

Real-Time Dwell Scheduling of Component-Oriented Phased Array Radars

Tei-Wei Kuo, *Senior Member, IEEE*, Yung-Sheng Chao, Chin-Fu Kuo, and Cheng Chang

Abstract—A multifunction phased array radar must search and track suspicious targets in its surveillance space in a real-time fashion. With inefficient scheduling implementations in many traditional systems, much radar resource is wasted with a very limited performance gain. This paper targets one of the most important issues in the design of modern phased array radars: real-time dwell scheduling. We formalize the typical workload of a modern phased array radar and propose a rate-based approach to schedule radar dwells in a real-time fashion. We show how to reserve radar resources to guarantee the minimum radar operation without sacrificing the stability of the system. The strength of our approach is verified by a series of simulation experiments based on a real phased array radar for air defense frigates [9]. A significant improvement in the performance of phased array radars was shown.

Index Terms—Phased array radar, real-time dwell scheduling, rate-based scheduling, radar control computer.

1 INTRODUCTION

A multifunction phased array radar must search for and track suspicious targets in its surveillance space in a real-time fashion. There are two major modules in a phased array radar: Radar Control Computer (RCC) and Signal Processor (SP). RCC is responsible for scheduling radar beam transmissions for searching and tracking based on the targets' status and search types. SP must process returned signals in a real-time fashion. With the advance of software and hardware technology, a modern phased array radar is no longer a complicated hardware system with everything wired. Instead, engineers now are building phased array radars with commercial-off-the-shelf (COTS) components and the functions of many hardware components are now reimplemented by software modules [5]. We must point out that the real-time scheduling problem explored in this paper for component-oriented radar systems is closely related to many scheduling problems for multiserver systems in which component-oriented designs, such as those adopted by CORBA and DCOM, are adopted. Similar to the timely processing of radar tasks over multiple processing units in SP, client requests in many multiserver systems must be handled in a timely fashion. While the component-oriented design concept becomes popular in many industrial sectors, how to schedule requests with good timing requirements is of paramount importance. Although technology is developed in this paper to schedule radar workloads in a timely fashion, the results derived in

this paper could be applied to scheduling problems for many component-oriented multiserver systems.

The development of component-oriented phased array radars is strongly influenced by the Rapid prototyping of Application Specific Signal Processors (RASSP) program lead by the US Department of Defense [25]. The RASSP program formalized an engineering process for developing SPs to reduce the total product development time and cost by a factor of four. Such a development opens up a new era in radar design and flexible resource scheduling. While a number of researchers have proposed highly efficient real-time scheduling algorithms, e.g., [14], [18], [21], [22], [28], little work is done for radar scheduling. The task models and the work presented in [4], [8], [9], [14], [18] are among the few closely related to real-time dwell scheduling at the RCC level, where a dwell consists of RCC-SP command submission, round-trip radar beam traveling, etc. (which will be explained later) [2]. Note that, although researchers have explored the scheduling of SDF graphs for the SAR benchmark and other related signal processing applications in the embedded systems and signal processing literature, they are more related to the scheduling of flow-based signal processing workloads. Most importantly, many existing (or even advanced) phased array radar systems still adopt inefficient or even non-real-time resource scheduling mechanisms, such as those based on FIFO-like or cyclic-executive-like scheduling algorithms [2]. The common reason in doing so is mainly because of hardware constraints and insufficient knowledge of real-time technology. As a result, much resource is wasted with a very limited guarantee on system performance. This observation underlines the motivation of this research and our implementation work for the next-generation phased array radar systems in Taiwan. In this paper, we shall present our rate-based approach in the guarantee of the minimum system operation and demonstrate the capability of real-time scheduling technology, especially on rate-based scheduling, in improving the performance of a modern phased array radar.

- T.-W. Kuo and C.-F. Kuo are with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan 106, ROC. E-mail: {ktw, d89005}@csie.ntu.edu.tw.
- Y.-S. Chao is with the Electronic Research Division, Chung Shan Institute of Science and Technology, TaoYuan, Taiwan 325, ROC. E-mail: y.s.chao@seed.net.tw.
- C. Chang is with the System Development Center, Chung Shan Institute of Science and Technology, TaoYuan, Taiwan 325, ROC. E-mail: cchang60@ms37.hinet.net.

Manuscript received 1 Nov. 2002; revised 15 June 2004; accepted 22 July 2004; published online 16 Nov. 2004.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 117695.

Rate-based scheduling has been an active research topic in the past few years. Researchers proposed various rate-based scheduling algorithms for periodic and sporadic processes based on the notion of General Processor Sharing (GPS) [1], [14], [19], [26], [29], [31], [32]. The idea of GPS-based scheduling is very different from common disciplines such as priority-driven scheduling [3], [21], [22], [28] and time-driven scheduling [10], [11]. The GPS-based scheduling is a work-conserving scheduling mechanism. The schedulability of each process in GPS-based systems is guaranteed with an assigned CPU service rate, independently of the demands of other processes. The enforcement of a guaranteed CPU service rate for a process must rely on certain admission control mechanisms to manage the total workload of the system [26].

Parekh and Gallager [24] were the first to use the idea of GPS for rate-based flow and congestion control at network gateway nodes. Waldspurger and Weihl [31], [32] were the first to develop a process scheduling algorithm based on the idea of GPS. Their algorithm allocates a time quantum to the process which has the worst progress in the system according to its resource reservation. Stoica et al. [29] proposed and analyzed another GPS-based process scheduling algorithm, where CPU time is allocated in discretized time quanta such that a process, regardless of whether it is a real-time or non-real-time process, makes progress at a uniform rate. Spuri and Buttazzo [26] proposed an effective mechanism, called *TB server*, to service sporadic processes. The capacity and performance of a TB server is guaranteed by preserving the CPU bandwidth. Deng et al. [6], [7] proposed the idea of constant utilization servers (CUS) to reserve system resources for applications in an open system architecture, especially when applications may share global resources. Kuo and Li [17] later extended their work on the open system architecture for fixed-priority-driven OS schedulers. Kuo et al. [19] proposed a GPS-based scheduling framework for periodic and sporadic process scheduling, jitter control, service rate adjustment, and mixed scheduling of soft and hard real-time processes. Jeffay and Goddard [14] proposed sufficient and necessary feasibility conditions for rate-based process sets, where each process may request a number of executions within every specified amount of time. The algorithm assigns deadlines so that consecutive deadlines are separated by at least one period and then uses EDF to schedule tasks.

Past work in real-time dwell scheduling suffers from several serious problems: First, various inefficient scheduling mechanisms or data structures such as FIFO queues were adopted, making the deadlines of critical tasks difficult to guarantee due to lengthy priority inversion time. The situation is further worsened by the highly dynamic nature of radar workloads. In order to guarantee the minimum radar operation, engineers usually use a large number of radar resources (including computing power) which may not be needed under proper real-time scheduling algorithms to accomplish their performance goals.

This paper targets an essential issue for the design of modern phased array radars, i.e., real-time dwell scheduling. The concept of resource reservation is introduced for workload scheduling of component-oriented systems. As a

result, the two most important issues in real-time dwell scheduling are addressed: 1) the minimum operation guarantee of phased array radars and 2) the system capacity estimation. We first formalize the typical workload of radar tasks.¹ We then propose a rate-based approach to guarantee the minimum operation of a phased array radar and show that our approach can effectively schedule radar tasks with a good radar resource utilization. We then show how to utilize the semantics of radar task flows to better reserve radar resources in dwell scheduling and the system capacity is thus derived. It is shown that our approach can effectively distribute multiple instances of a search task in a period more evenly. Such a result is very important to radar performance. Our methodology is verified by a series of experiments based on a real phased array radar for air defense frigates [9] for which we have very encouraging results. We show that real-time technology does improve radar performance significantly. In particular, the proposed Batch-TB can greatly improve the performance of a phased array radar by tracking over 116 percent more targets than the traditional Partial Template scheduling algorithm (PM) does [2], [9] and, at the same time, improving the capacity of High-Precision Track by over 40 percent, compared to that under PM.

The rest of this paper is as follows: Section 2 illustrates the functions of modern phased array radars and formalizes their typical workload. Section 3 presents a rate-based approach in real-time dwell scheduling. A way of resource reservation is presented and nice properties are proven. In Section 4, a series of simulation experiments was done based on a real example system. Section 5 is the conclusion.

2 SYSTEM MODEL AND WORKLOAD CHARACTERISTICS OF PHASED ARRAY RADARS

2.1 Functions of a Phased Array Radar

A typical radar transmits a beam of microwaves toward a target and waits for the reflection. A target is said to be *detected* if the radar receives the microwaves reflected by the target. The direction of the target is the direction of the radar beam and the distance (called *range*) of that target can be calculated by how long it takes to receive the microwave reflection. One major difference between a traditional radar and a phased array radar is that a traditional radar has only one antenna, but a phased array radar consists of an array of antennae [2]. In a traditional radar, the radar beam is steered by the mechanical radar pedestal, while, in a phased array radar, the radar beam is steered by an electronic Beam Steering Controller (BSC). BSC steers the radar beam by adjusting the phase difference of each antenna during microwave transmission such that the microwave energy is concentrated toward the desired direction. As a result, the Radar Control Computer (RCC) of the phased array radar can steer the radar beam from one direction to another direction in the electronic speed, instead of some mechanical speed for traditional radars [2]. With an advanced multifunction signal processor (SP), a phased array radar

1. The timing constraints of search tasks in phased array radars are close to the task model in [14].

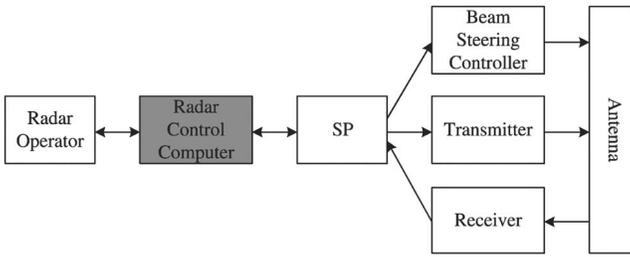


Fig. 1. Radar system architecture.

can be a multifunction radar that supports search, track, missile guidance, etc., simultaneously [2], [4], [8], [9], [16].

A typical phased array radar consists of several important components: Radar Control Computer (RCC), Signal Processor (SP), Beam Steering Controller (BSC), Receiver, Antenna, and Transmitter, as shown in Fig. 1 [2], [9], [23], [27] (here, Antenna is an array of antennae). RCC schedules beam transmission in a real-time fashion by sending SP radar beam commands; SP then controls BSC and Transmitter to transmit microwaves through Antenna. The reflected microwaves are received by Antenna and then passed to Receiver and SP for signal processing. The processed digital data is then reported to RCC by SP. RCC, which is the control and data processing unit of a radar system, must manage radar resources to perform radar functions, such as search, track, missile guidance, etc. RCC usually has interfaces with radar operators and external command, control, communication, and intelligence (C³I) system. For the rest of this paper, we only consider multifunction phased array radars.

2.2 Workload Characteristics

A phased array radar must scan its surveillance space periodically for suspicious targets, e.g., in terms of Horizon Search or Long Range Search tasks. The scan must be done in a hard real-time fashion. Such hard real-time search tasks are called *high-priority search tasks* in this paper. When suspicious targets are detected at RCC (because of reflected signals), a Track Confirmation task is issued for each detected target in the direction of the target to verify its presence. Track Confirmation tasks must be done in a hard real-time fashion to *identify* suspicious targets. Track Confirmation tasks can also be triggered by less critical searches, called *low-priority search tasks*, such as Normal Volume Search. Note that low-priority search tasks are conducted when there are free system resources available after servicing highly critical tasks. Operators might also choose to initiate Track Confirmation tasks because of

various reasons, as shown in Fig. 2. Once a target is identified, the corresponding Track Confirmation task is no longer needed. Instead, a sequence of “semiperiodic” Normal Track dwells (or beams) are issued to track the target. The transition of a Track Confirmation task to a Normal Track task may be requested by the operators, too [2]. The word *semiperiodic* means “periodic” but with dynamically changing periods. The reason for tracking being semiperiodic 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 Precision Track tasks, which are also “semiperiodic,” because the tracking of the target needs better precision. The transition of a Normal Track task to a Precision Track task may be requested by the operators [2]. High Precision Track 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 consists of high-priority search (HS) tasks, low-priority search (LS) tasks, High-Precision Track (HPT) tasks, Precision Track (PT) tasks, Normal Track (NT) tasks, and Track Confirmation (TC) tasks, as shown in Table 1. The scheduling of these tasks at RCC results in a dwell schedule for SP, where the executing of a dwell consists of RCC-SP command submission, round-trip radar beam travelling (including beam transmission and returning of the reflected signal), etc. [2].

RCC schedules tasks in units called *scheduling interval* (SI) [2], [8], [12], where the length of a scheduling interval (SI) is determined by various factors from the system specifications and usually in tens-of-milliseconds. In other words, RCC sends a sequence of commands to SP for dwell transmissions in the beginning of an SI and retrieves results from SP as well. With the synchronization behavior between RCC and SP and the definition of SI, radar tasks are always considered to arrive at a multiple of SI and have deadlines as multiples of SI. The responsibility of the RCC scheduler is to schedule dwells in the next SI, as shown in Fig. 3. Because of the regularity of search tasks and the scheduling slacks of other tasks, i.e., TC, NT, PT, and HPT tasks (to be explained later), it is assumed that RCC has enough time to schedule dwells in the next SI. Also, because RCC and SP synchronize with each other in the beginning of every SI, it makes no difference where a task is scheduled in an SI as long as the task is scheduled in the same SI. Note

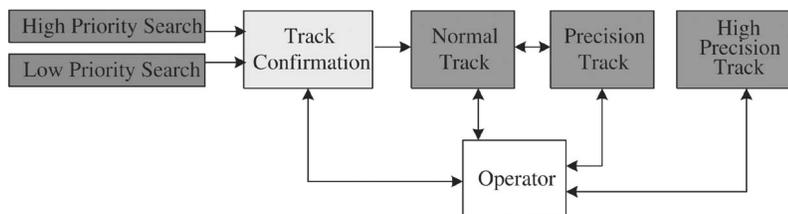


Fig. 2. Mode transitions of a phased array radar.

TABLE 1
Timing Parameters of a Typical Phased Array Radar

Task Types	Timing Constraints	Periodic	Deadline Type	Priority
High-Priority Search (HS)	$\{B_{HS}^i, C_{HS}^i, D_{HS}^i, P_{HS}^i\}$	periodic	hard deadline	1
Track Confirmation (TC)	$\{C_{TC}, D_{TC}\}$	aperiodic	hard deadline	2
High-Precision Track (HPT)	$\{C_{HPT}, DT_{HPT}, P_{HPT}^H, P_{HPT}^L\}$	semi-periodic	hard deadline	3
Precision Track (PT)	$\{C_{PT}, DT_{PT}, P_{PT}^H, P_{PT}^L\}$	semi-periodic	hard deadline	4
Normal Track (NT)	$\{C_{NT}, DT_{NT}, P_{NT}^H, P_{NT}^L\}$	semi-periodic	hard deadline	5
Low-Priority Search (LS)	$\{B_{LS}^i, C_{LS}^i, D_{LS}^i, P_{LS}^i\}$	periodic	soft deadline	6

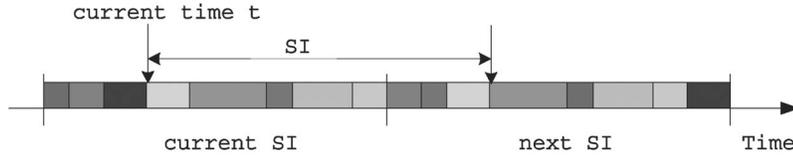


Fig. 3. Scheduling of radar tasks.

that each shaded area for a task represents the corresponding dwell of the task executing in the SI. The execution of the task is scattered over BSC, Receiver, Antenna, and Transmitter, instead of RCC. For the rest of this paper, we shall use terminologies “dwell” and “task” interchangeably.

An HS task must issue B_{HS}^i beams every P_{HS}^i time units, where C_{HS}^i is the dwell length. A dwell length consists of RCC-SP command submission time, hardware (including BSC) reset time, round-trip beam traveling time (including beam transmission time and returning time of the reflected signal) [2]. The relative deadline D_{HS}^i of the HS task is P_{HS}^i . For each suspicious target, a TC task is issued with a hard relative deadline D_{TC} and a dwell length C_{TC} . Note that all tasks of the same kind may share the same bounds on the timing constraints listed in Table 1 because they are usually well bounded by the same timing constraints at the RCC level [9]. Once the target is identified, a sequence of semiperiodic NT dwells is issued to track the target. The sequence of semiperiodic NT dwells is issued by a semiperiodic NT task with a maximum dwell length C_{NT} . The period of the NT task is bounded by a lower bound P_{NT}^L and an upper bound P_{NT}^H and each of its instances must be done in a hard real-time fashion, where the deadline is its period minus a dormant time DT_{NT} to leave enough slack for the RCC scheduler to schedule dwells in the next SI. The period of a PT (/HPT) task is also bounded by a lower bound P_{PT}^L (/ P_{HPT}^L) and an upper bound P_{PT}^H (/ P_{HPT}^H) and its hard deadline is also equal to its current period minus a dormant time DT_{PT} (/ DT_{HPT}). Its maximum dwell length is bounded by C_{PT} (/ C_{HPT}).

When there are available radar resources left in the next SI, the RCC scheduler may schedule LS tasks, such as that for Normal Volume Search, to search for suspicious targets. An LS task must issue B_{LS}^i beams every P_{LS}^i time units, where C_{LS}^i is the dwell length and its relative soft deadline D_{LS}^i is equal to P_{LS}^i . Although different phased array radars may have different system specifications and objectives, Table 1 lists a typical priority order of radar tasks, where 1 is the highest priority and 6 is the lowest priority.

3 REAL-TIME DWELL SCHEDULING OF PHASED ARRAY RADARS—A CASE STUDY

3.1 Motivation and Radar Task Scheduling

Traditional task scheduling at RCC adopts algorithms such as *Fixed-Template*, *Multi-Template*, and *Partial Template* [2]: The Fixed-Template algorithm is a cyclic executive algorithm which predetermines a task scheduling pattern in each SI. The scheduling logic is to assign specific radar tasks from the priority queues to match the predetermined pattern. The Multi-Template algorithm is extended from the Fixed-Template algorithm, where several templates, i.e., patterns, are available to have the best matching. Templates are predetermined with careful but costly offline analysis to reduce the runtime complexity in deriving a near-optimal scheduling solution. For example, we could have several predetermined templates in the system: a scheduling pattern with search but without any target tracking, that with 10 target tracking but without search, that with five target tracking and search, etc. When we need to have search and three targets to track simultaneously, the third pattern would be selected because of the best matching. We could use the scheduling slots reserved for five target tracking to track the three targets. The Partial Template algorithm (PM) is the most popular one, in which a portion of each SI is reserved for minimum operation guarantee, while the rest is open for competition using any priority-driven scheduling to satisfy immediate operational properties and equipment constraints. The reserved portion is usually for necessary tasks to maintain the minimum operation guarantee of the system, e.g., HS tasks, as shown in Fig. 4. The Fixed-Template and Multi-Template approaches are considered too restrictive for modern (multi-function) phased array radars, while the Partial Template algorithm is the one adopted most often.

Recently, researchers and engineers have started exploring real-time dwell scheduling at RCC and the proposed algorithms are mainly variations of the Partial Template algorithm. In particular, Huizing and Bloemen [9] proposed a double-queue priority-driven scheduling algorithm for dwell scheduling: One queue is for critical tasks, e.g.,

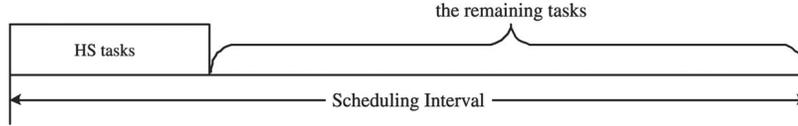


Fig. 4. Scheduling template by the Partial Template algorithm in an SI.

Terminal Illumination, which is for missile seeker locking. The other queue is for the remaining tasks. Both queues are FIFO queues. There is a limitation on the lengths of queues and dwell requests may be rejected based on their priorities and deadlines. No performance evaluation is reported. Izquierdo-Fuente and Casar-Corredera [12] dynamically partition each SI into two parts: tracking and surveillance. The system will use the first part of the SI to run as many as possible tasks for target tracking from the FIFO queue and then use the rest to run search tasks according to their priority order. Izquierdo-Fuente and Casar-Corredera [13] also proposed a dwell scheduling algorithm based on the neural network concept. They proposed a network layout and established a criterion for assigning weights and neuron activations to insert tasks into time slots. The algorithm may not be suitable for online usage because of the highly dynamic nature of radar workloads.

The major drawback of existing task scheduling algorithms, such as those based on the Partial Template algorithm, is on their restriction in resource utilization. Radar engineers usually consider the maximum workload of tasks at each priority level (e.g., those in Table 1), although radar workloads are highly dynamic. In order to guarantee the minimum radar operation, engineers usually use a large number of radar resources which may not be needed under proper real-time scheduling algorithms to accomplish their performance goals. In this paper, we present a rate-based approach for radar task scheduling to not only guarantee the minimum operation of a phased array radar (on both search and tracking) but also provide an easy way to estimate the system capacity. With the consideration of the workload characteristics of typical phased array radars, we demonstrate that better system performance can be achieved with a rate-based approach without sacrificing the stability of the system.

3.2 Real-Time Rate-Based Dwell Scheduling

3.2.1 GPS and TB-Server-Based Scheduling

In this paper, we propose a dwell scheduling algorithm based on the concept of GPS. Before we proceed with further discussions, the concept of GPS is summarized as a theoretical foundation: Parekh and Gallager [24] were the first to use the idea of GPS for rate-based flow and congestion control at network gateway nodes. GPS is a scheduling algorithm based on the concept of the reservation ratio of processor computation time. Suppose a GPS server executes at a fixed rate r and each session i is characterized by a positive real number θ_i , $i = 1, 2, \dots, N$ (θ_i is called the *reservation ratio* of traffic session i). Each session i is guaranteed to be served at a rate of

$$g_i = \frac{\theta_i}{\sum_j \theta_j} r.$$

Suppose that $S_i(t_1, t_2)$ is the amount of session τ_i served in the interval $[t_1, t_2]$. For any session i backlogged² in the interval $[t_1, t_2]$,

$$\frac{S_i(t_1, t_2)}{S_j(t_1, t_2)} \geq \frac{\theta_i}{\theta_j}, j = 1, 2, \dots, N.$$

With leaky-bucket-based admission control [30], a worst-case guarantee on throughput and delay can be guaranteed.

GPS is a work-conserving algorithm and can guarantee 100 percent server utilization. However, GPS is an idealized multiplexing discipline for packet transmission since it assumes that the traffic is infinitely divisible such that the server can serve multiple traffic sessions simultaneously. In [26], Spuri and Buttazzo proposed an effective GPS-based mechanism called *TB server* to service sporadic processes. The schedulability of a TB server and other processes in the system is guaranteed under the framework of the EDF scheduling. Suppose that a TB server TB_i with a reservation ratio θ_i is used to service a collection of aperiodic tasks. Let the k th aperiodic task of TB_i arrive at time t and its maximum CPU requirement be $c_{i,k}$. The deadline of the k th aperiodic task is defined as:

$$d_{i,k} = \max\{t, d_{i,k-1}\} + \frac{c_{i,k}}{\theta_i},$$

where θ_i is called the server utilization factor in [26]. By definition, $d_{i,0} = 0$. The request is inserted into the ready queue of the system and scheduled by EDF.

The *constant utilization server* (CUS) proposed by Deng and Liu [7] is a variation of the TB server in which the OS scheduler sets the CUS budget to the execution time of the job at the ready queue of the CUS. CUS is proposed to reserve CPU bandwidth for applications which adopt nonpreemptive scheduling algorithms, especially when the arrival times of processes are unpredictable. Applications which share global resources are considered. They showed an important property for CUS and TB-server-based scheduling:

Theorem 1 [7]. *A real-time application A_k whose required capacity is $U_k < 1$ is schedulable in the open system when it is executed by a server S_k with a reservation ratio θ_i provided that all of the following conditions are true:*

1. *If A_k adopts some preemptive algorithm, a TB server is adopted. If A_k adopts some nonpreemptive algorithm, a CUS server is adopted.*

2. *A session is backlogged at time t if a positive amount of that session's traffic is queued at time t .*

2. If A_k is a nonpreemptive application, θ_i is equal to U_k . At each replenishment time of the server, the budget is set to the execution time of the job at the head of the ready queue.
3. The total size of all servers in the open system is no more than $(1 - \max_{j \geq 1} \{B_j/\delta_j\})$, where B_j is the maximum duration of nonpreemptable sections of all applications other than application A_j and δ_j is the shortest relative deadline of all jobs in A_j .

$\max_{j \geq 1} \{B_j/\delta_j\}$ is called the *maximum global blocking time* for RCC dwell scheduling later in this paper.

3.2.2 Real-Time Rate-Based Dwell Scheduling—The Basic Approach

The performance specifications of a phased array radar often contain a collection of HS tasks, a minimum number of targets to be tracked simultaneously by the system, and a minimum number of HPT tasks to be serviced simultaneously by the system, where HPT tasks are needed in many attack-oriented or even defense-oriented systems, e.g., for missile guidance. In this section, we shall first define the reservation ratio θ_i of tasks to satisfy the performance specifications. We will then present an efficient TB-server-based scheduling algorithm, called the *Batch-TB-Server-Based Scheduling* algorithm (Batch-TB), for dwell scheduling at RCC. When TB servers are adopted, the virtual deadlines and budgets of the servers are set at the beginning of an SI. It is because the information of dwells issued in each SI is known in the previous SI. Note that the Batch-TB algorithm might use all available bandwidth beyond the reserved bandwidth of a task.

Let $\Pi = \{\tau_{HS}^1, \tau_{HS}^2, \dots, \tau_{HS}^N\}$ be a collection of HS tasks necessary for the minimum operation of the system. The reservation ratio θ_{HS}^i of each HS task τ_{HS}^i is set as

$$\frac{B_{HS}^i C_{HS}^i}{P_{HS}^i},$$

where B_{HS}^i , C_{HS}^i , and P_{HS}^i are the number of beams, dwell length, and period of τ_{HS}^i , respectively. In each j th period P_{HS}^i of τ_{HS}^i , τ_{HS}^i initiates B_{HS}^i instances, where each instance has a dwell length C_{HS}^i . LS tasks do not have a reservation ratio because they only utilize unused dwell time in each SI. In each j th period P_{LS}^i of an LS task τ_{LS}^i , τ_{LS}^i initiates B_{LS}^i instances, where each instance has a dwell length C_{LS}^i .

Because the tracking of an identified target starts with a CT task, an NT task, and possibly a PT task, the reservation ratio of each target-tracking task is defined as

$$\max \left\{ \frac{C_{TC}}{D_{TC}}, \frac{C_{NT}}{(P_{NT}^L - DT_{NT})}, \frac{C_{PT}}{(P_{PT}^L - DT_{PT})} \right\},$$

where C_{TC} and D_{TC} are the dwell length and relative deadline of a TC task, respectively, and C_{NT} , DT_{NT} , and P_{NT}^L ($/C_{PT}$, DT_{PT} , and P_{PT}^L) are the dwell length, the dormant time, and the lower bound of the period of an NT ($/PT$) task, respectively. Note that each target-tracking task first appears as a TC task because no target is identified yet. A TC task is then replaced with an NT task with a target being identified and tracked. An NT task might be removed because tracking is no longer needed for the corresponding

target or it might be replaced with a PT task because a higher tracking precision is needed.

An HPT task issues one beam per period and the timing constraints of all HPT tasks are the same (as explained in previous sections). The reservation ratio of an HPT task is defined as

$$\frac{C_{HPT}}{(P_{HPT}^L - DT_{HPT})},$$

where C_{HPT} , DT_{HPT} , and P_{HPT}^L are the dwell length, the dormant time, and the lower bound of the period of an HPT task, respectively. The above settings of reservation ratios consider the worst-case resource reservation.

The *Batch-TB-Server-Based Scheduling* algorithm (Batch-TB) is a batch version of the TB-server scheduling algorithm [26], where dwell scheduling is done for every SI and no preemption is allowed. The nonpreemptivity of dwell scheduling is because each operation of BSC, SP, Transmitter, Receiver, and Antenna is indivisible and each dwell is processed as a pipelined logical unit. The reservation ratios for different servers are presented in the previous paragraph.

Let a task τ_i have a reservation ratio θ_i . Each instance of the task is given a *virtual deadline* for EDF scheduling. Suppose that the j th instance of τ_i arrives at time t and the virtual deadline of the $(j-1)$ th instance of τ_i is $d_{i,j-1}$. The virtual deadline $d_{i,j}$ of the j th instance of τ_i is defined as follows: Suppose that the dwell length of the instance is $c_{i,j}$.

$$d_{i,j} = \max\{t, d_{i,j-1}\} + \frac{c_{i,j}}{\theta_i}.$$

By definition, $d_{i,0} = 0$. The request is inserted into the ready queue of the system and scheduled by EDF in the next SI. In other words, the RCC scheduler can get the information of tasks arriving in the next SI. When there are multiple instances of a task (such as search tasks) arriving at the same time, their deadlines are set as above and all inserted into the ready queue. Note that only dwells which can fit in one SI are scheduled in the next SI and the scheduling of the rest will be delayed until the end of the next SI.

Example 1: A Batch-TB Schedule. Suppose that one HS task, two HPT tasks, two PT tasks, and one NT task arrive at the beginning of the i th SI and one HPT task and one PT task arrive at the beginning of the $(i+1)$ th SI, as shown in Fig. 5. Let the HS task τ_{HS} issue 45 instances (for simplicity in the presentation of the Batch-TB algorithm, assume that all instances of τ_{HS} issued in previous SIs were processed).

The timing constraints of all tasks are as shown in Table 2. According to the definitions of the reservation ratios, the reservation ratio of τ_{HS} is $(B_{HS}^i C_{HS}^i)/P_{HS}^i = (45 * 6ms)/1000ms = 0.27$. Let the dormant time of every track task be one SI, i.e., $25ms$. The reservation ratio for each HPT task is derived by the following formula:

$$\frac{C_{HPT}}{(P_{HPT}^L - DT_{HPT})} = \frac{2ms}{75ms} = 0.02666.$$

Let the dwell length C_{TC} and deadline D_{TC} of a TC be $6ms$ and $20SI$, respectively. The reservation ratio for

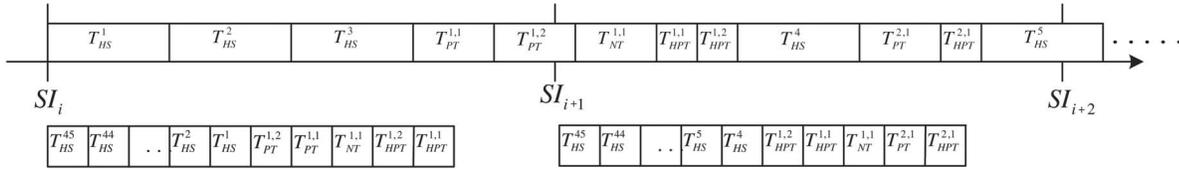


Fig. 5. A Batch-TB schedule.

 TABLE 2
 Task Parameters ($1SI = 25ms$)

Tasks	Number of Dwells	Dwell Length	Period	Reservation Ratio
HS task (τ_{HS})	45	6ms	40SI	0.27
HPT tasks ($\tau_{HPT}^{1,1}$, $\tau_{HPT}^{1,2}$, and $\tau_{HPT}^{2,1}$)	1	2ms	(4SI, 10SI)	0.02666
NT/PT tasks ($\tau_{PT}^{1,1}$, $\tau_{PT}^{1,2}$, $\tau_{NT}^{1,1}$, and $\tau_{PT}^{2,1}$)	1	$C_{PT} = 4ms$ $C_{NT} = 4ms$	(4SI, 10SI) for $\tau_{PT}^{1,1}$, $\tau_{PT}^{1,2}$, and $\tau_{PT}^{2,1}$ (10SI, 80SI) for $\tau_{NT}^{1,1}$	0.05333

When a period range (LP, UP) is provided, LP and UP denote the lower and upper bounds of the period, respectively.

each target-tracking task is derived by the following formula:

$$\max \left\{ \frac{C_{TC}}{D_{TC}}, \frac{C_{NT}}{(P_{NT}^L - DT_{NT})}, \frac{C_{PT}}{(P_{PT}^L - DT_{PT})} \right\}$$

$$= \max \left\{ \frac{6ms}{500ms}, \frac{4ms}{225ms}, \frac{4ms}{75ms} \right\} = 0.05333.$$

The different reservation ratios calculated above are for servers of different tasks. The virtual deadlines of instances of τ_{HS} , $\tau_{HPT}^{1,1}$, $\tau_{HPT}^{1,2}$, $\tau_{PT}^{1,1}$, $\tau_{PT}^{1,2}$, and $\tau_{NT}^{1,1}$ in SI_i are listed in Table 3a. Those of instances of τ_{HS}^i ($4 \leq i \leq 45$), $\tau_{HPT}^{1,1}$, $\tau_{HPT}^{1,2}$, and $\tau_{NT}^{1,1}$ in SI_i and those of instances of $\tau_{HPT}^{2,1}$ and $\tau_{PT}^{2,1}$ in SI_{i+1} are listed in Table 3b.

Tasks are sorted in the ready queue of the Batch-TB scheduling algorithm and scheduled in the i th SI, as shown in Fig. 5. Note that only τ_{HS}^1 , τ_{HS}^2 , τ_{HS}^3 , $\tau_{PT}^{1,1}$, and $\tau_{PT}^{1,2}$ are scheduled in the i th SI. $\tau_{NT}^{1,1}$, $\tau_{HPT}^{1,1}$, $\tau_{HPT}^{1,2}$, τ_{HS}^4 , $\tau_{PT}^{2,1}$,

$\tau_{HPT}^{2,1}$, and τ_{HS}^5 are scheduled in the $(i+1)$ th SI. As astute readers may notice, the execution of $\tau_{PT}^{1,2}$ crosses over the i th and $(i+1)$ th SIs. The execution of τ_{HS}^5 crosses over the $(i+1)$ th and $(i+2)$ th SIs. Because of the non-preemptibility of dwells, RCC cannot schedule any urgent dwell in the beginning of the $(i+1)$ th and $(i+2)$ th SI until $\tau_{PT}^{1,2}$ and τ_{HS}^5 finish their executions, respectively. Such a blocking phenomenon can be modeled by the blocking time due to the global resource synchronization presented in Theorem 1. In the following section, we shall show the schedulability formula for radar task scheduling.

3.2.3 Properties and System Capacity Estimation

The purpose of this section is to show important properties of Batch-TB. In particular, we shall derive a formula which quantifies the relationship between the system specifications and the reservation ratios of radar tasks.

The RCC scheduler has the information of the tasks that will arrive at the beginning of the $(i+1)$ th SI in the i th SI. Because of the predictability of task arrivals in the next SI, RCC can avoid preemptive executions. TB servers for a specific task is said to be backlogged when it is associated with a yet-to-be-completed task. When the task finishes its execution, the corresponding server will become idle. For example, in Example 1, there are one TB server for the HS task instances, two TB servers for the PT tasks, one TB server for the NT task, and three TB servers for the HPT tasks.

Corollary 1. Given N radar tasks, if every radar task τ_i reserves a reservation ratio θ_i according to the amount defined in the previous section and the total size of all servers in the system is no more than $(1 - \max_{j \geq 1} \{B_j/\delta_j\})$, then every radar task can issue the specified number of beams before its deadline, where B_j is the maximum dwell length of all radar tasks other than τ_j and δ_j is the shortest relative deadline of all dwells in τ_j .

TABLE 3
 The Virtual Deadlines of Tasks Ready at the Beginning of SI_i and SI_{i+1} ($1SI = 25ms$):
 (a) at the Beginning of SI_i , (b) at the Beginning of SI_{i+1}

Tasks	Virtual Deadline
τ_{HS}^1	$SI_i + 22.22$
τ_{HS}^2	$SI_i + 44.44$
τ_{HS}^3	$SI_i + 66.66$
τ_{HS}^4	$SI_i + 88.88$
...	...
τ_{HS}^{45}	$SI_i + 999$
$\tau_{HPT}^{1,1}$	$SI_i + 75.018$
$\tau_{HPT}^{1,2}$	$SI_i + 75.018$
$\tau_{PT}^{1,1}$	$SI_i + 75.005$
$\tau_{PT}^{1,2}$	$SI_i + 75.005$
$\tau_{NT}^{1,1}$	$SI_i + 75.005$

(a)

Tasks	Virtual Deadline
τ_{HS}^4	$SI_i + 88.88$
τ_{HS}^5	$SI_i + 111.10$
τ_{HS}^6	$SI_i + 133.32$
τ_{HS}^7	$SI_i + 155.52$
...	...
τ_{HS}^{45}	$SI_i + 999$
$\tau_{HPT}^{1,1}$	$SI_i + 75.018$
$\tau_{HPT}^{1,2}$	$SI_i + 75.018$
$\tau_{NT}^{1,1}$	$SI_i + 75.005$
$\tau_{HPT}^{2,1}$	$SI_{i+1} + 75.018$
$\tau_{PT}^{2,1}$	$SI_{i+1} + 75.005$

(b)

Proof. The correctness of this corollary directly follows from Theorem 1. \square

Note that if a radar task is an HS task, τ_{HS}^i is guaranteed to issue B_{HS}^i beams per P_{HS}^i units of time although multiple instances of the search task are issued in the beginning of each of its periods. Note that Corollary 1 provides not only the schedulability analysis foundation for dwell scheduling, but also a formula to derive the system capacity of RCC: Suppose that RCC must support a collection of HS tasks $\Pi = \{\tau_{HS}^1, \tau_{HS}^2, \dots, \tau_{HS}^N\}$, a minimum number Num_t of simultaneous target-tracking tasks (i.e., TC, NT, or PT tasks), and a minimum number Num_{ht} of simultaneous HPT tasks. The following formula must be satisfied:

$$\left(\sum_{\tau_{HS}^i \in \Pi} \frac{B_{HS}^i C_{HS}^i}{P_{HS}^i} \right) + Num_t \left(\max \left\{ \frac{C_{TC}}{D_{TC}}, \frac{C_{NT}}{(P_{NT}^L - DT_{NT})}, \frac{C_{PT}}{(P_{PT}^L - DT_{PT})} \right\} \right) + Num_{ht} \left(\frac{C_{HPT}}{(P_{HPT}^L - DT_{HPT})} \right) \leq \left(100\% - \max_{j \geq 1} \{B_j / \delta_j\} \right).$$

Since C_{HS}^i is usually larger than other dwell lengths and the deadline of the HS task is relatively large, the maximum global blocking time $\max_{j \geq 1} \{B_j / \delta_j\}$ is equal to

$$\max \{ C_{HS}^i / \min \{ D_{TC}, (P_{NT}^L - DT_{NT}), (P_{PT}^L - DT_{PT}), (P_{HPT}^L - DT_{HPT}) \} \}.$$

3.3 Remark on Rate Reservation: Application Semantics

3.3.1 Reservation Ratio Sharing: Task Modes

The reservation ratio for each target-tracking task considers the worst-case radar resource utilization. Due to different resource requirements of target-tracking tasks at different stages, a single reservation ratio seems restrictive on the entire system utilization.³ The purpose of this section is to explore the ratio of target-tracking tasks in different modes to maximize the system utilization and to guarantee a better system capacity.

One common observation in system specifications and implementations of phased array radars is to have a worst-case ratio between the numbers of tasks which are in the Precision Tracking and Normal Tracking modes, where tasks in the Precision Tracking mode are more demanding in resources than tasks in the Normal Tracking mode. In other words, if there are M NT tasks, then there are at most $R_{P/N} \cdot M$ PT tasks, where $R_{P/N}$ is the worst-case ratio between the numbers of tasks which are in the Precision Tracking and Normal Tracking modes.

Suppose that a phased array radar must support a minimum number Num_t of simultaneous target tracking and $R_{P/N}$ is the worst-case ratio between the number of tasks which are in the Precision Tracking and Normal Tracking modes. Instead of reserving a reservation ratio for a target-tracking task enough for the three modes, the system first assigns the task a reservation ratio $R_{C\&N}$

enough for its execution in the Track Confirmation and Normal Tracking modes:

$$R_{C\&N} = \max \left\{ \frac{C_{TC}}{D_{TC}}, \frac{C_{NT}}{(P_{NT}^L - DT_{NT})} \right\}.$$

A sharable reservation ratio GR_{PT} is reserved to guarantee the upgrading of Num_t tasks from the Normal Tracking mode to the Precision Tracking mode:

$$GR_{PT} = Num_t \left(\max \left\{ \frac{C_{TC}}{D_{TC}}, \frac{C_{PT}}{(P_{PT}^L - DT_{PT})} \right\} - R_{C\&N} \right).$$

When the mode of a task needs to be changed from the Normal Tracking mode to the Precision Tracking mode, the system increases the reservation ratio of the task dynamically by withdrawing the required ratio from GR_{PT} . In this way, the system could guarantee the simultaneous tracking of Num_t targets in any of the three modes. Note that $GR_{PT} = 0$ when $\frac{C_{TC}}{D_{TC}}$ is no less than $\frac{C_{NT}}{(P_{NT}^L - DT_{NT})}$ and $\frac{C_{PT}}{(P_{PT}^L - DT_{PT})}$. That is, the concept of the reservation ratio sharing does not have any advantage in the saving of reservation ratios. Such a case might happen when the dwell length for Track Confirmation is relatively long or when the deadline for Track Confirmation is fairly short. An example system is an antiballistic missile radar system. The formula for estimating the system capacity in the previous section can be changed, as follows:

Suppose that RCC must support a collection of HS tasks $\Pi = \{\tau_{HS}^1, \tau_{HS}^2, \dots, \tau_{HS}^N\}$, a minimum number Num_t of simultaneous target tracking, and a minimum number Num_{ht} of simultaneous HPT tasks. The following formula must be satisfied:

$$\left(\sum_{\tau_{HS}^i \in \Pi} \frac{B_{HS}^i C_{HS}^i}{P_{HS}^i} \right) + Num_t \left(\max \left\{ \frac{C_{TC}}{D_{TC}}, \frac{C_{NT}}{(P_{NT}^L - DT_{NT})} \right\} \right) + GR_{PT} + Num_{ht} \left(\frac{C_{HPT}}{(P_{HPT}^L - DT_{HPT})} \right) \leq \left(100\% - \max_{j \geq 1} \{B_j / \delta_j\} \right).$$

Obviously, if the resource demand of a task in the Track Confirmation mode is more than that in the Normal Tracking mode, then the system should initially assign a reservation ratio (C_{TC}/D_{TC}) to a target-tracking task. When a target-tracking task steps into the Normal Tracking mode, the system can reduce its reservation ratio from (C_{TC}/D_{TC}) to $\frac{C_{NT}}{(P_{NT}^L - DT_{NT})}$. When the task needs to upgrade itself to the Precision Tracking mode, the system increases its reservation ratio to $\frac{C_{PT}}{(P_{PT}^L - DT_{PT})}$ if there is still some reserved ratio available. Note that, in this case, the above formula for estimating the system capacity can be modified accordingly.

3. A target-tracking task can be a TC, NT, or PT task.

3.3.2 Search Workload Distribution: Alternatives in Resource Reservation

In the previous section, the reservation ratio θ_{HS}^i of each HS task τ_{HS}^i is set as $(B_{HS}^i C_{HS}^i) / P_{HS}^i$. In each j th period P_{HS}^i of τ_{HS}^i , τ_{HS}^i initiates B_{HS}^i instances, where each instance has a dwell length C_{HS}^i . The virtual deadline difference of the B_{HS}^i instances assigned by Batch-TB is $(C_{HS}^i / \theta_{HS}^i) = (P_{HS}^i / B_{HS}^i)$. In fact, the deadline difference of every two consecutive instances is exactly (P_{HS}^i / B_{HS}^i) for the HS task τ_{HS}^i . Batch-TB tends to derive a good scheduling pattern with small jitters in issuing beams because the deadlines of the B_{HS}^i instances of an HS task are evenly scattered in its period.

As astute readers may point out, an extreme alternative is to split each HS task τ_{HS}^i into B_{HS}^i subtasks $\tau_{i,j}^{HS}$ and to assign each $\tau_{i,j}^{HS}$ an equal reservation ratio C_{HS}^i / P_{HS}^i . We must point out that such an approach will result in the same virtual deadline P_{HS}^i for each of the subtasks of $\tau_{i,j}^{HS}$ in every P_{HS}^i . As a consequence, the B_{HS}^i dwells of τ_{HS}^i in each of its periods may have a less chance to scatter evenly in its period.

Other alternatives include the splitting of each HS task τ_{HS}^i into a lesser number of subtasks. As observed in previous paragraphs, the more subtasks split for HS tasks, the worse the execution jitter might be. We must point out that splitting of search tasks does not help the scheduling of tracking tasks with a reservation ratio. However, splitting does potentially help the scheduling of tracking tasks without reservation ratios and LS tasks (which have no reservation ratios) because splitting of subtasks results in longer virtual deadlines for search task instances and, therefore, is less demanding of radar resources (in terms of deadlines). Since search tasks must have a good jitter execution template to have a quality result, we conclude that, in rate-based real-time dwell scheduling, no splitting is preferred for search tasks.

3.3.3 Reclamation of Reservation Ratio

Each target-tracking task under Batch-TB is assigned a reservation ratio when the tracking of the corresponding target is guaranteed. The reservation ratio of each target-tracking task considers the worst-case resource utilization, i.e., the maximum usage of the radar resource (please see Section 3.2.2). Because the upper and lower bounds of the tracking period for each target-tracking task might differ by a large number, radar resource tends to be reserved very conservatively. As a result, much radar resource is wasted. A similar phenomenon can be observed for the reservation ratio of HPT tasks. The purpose of this section is to propose a mechanism to reclaim the wasted radar resource that is reserved conservatively due to the guarantee of the minimum operation.

Let P^L and P^H be the lower and upper bounds of the period of a task τ , where a task here could be an NT, PT, or HPT task. (Note that the HS task, the LS task, and target-tracking tasks in the Track Confirmation mode are not discussed here because they not semiperiodic.) Suppose that the timely servicing of τ is guaranteed with the worst-case resource utilization. In other words, the reservation ratio of τ is reserved, as presented in Section 3.2.2.

Since the reservation ratio of τ is reserved under the worst-case consideration (i.e., with the lower period bound), RCC

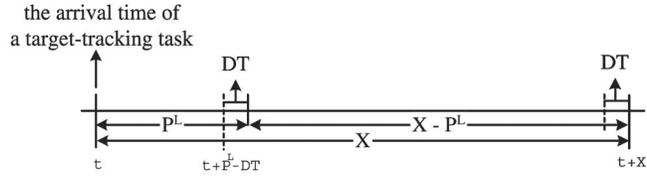


Fig. 6. The reclamation duration of a reservation ratio.

could always finish its dwell within $(P^L - DT)$ amount of the time with the assigned reservation ratio, where DT is the dormant time of τ . Since τ is semiperiodic, the period of τ changes from time to time. Let the period of τ be X at this time point t , where $X \geq P^L$. In other words, the reservation ratio for τ is not used between time $t + (P^L - DT)$ and time $t + X$, as shown in Fig. 6. In other words, we can reclaim the reservation ratio of τ in an interval equal to $(X - P^L + DT)$ and return the reservation ratio of τ during this interval back to RCC to schedule other “nonguaranteed” tasks. Since the reclamation of the reservation ratio of τ happens after τ is serviced, the reclamation of the reservation ratio does not hurt the schedulability guarantee of τ . The guarantee for the returning of the reclaimed reservation ratio presents no technical difficulty provided that the virtual deadline of any tracking task which uses the reclaimed reservation ratio is no later than the time when the reclaimed reservation ratio must be returned. A single system server can use reclaimed reservation ratios to service any “nonguaranteed” tasks! Such a reclamation of reservation ratios can be applied to any NT, PT, or HPT task and make the radar resource better utilized.

4 SIMULATION EXPERIMENTS

4.1 Data Set and Measurement

The experiments described in this section are meant to assess the capability of the Batch-TB algorithm with reservation ratio reclamation in dwell scheduling. We have implemented a simulation model for a multifunction phased array radar. We compare the performance of the Batch-TB algorithm (with reservation ratio reclamation), the earliest deadline first algorithm (EDF), and the traditional Partial Template scheduling algorithm (PM) with and without an even search workload distribution. When an even search workload distribution was adopted for the traditional Partial Template scheduling algorithm, dwells of each period were evenly distributed in each SI.

The primary performance metric is the ratio of requests that miss deadlines, referred to as the *Miss Ratio*. Let num_i and $miss_i$ be the total number of task requests and deadline violations during an experiment, respectively. *Miss Ratio* is calculated as $\frac{miss_i}{num_i}$.

The test data sets were generated based on a multifunction phased array radar for air defense frigates [9]. There was only one HS task, which had to issue 45 beams for every 40SI, where 40SI was equivalent to one second, and each dwell length was 6ms. TC tasks, which were not periodic, had a deadline equal to 20SI. Their dwell lengths were 6ms. The lower and upper period bounds of HPT tasks were 4SI and 10SI, respectively. The dwell length of

TABLE 4
Task Parameters of Simulation Experiments ($1SI = 25ms$)

Task Types	Dwell Length	Beams per Period	Period	Priority
HS task	$6ms$	45 beams	$40SI$	1
TC task	$6ms$	1 beam	$Deadline = 20SI$	2
HPT task	$2ms$	1 beam	$(4SI, 10SI)$	3
PT task	$4ms$	1 beam	$(4SI, 10SI)$	4
NT task	$4ms$	1 beam	$(10SI, 80SI)$	5
LS task	$2ms$	20 beams	$40SI$	6

When a period tange (LP, UP) is provided, LP and UP denote the lower and upper bounds of the period, respectively.

TABLE 5
Simulation Parameters ($1SI = 25ms$)

Parameters	Value
The reservation ratio of the HS task	0.27
The reservation ratio of each target-tracking task	0.05333
The reservation ratio of each HPT task	0.02666
The guaranteed number Num_t of simultaneous tracked targets	9
The guaranteed number Num_{ht} of simultaneous HPT tasks	4
Dormant time for all tasks	$1SI = 25ms$
Simulation time	$400,000SI$

each HPT task was $2ms$. The lower and upper period bounds of PT tasks were $4SI$ and $10SI$, respectively. The dwell length of each PT task was $4ms$. The lower and upper period bounds of NT tasks were $10SI$ and $80SI$, respectively, where both bounds of NT tasks were larger than those of PT tasks. The dwell length of each NT task was $4ms$. There was one LS task (for Normal Volume Search) in the system, which issued 20 beams for every $40SI$, and each dwell length was $2ms$. The parameters of the tasks are summarized in Table 4.

Let the dormant time of all NT, PT, and HPT tasks be one SI, i.e., $25ms$. According to the task parameters in Table 4, the reservation ratio of the HS task was

$$(B_{HS}^i C_{HS}^i) / P_{HS}^i = (45 * 6ms) / 1000ms = 0.27$$

and the maximum global blocking time

$$b_{max}(np) / D_{min} = (6ms / 75ms) = 0.08.$$

Suppose that the rest of the RCC resource was partitioned for target-tracking tasks (i.e., TC, NT, and PT tasks) and HPT tasks, and the system specifications offered 80 percent of the remaining radar resource for target-tracking tasks. The reservation ratio for each target-tracking task was derived by the following formula:

$$\begin{aligned} & \max \left\{ \frac{C_{TC}}{D_{TC}}, \frac{C_{NT}}{(P_{NT}^L - DT_{NT})}, \frac{C_{PT}}{(P_{PT}^L - DT_{PT})} \right\} \\ & = \max \left\{ \frac{6ms}{500ms}, \frac{4ms}{225ms}, \frac{4ms}{75ms} \right\} = 0.05333. \end{aligned}$$

The reservation ratio for each HPT task was derived by the following formula:

$$\frac{C_{HPT}}{(P_{HPT}^L - DT_{HPT})} = \frac{2ms}{75ms} = 0.02666.$$

Because 80 percent of the remaining radar resource was reserved for target-tracking tasks, the system specifications could guarantee that $Num_t = \lfloor (0.8 * (1 - 0.27 - 0.08)) / 0.05333 \rfloor = 9$ targets could be tracked simultaneously. With 20 percent of the remaining radar resource left by the HS task, the system specifications could guarantee the timely services of $Num_{ht} = \lfloor (0.2 * (1 - 0.27 - 0.08)) / 0.02666 \rfloor = 4$ HPT tasks simultaneously.

The experiments simulated task sets with HPT tasks and target-tracking tasks of different modes. The number of tasks per task set ranged from 20 to 100 and the arrival pattern of each task had a Poission distribution. The ratio of a task set size to the number of its TC tasks was about 10 : 1. When Batch-TB was adopted, radar resource was reserved for each task when it arrived. Each task set was simulated for 400,000 SI, which was equivalent to 10,000 seconds. Ten task sets per workload were simulated and their results were averaged. Note that a target-tracking task might have three modes: Track Confirmation, Normal Tracking, and Precision Tracking. In the experiments, the ratio of the number of target-tracking tasks (i.e., CT, NT, and PT tasks) to the number of HPT tasks remained 2 : 1. For example, when the total number of target-tracking tasks and HPT tasks was 30, the numbers of TC tasks, HPT tasks, and NT/PT tasks were 3, 10, and 17, respectively. The simulation parameters are summarized in Table 5.

4.2 Experimental Results

Fig. 7a shows the miss ratio of the HS task. Since both Batch-TB and PM guaranteed its service, its miss ratio remained zero, regardless of the number of target-tracking and HPT tasks in the system. On the other hand, EDF started missing

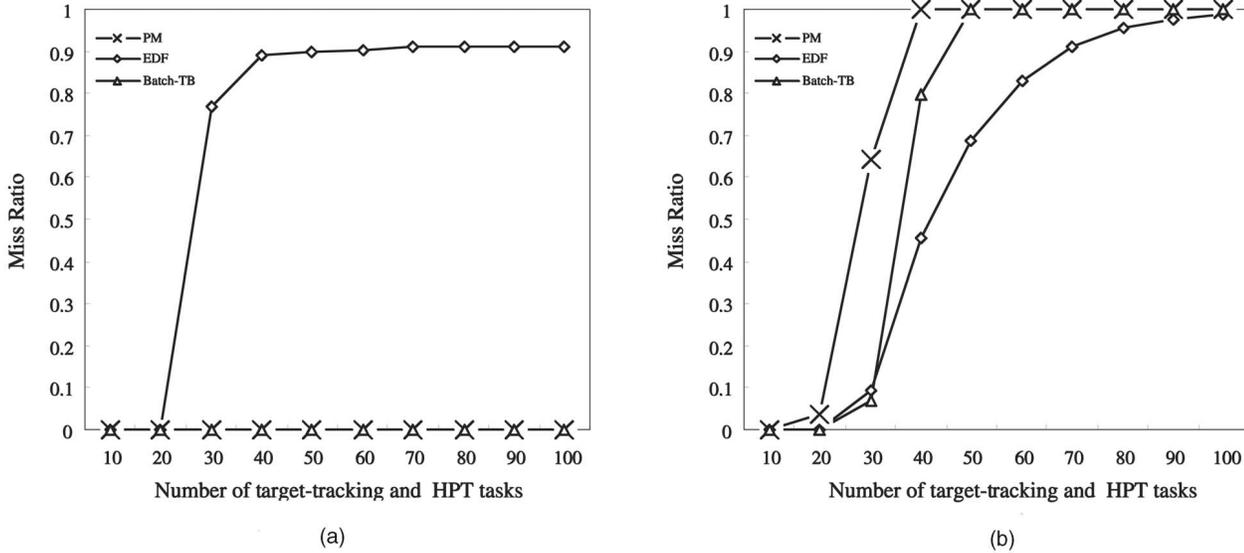


Fig. 7. (a) and (b) are the miss ratio of the HS task and HPT tasks, without an even search workload distribution, respectively.

most of the deadlines of the HS task when the number of tracking tasks in the system was over 20 (which is not acceptable for a phased array radar).

Fig. 7b shows the miss ratio of HPT tasks. Note that, in the experiments, the ratio of target-tracking tasks to HPT tasks remained 2 : 1. Therefore, Batch-TB and EDF could both schedule up to seven HPT tasks without missing any of their deadlines and the miss ratio of HPT tasks remained low until the number of HPT tasks was around 10, i.e., the number of target-tracking and HPT tasks was over 30. On the other hand, PM could only schedule four HPT tasks without missing any of their deadlines (here, HPT tasks were favored under PM). In other words, Batch-TB and EDF could improve the performance of RCC by 43 percent (i.e., 3/7). However, we must point out that EDF would suffer from overload easily and could not maintain the minimum operation guarantee of the system.

Fig. 8a shows the miss ratio of all target-tracking tasks. Batch-TB and EDF also greatly outperformed PM. Batch-TB and EDF could both schedule 13 target-tracking tasks without missing any of their deadlines. On the other hand, PM could only schedule six target-tracking tasks, respectively, without missing any of their deadlines. Batch-TB and EDF could both yield the 116 percent (i.e., $(13/6) - 1$) improvement, compared with PM. There were two main reasons why EDF performed better than PM: First, it was because of the common pool of radar resource in scheduling search and target-tracking tasks. Second, it was because TC tasks had a higher static priority than NT and PT tasks under PM and the dwell length of a TC task was much larger than those of NT and PT tasks. In general, the performance of Batch-TB was better than that of EDF because the miss ratio of target-tracking tasks under Batch-TB remained very low until the number of target-tracking and HPT tasks was 30. Because the ratio of target-tracking tasks to HPT tasks remained 2 : 1, the number of the target-tracking tasks was 20. In other words, Batch-TB could track up to nearly 20 targets! Batch-TB outperformed EDF because target-tracking tasks competed

with each other seriously such that there was a little chance for everyone to meet its deadline and Batch-TB prevented such resource competition.

Fig. 8b shows the miss ratio of the LS task. It was surprising to see that Batch-TB always met the deadlines of the LS task, regardless of the number of target-tracking and HPT tasks in the system. It was because the period of the LS task was long and Batch-TB happened to have bandwidth left in the reservation for target-tracking tasks, HPT tasks, and the HS task. PM could hardly schedule the LS task when the system was heavily loaded because it had the lowest static priority and other tasks always received RCC's attention. Under EDF, the large deadline of the LS task put itself at the bad side in resource competition.

Figs. 9 and 10 show the miss ratios of the HS task, HPT tasks, target-tracking tasks, and the LS task, when an even search workload distribution was adopted. Their results were about the same as their counterparts in Figs. 7 and 8. We must point out that, although the performance of all RCC scheduling algorithms remained the same, when an even search workload distribution was adopted, the signal processor of a phased array radar could benefit a lot from the even search workload distribution because of even workload distribution. We refer interested readers to [20] for details.⁴

It was clear that Batch-TB could significantly improve the performance of a phased array radar by not only guaranteeing the minimum operation (e.g., the servicing of the HS task) but also greatly increasing the numbers of target-tracking tasks and HPT tasks in the system. In general, the capacity of a phased array radar under Batch-TB could be close to that under EDF (although Batch-TB was slightly better). Batch-TB

4. A task set without an even search workload distribution would have many more beams being issued in the first several SIs of a search task execution, when an earliest-deadline-first technique was adopted. Such a phenomenon might result in a very unbalanced distribution of workloads in some period of a search task. As a result, the radar system must have much more resource deployed to guarantee the timely processing of the returned signals of a search task.

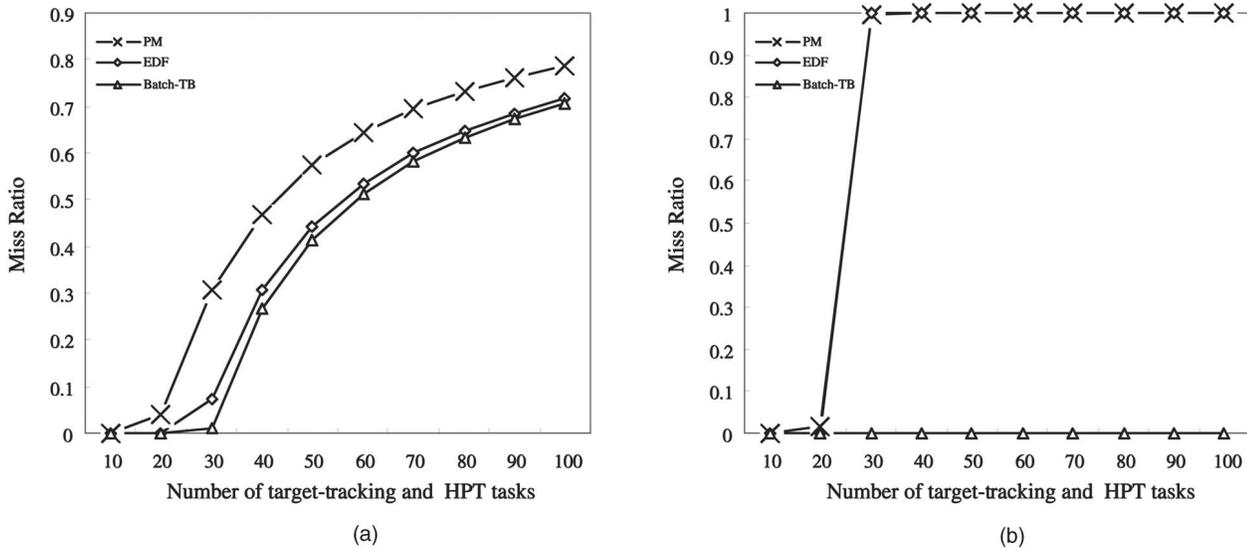


Fig. 8. (a) and (b) are the miss ratio of all target-tracking tasks (including NT, TC, and PT tasks) and the LS task, without an even search workload distribution, respectively.

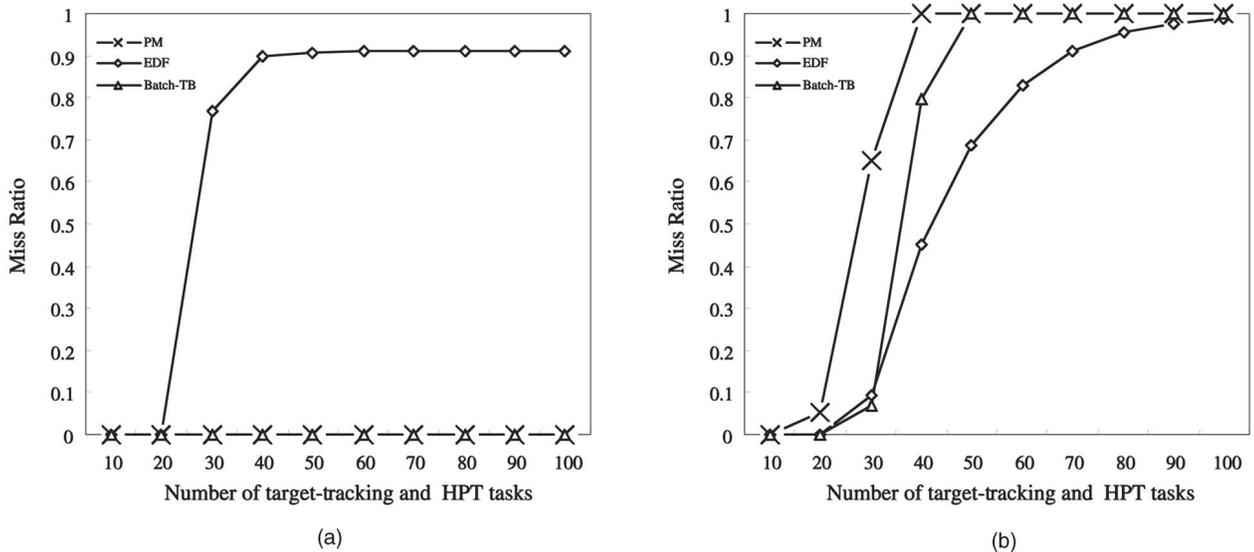


Fig. 9. (a) and (b) are the miss ratio of the HS task and HPT tasks, with an even search workload distribution, respectively.

could schedule over 116 percent of the number of target-tracking tasks than PM did and be over 40 percent better than that under PM for the number of HPT tasks.

5 CONCLUSION

Many existing phased array radar systems still adopt inefficient or even non-real-time resource scheduling mechanisms, such as those based on FIFO-like or cyclic-executive-like scheduling algorithms [2]. As a result, much radar resource is wasted without significant performance improvement. This paper targets the essential issue for the design of modern phased array radar, i.e., real-time dwell scheduling. We not only formalize the typical workload of radar tasks, but also present an effective rate-based approach for real-time dwell scheduling. A Batch-TB-based scheduling algorithm with reservation ratio reclamation is proposed to effectively schedule radar tasks with a good resource

utilization and an even distribution of search workload. The system capacity can also be easily derived. The proposed approach is verified by a series of experiments based on a real phased array radar for air defense frigates [9] for which we have very encouraging results. We show that real-time technology does improve radar performance significantly. In particular, the proposed Batch-TB can greatly improve the performance of a phased array radar by tracking over 116 percent more targets than PM does and, at the same time, improving the capacity of High-Precision Track by over 40 percent, compared to that under PM.

We are now building a phased array radar prototype for the next-generation radar systems in Taiwan. We believe that, with the complexity of the system we are building, only rigorous theory which lends itself to advanced implementation methods can provide us with the reliability and performance acceptable to users. For the future work, we will explore real-time dwell scheduling of a multi-SP

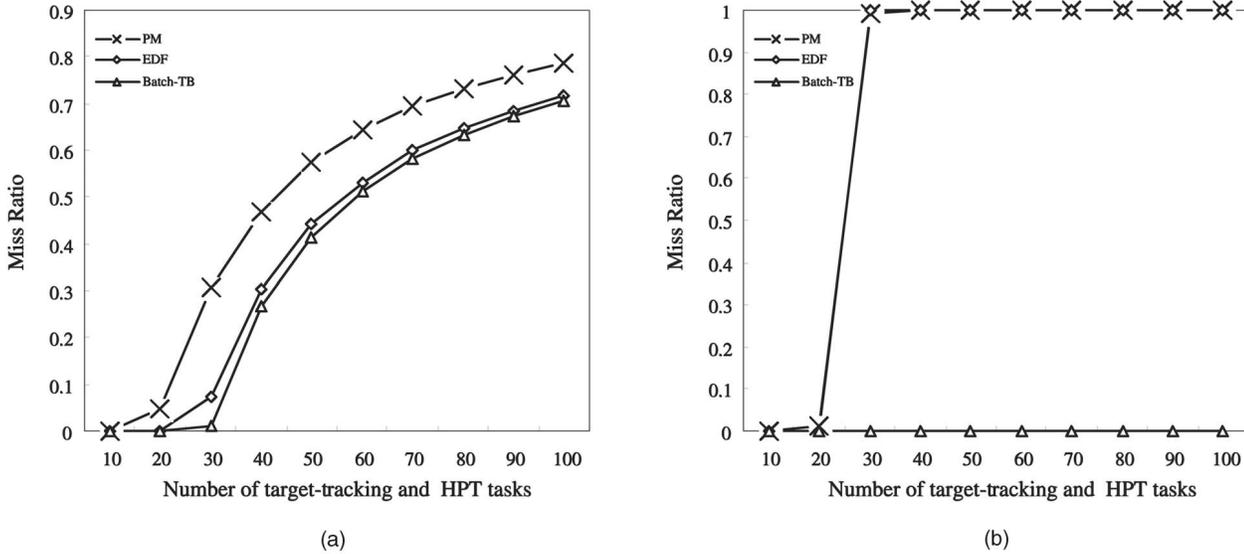


Fig. 10. (a) and (b) are the miss ratio of all target-tracking tasks (including NT, TC, and PT tasks) and the LS task, with an even search workload distribution, respectively.

TABLE 6
Notations and Abbreviations

Symbol/Term	Description	Symbol/Term	Description
HS	High-Priority Search	D_{LS}^i	The relative deadline of an LS task
TC	Track Confirmation	DT_{NT}	The dormant time of an NT task
NT	Normal Track	DT_{PT}	The dormant time of a PT task
PT	Precision Track	DT_{HPT}	The dormant time of an HPT task
HPT	High-Precision Track	P_{HS}^i	The period of an HS task
SI	Scheduling Interval: SI is a time unit for RCC task scheduling	P_{LS}^i	The period of an LS task
B_{HS}^i	The number of beams issued by an HS task in every P_{HS}^i time units	P_{NT}^H	The higher period bound of an NT task
B_{LS}^i	The number of beams issued by an LS task in every P_{LS}^i time units	P_{NT}^L	The lower period bound of an NT task
C_{HS}^i	The maximum dwell length of an HS task	P_{PT}^H	The higher period bound of a PT task
C_{LS}^i	The maximum dwell length of an LS task	P_{PT}^L	The lower period bound of a PT task
C_{TC}	The maximum dwell length of a TC task	P_{HPT}^H	The higher period bound of an HPT task
C_{NT}	The maximum dwell length of an NT task	P_{HPT}^L	The lower period bound of an HPT task
C_{PT}	The maximum dwell length of a PT task	τ_{HS}^i	A High-Priority Search task
C_{HPT}	The maximum dwell length of an HPT task	τ_{LS}^i	A Low-Priority Search task
D_{HS}^i	The relative deadline of an HS task	τ_{TC}^i	A target-tracking task
D_{TC}	The relative deadline of a TC task	θ_{HS}^i	The reservation ratio of an HS task

radar system and tune up our methodology for radar systems of different scales and goals. We will propose integrated scheduling methodologies at both the RCC and SP levels. A distributed radar system deployed in a large region is also under investigation.

APPENDIX

Notations and abbreviations for this paper are in Table 6.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Tai-Chung Wang and Dr. Yu-Lin Su for their help in this work. This research was supported in part by the National Science Council under grants NSC-92-2213-E-002-091 and NSC-92-2213-E-002-093.

This paper is an extended version of a paper that appeared in the *Proceedings of the IEEE 2002 Radar Conference*, April 2002.

REFERENCES

- [1] L. Abeni and G. Buttazzo, "Integrating Multimedia Applications in Hard Real-Time Systems," *Proc. IEEE 19th Real-Time Systems Symp.*, Dec. 1998.
- [2] R.A. Baugh, *Computer Control of Modern Radars*. RCAM&SR-Moorestown Library, 1973.
- [3] T.P. Baker, "A Stack-Based Resource Allocation Policy for Real Time Processes," *Proc. IEEE 11th Real-Time Systems Symp.*, Dec. 1990.
- [4] A. Barbato and P. Giustiniani, "An Improved Scheduling Algorithm for a Naval Phased Array Radar," ALENIA Defence Systems, Italy, 1992.
- [5] C. Chang and T.-C. Wang, "Use Object-Oriented Paradigm to Design a Programmable Radar Digital Signal Processor," *Proc. Third Workshop Object-Oriented Technology and Applications*, Sept. 1997.

- [6] Z. Deng, J.W.-S. Liu, and J. Sun, "A Scheme for Scheduling Hard Real-Time Applications in Open System Environment," *Proc. Ninth Euromicro Workshop Real-Time Systems*, pp. 191-199, June 1997.
- [7] Z. Deng and J.W.-S. Liu, "Scheduling Real-Time Applications in an Open Environment," *Proc. IEEE 18th Real-Time Systems Symp.*, Dec. 1997.
- [8] R. Filippi and S. Pardini, "An Example of Resources Management in a Multifunctional Rotating Phased Array Radar," *Proc. Real-Time Management of Adaptive Radar Systems, IEE Colloquium*, pp. 2/1-2/3, 1990.
- [9] A.G. Huizing and A.A. F. Bloemen, "An Efficient Scheduling Algorithm for a Multifunction Radar," *Proc. IEEE Int'l Radar Conf.*, pp. 359-364, 1996.
- [10] C.W. Hsueh and K.J. Lin, "An Optimal Pinwheel Scheduler Using the Single-Number Reduction Technique," *Proc. 17th IEEE Real-Time Systems Symp.*, pp. 196-205, Dec. 1996.
- [11] C.W. Hsueh and K.J. Lin, "On-Line Schedulers for Pinwheel Tasks Using the Time-Driven Approach," *Proc. 10th Euromicro on Real-Time Systems*, pp. 180-187, June 1998.
- [12] A. Izquierdo-Fuente and J.R. Casar-Corredera, "Approach to Multifunction Radar Scheduling Simulation," *Proc. IEEE Telesystems Conf.*, pp. 67-70, 1994.
- [13] A. Izquierdo-Fuente and J.R. Casar-Corredera, "Optimal Radar Pulse Scheduling Using a Neural Network," *Proc. IEEE Int'l Conf. Neural Networks*, vol. 7, pp. 4588-4591, 1994.
- [14] K. Jeffay and S. Goddard, "A Theory of Rate-Based Execution," *Proc. IEEE 20th Real-Time Systems Symp.*, pp. 304-314, Dec. 1999.
- [15] J.W. Layland, *Real-Time System*. Prentice Hall, 2000.
- [16] G.V. Keuk and S.S. Blackman, "On Phased-Array Radar Tracking and Parameter Control," *IEEE Trans. Aerospace and Electronic Systems*, vol. 8, pp. 186-194, Jan. 1993.
- [17] T.-W. Kuo and C.-H. Li, "A Fixed-Priority-Driven Open Environment for Real-Time Applications," *Proc. IEEE 20th Real-Time Systems Symp.*, Dec. 1999.
- [18] D.-I. Kang, R. Gerber, and M. Saksena, "Performance-Based Design of Distributed Real-Time Systems," *Proc. IEEE 1997 Real-Time Technology and Applications Symp.*, pp. 2-13, June 1997.
- [19] T.-W. Kuo, W.-R. Yang, and K.J. Lin, "EGPS: A Class of Real-Time Scheduling Algorithms Based on Processor Sharing," *Proc. 10th Euromicro Workshop Real-Time Systems*, pp. 27-34, June 1998.
- [20] T.-W. Kuo, C.-F. Kuo, and C. Chang, "Real-Time Digital Signal Processing of Component-Oriented Phased Array Radars," *Proc. IEEE 21st Real-Time Systems Symp.*, 2000.
- [21] C.L. Liu and J.W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *J. ACM*, vol. 20, no. 1, pp. 46-61, Jan. 1973.
- [22] A.K. Mok, "Fundamental Design Problems for the Hard Real-Time Environment," PhD dissertation, Massachusetts Inst. of Technology, Cambridge, 1983.
- [23] R.L. Nevin and F.W. Schatz, "AN/APG-67 Multimode Radar Development," *Proc. IEEE Int'l Radar Conf.*, pp. 1-8, 1985.
- [24] A.K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case," *Proc. IEEE INFOCOM*, 1992.
- [25] "Rapid Prototyping of Application Specific Signal Processors (RASSP)," <http://eto.sysplan.com/eto/rassp>, 2000.
- [26] M. Spuri and G. Buttazzo, "Scheduling Aperiodic Tasks in Dynamic Priority Systems," *J. Real-Time Systems*, vol. 10, no. 2, pp. 179-210, Mar. 1996.
- [27] D. Stromberg and P. Grahn, "Scheduling of Tasks in Phased Array Radar," *Proc. IEEE Int'l Radar Conf.*, pp. 318-321, 1996.
- [28] L. Sha, R. Rajkumar, and J.P. Lehoczky, "Priority Inheritance Protocols: An Approach to Real-Time Synchronization," *IEEE Trans. Computers*, vol. 39, no. 9, Sept. 1990.
- [29] I. Stoica, H. Abdel-Wahab, K. Jeffay, and S.K. Baruah, "A Proportional Share Resource Allocation Algorithm for Real-Time, Time-Shared Systems," *Proc. IEEE 17th Real-Time Systems Symp.*, pp. 288-299, 1996.
- [30] J. Tuner, "New Directions in Communications (or Which Way to the Information Age?)," *IEEE Comm. Magazine*, vol. 24, no. 10, pp. 8-15, Oct. 1986.
- [31] C.A. Waldspurger and W.E. Wehl, "Stride Scheduling Deterministic Proportional Share Resource Management," Technical Memorandum, MIT/LCS/TM-528, Laboratory for CS, Massachusetts Inst. of Technology, July 1995.

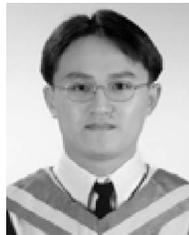
- [32] C.A. Waldspurger, "Lottery and Stride Scheduling: Flexible Proportional-Share Resource Management," PhD thesis, Technical Report, MIT/LCS/TR-667, Laboratory for CS, Massachusetts Inst. of Technology, Sept. 1995.
- [33] T.-C. Wang, "A Programmable Parallel Radar Signal Processor—PRSP," *Proc. 1996 Workshop Distributed System Technologies and Applications*, pp. 125-129, May 1996.



Tei-Wei Kuo received the BSE degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, Republic of China (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 (RTAS), 2001, and has been 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. He is a senior member of the IEEE and the IEEE Computer Society and has received several awards for his research achievements, including the Distinguished Research Award from the ROC National Science Council in 2003.



Yung-Sheng Chao received the BS degree in applied information and computer engineering from Chung Yuan Christian University in Chung-Li, Taiwan, in 1998. He received the MS degree in computer science and information engineering from National Chung Cheng University in Chiayi, Taiwan, in 2000. He is currently an assistant researcher in the Electronic System Research Division of the Chung Shan Institute of Science and Technology, TaoYuan, Taiwan, Republic of China. His research interests include real-time systems, software engineering, and project management.



Chin-Fu Kuo received the BS and MS degrees from the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan, Republic of China (ROC), in 1998 and 2000, respectively. He is currently a PhD student in the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, ROC. His research interests include real-time process scheduling and system security.



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 the 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, Republic of China (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.