb{K}) > (1 - \frac{Max(npbi)}{Min(d_j)})$, where $Max(npbi)$ and $Min(d_j)$ were the maximum execution time and the minimum of the relative deadlines of all TR subtasks, respectively. Otherwise (i.e., $U_{TR,i}(\mathbb{K}) \leq (1 - \frac{Max(npbi)}{Min(d_j)})$), we could use Theorem 3 to calculate N_{VSP} for Algorithm \mathbb{K} , and this algorithm could schedule the tested task set TS_i .

The Performance Comparison for Different Algorithms

with SI Considerations. In Figure 2.(a), we compared N_{VSP} for different algorithms where SI was considered. Obviously, UD, PD, and ED could not schedule any tested task set. The reason was that $D_{i,1}$ ($D_{i,2}$) of a task τ_i under ED (UD or PD) was too small to guarantee the scheduling of the TR (SP) subtasks. When $N_T \leq 12$, PRTS_{95%}, EQD, EQS, and EQF could schedule the tested task sets, and the N_{VSP} values for PRTS_{95%} were smaller than that for EQD, EQS, and EQF. The reason was that $D_{i,2}$ for PRTS_{95%} was larger than that for EQD, EQS, or EQF.

The Performance Comparison for Different Algorithms

without SI Considerations. In Figure 2.(b), we compared N_{VSP} for different algorithms where SI was not considered. We observed that UD could not schedule any tested task set. EQD and EQS could schedule the tested task sets when $N_T \leq 12$. EQF could schedule the tested task sets only when $N_T = 10$. Although ED and PRTS_{95%} could schedule the tested task sets when $N_T \leq 18$, PRTS_{95%} had smaller N_{VSP} values. We knew that PRTS_{95%} outperformed the other algorithms in terms of N_{VSP} and had the highest guarantee for schedulability.

The Effects of Probabilistic Guarantees on PRTS without SI Considerations.

For PRTS, we might set up different probabilistic guarantees when a task set was scheduled. In Figure 3, we investigated the effects of the probabilistic guarantees on PRTS when SI was not considered. We set probabilistic guarantees as 91%, 93%, 95%, 97%, and 99% and re-run the experiments. For the situation when SI was considered, we observed similar phenomena, so the results for PRTS with SI considerations were not shown in this paper. We observed the following phenomena: (1) When $N_T \leq 14$, the effects of the probabilistic guarantees were insignificant (i.e., the N_{VSP} values were almost the same). (2) When $N_T = 16$, PRTS with the higher probabilistic guarantee needed more VSP's. (3) When $N_T = 18$, the N_{VSP} values for PRTS with different probabilistic guarantee setups were the same, which were close to

20. Phenomenon 1 was observed because TR utilization was relatively low so that the waiting time of a TR subtask was not very long even under different probabilistic guarantees. Phenomenon 1 indicated that when $N_T \leq 14$, we could set the probabilistic guarantee as 99% (i.e., no more than 1% task instances missed their deadlines), PRTS could provide the same schedulability as that the other algorithms provided, and the N_{VSP} value for PRTS was the smallest among UD, PD, EQD, EQF, EQS, ED, and PRTS (Please see Figure 2.(b).). Phenomenon 2 showed that PRTS needed more VPS's to provide a higher probabilistic guarantee with the increasing of N_T (i.e., $N_T = 16$). Note that when $N_T = 18$, the system utilization of TR was about 99%. Since TR was almost fully utilized, different probabilistic guarantees did not affect the N_{VSP} values, as shown in Phenomenon 3. When $N_T = 20$ (i.e., the TR utilization = 1.06), TR might be overloaded, no scheduling algorithms could guarantee the schedulability of all tasks without any deadline violation.

In the experiments, PRTS always guaranteed the schedulability of search tasks and, at the same time, reduced the number of VSP's needed for phased array radars, compared to UD, PD, EQD, EQF, EQS, and ED. Even when the number of track tasks per SI was 20, 99.96% of the executions of track tasks would meet their deadline requirements (even though a 100% or even 95% guarantee was impossible), where PRTS without SI consideration was applied. The experimental results for PRTS with SI consideration were similar to those without SI consideration.

5 Conclusion

This paper addresses two important issues on radar scheduling: (1) schedulability guarantees for radar tasks (2) the system capacity estimation. While lots of existing work suffers from conservative resource allocation problems, we aim at the proposing of a joint real-time scheduling algorithm for TR and SP workloads with an analytical framework for off-line probabilistic analysis and on-line 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 multi-processor 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 [6].

For future research, we shall further extend the results to RCC scheduling with multiple TR's, especially when they are applied in a more structured way. We shall also apply

the proposed two-stage scheduling algorithm to more general distributed real-time systems.

References

- [1] A. Barbato and P. Giustiniani. An improved scheduling algorithm for a naval phased array radar. In *Radar 92. International Conference*, pages 42–45, 1992.
- [2] S. Baruah, J. Goossens, and G. Lipari. Implementing constant-bandwidth servers upon multiprocessor platforms. In *Eighth IEEE Real-Time and Embedded Technology and Applications Symposium*, 2002.
- [3] R.A. Baugh. *Computer Control of Modern Radars*. RCAM&SR-Moorestown Library, 1973.
- [4] C. Chang and T.-C. Wang. Use object-oriented paradigm to design a programmable radar digital signal processor. In *Third Workshop on Object-Oriented Technology and Applications*, 1997.
- [5] M.R. Garey and D.S. Johnson. *Computers and Intractability: A guide to the Theory of NP-Completeness*. W. H. Freeman & Company, 1979.
- [6] A.G. Huizing and A.A.F. Bloemen. An efficient scheduling algorithm for a multifunction radar. In *IEEE International Radar Conference*, pages 359–364, 1996.
- [7] B. Kao and H. Garcia-Molina. Deadline assignment in a distributed soft real-time system. *IEEE Transactions on Parallel and Distributed Systems*, 8(12):1268–1274, 1997.
- [8] L. Kleinrock. *Queueing Systems Volume II: Computer Applications*. Wiley-Interscience Publication, 1976.
- [9] J. W. Layland. *Real-Time System*. Prentice Hall, 2000.
- [10] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single node case. In *The Proceedings of IEEE INFOCOM*, 1992.
- [11] Rapid prototyping of Application Specific Signal Processors (RASSP). <http://eto.sysplan.com/eto/rassp>. 2000.
- [12] M. Spuri, G. Buttazzo, and F. Sensini. Scheduling aperiodic tasks in dynamic scheduling environment. In *IEEE Real-Time Systems Symposium*, 1995.
- [13] L.H. Su. A hybrid two-stage flowshop with limited waiting time constraints. *Computers and Industrial Engineering*, 44(3):409–424, 2003.
- [14] E.J. Watson. *Laplace Transforms and Applications*. Birkhauser, 1981.