하드 데드라인을 가지는 다중 실시간 주기적 태스크에서의 체크포인팅 기법

論 文
53D-8-7

Checkpoint Placement for Multiple Real-time Periodic Tasks with Hard Deadlines

郭 成 祐*
(Seong Woo Kwak)

Abstract - We analyze checkpoint strategy for multiple real-time periodic tasks with hard deadlines. Real-time tasks usually have deadlines associated with them. For multiple real-time tasks, checkpoint strategy considering deadlines of all tasks is very difficult to derive. We analyze the problem of checkpoint placement for such multiple periodic tasks. In our strategy, the interval between checkpoints is determined for each task considering its deadline. An approximated failure probability over a specified interval is derived. Then the number of checkpoints for each task is selected to minimize the approximated failure probability. To show the usefulness of our strategy, error bound between the exact and the approximated failure probability is estimated, which is revealed to be quite small.

Key Words : Checkpoint placement, Real-time system, Hard deadline, Multiple tasks

1. Introduction

Real-time computer systems are often used in harsh environments, such as aerospace and industry. Such systems are subject to many transient faults while in operation [1,2,3,4]. Checkpoint enables a reduction in the recovery time from a transient fault by saving intermediate states of a task in a reliable storage facility, and then, on detection of a fault, restoring from a previously stored state [5]. The interval between checkpoints affects the total execution time of the task: Whereas inserting more checkpoints and reducing the interval between them reduces the reprocessing time after faults, checkpoints have associated execution costs, and inserting extra checkpoints increases the overall task execution time. Thus, there is a trade-off between the reprocessing time and the checkpoint overhead. Though some researchers have investigated the problem of checkpoint placement for a single real-time task [6,7,8], there are few researches on checkpoint placement for multiple real-time tasks due to the complexity in considering multiple tasks. In this paper, we explore this problem.

From our previous research, we found that the optimal checkpoint interval depends on the execution time of task (e), the slack time (available time for rollback recovery), the checkpoint overhead (t_{cp}), and the occurrence and recovery rate of fault (λ, μ) for a single task [9,10]. Hence, given t_{cp}, λ, μ, and e, the optimal checkpoint interval is determined by the slack time. This can be applied to multiple tasks. That is, the optimal checkpoint interval for each task is related to the slack time corresponding to the task. This implies that the checkpoint interval should be selected separately for each task. However, the slack times are dependent on the scheduling strategy adopted by the system, and may not be determined uniquely. For the uniqueness of slack times and mathematical tractability, we restrict our problem as follows.

A1. Tasks are scheduled by the Rate Monotonic (RM) scheduling.

A2.Period of each task is in harmonics of a basic period. That is, $D_i \in \{T, 2T, \cdots, 2^M T\}$, where D_i is the period of task i, T is the basic period, and $M \geq 1$ is an integer.

A3. Deadline of each task is equal to its period.

The RM scheduling is the optimal static priority uni-processor scheduling algorithm and is very popular [3]. In the example shown in Figure 1, where A1, A2, and A3 are satisfied, slack times are determined uniquely as $S_1 = T - e_1$, $S_2 = 2T - 2e_1 - e_2$. Here, the quantities e_1 and e_2 represent execution time of task *1* and task *2* respectively. The problem that will be considered in this

* 正 會 員 : 啓明大 工大 電子工學科 專任講師 · 工博

接受日字 : 2004年 6月 2日
最終完了 : 2004年 7月 24日

paper can be stated as "how to select a checkpoint interval for each task that is characterized by A1, A2, and A3 to minimize the failure probability over a specified interval".

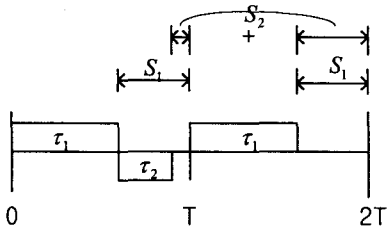


Fig. 1 Task schedule with the RM algorithm

2. Checkpoint Placement for Multiple Periodic Task

Since the period of each task is one of the harmonics of a basic period (Assumption A2), the interval of the largest period includes all information about scheduling of each task. For an example of two tasks with period T and 2T, the maximum period is 2T, and the scheduling pattern within 2T is repeated afterwards as in Figure 1. In our strategy for multiple tasks, checkpoints are placed for each task, that is, the interval between checkpoints is Δ_1 for task 1, and Δ_2 for task 2, and so on.

To find the checkpoint interval for each task, failure probability over the interval of the largest period will be considered. Task i will be denoted as τ_i. The index i for task τ_i is assigned in the increasing order of its period. We define $p(n_1, n_2, \cdots, n_r)$ as the probability of 1-p{all tasks complete their executions successfully within their deadlines over the interval of the largest period when n_i checkpoints are placed in task i $(1 \le i \le r)$}. Our goal is to find the set $\{n_1^*, n_2^*, \cdots, n_r^*\}$ that minimizes $p(n_1, n_2, \cdots, n_r)$ for total r tasks.

Several additional assumptions for our analysis are as follows.

A4. Faults occur according to Poisson process with rate λ and recover with rate μ.

A5. The occurrence of a fault always causes errors.

A6. Occurrence rate λ is sufficiently small to neglect the probability of more than one fault within the interval of the largest period.

Assumption A4 is common in many analyses [2,5,7,11,12]. Assumption 5 is conservative: Fault does not necessarily cause errors [2,9,10]. For real systems, it is usually true that the occurrence rate of a fault is small. Thus assumption A6 is reasonable.

3. Derivation of Failure Probability

3.1 Slack Time

In the RM scheduling, task priorities are assigned inversely related to their periods (or deadlines). Let S_i be the slack time for τ_i, which is represented by the maximum available time for τ_i while all task executions including τ_i are guaranteed within their periods. We can see two slack times(S_1, S_2) for τ_1 and τ_2 in the example of Figure 1. As mentioned in the previous section, since the interval of the largest period contains all the information of scheduling, it is sufficient to consider the task scheduling during the largest interval to find the slack time for each task. Figure 2 shows how to find S_i. Here, I_i represents the amount of time intruded by executions of lower priority tasks into the period of τ_i when all the lower priority tasks than τ_i are shifted to the right until their periods are reached. S_{M_i} is the slack time for τ_i when $I_i = 0$. Figure 3 (a) and (b) show one simple example to determine S_i, I_i, and S_{M_i} for two tasks with periods of T and 2T. There exist D_r / D_i periods for τ_i within the maximum period D_r. For the calculation of S_i, I_i and S_{M_i}, it is sufficient to consider the first period because S_i, I_i and S_{M_i} are constant for all periods.

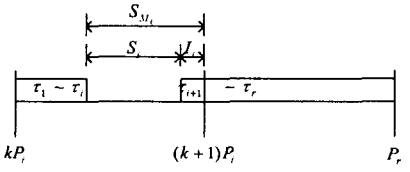


Fig. 2 S_i, I_i, and S_{M_i}

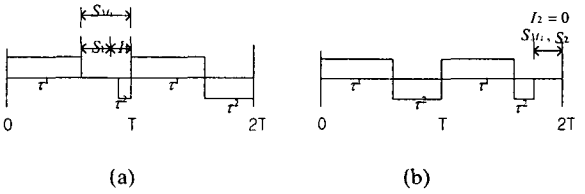


Fig. 3 An example to calculate S_i, I_i and S_{M_i}

From the definitions of S_i, I_i and S_{M_i}, the following expression holds

$$S_i = S_{M_i} - I_i \tag{1}$$

Since task periods and scheduling are characterized by the assumptions A2 and A1, the quantity S_{M_i} can be derived as follows.

$$S_{M_i} = D_i - \left(\frac{D_i}{D_1}\right)e_1 - \left(\frac{D_i}{D_2}\right)e_2 \cdots - \left(\frac{D_i}{D_i}\right)e_i \tag{2}$$

where e_i is execution time including checkpoint overhead(t_{cp}), and D_i is deadline(=period) of τ_i.

By inspecting the task scheduling carefully, we can find the I_i recursively as follows.

$$I_r = 0$$

$$I_i = \max\left\{\left(\frac{D_{i+1}-D_1}{D_1}\right)e_1 + \left(\frac{D_{i+1}-D_2}{D_2}\right)e_2 + \cdots + \left(\frac{D_{i+1}-D_i}{D_{i+1}}\right)e_{i+1} + I_{i+1} - (D_{i+1}-D_i),\ 0\right\} \tag{3}$$

From the relation among S_i, I_i and S_{M_i}, slack time for τ_i is derived as:

$$S_i = D_i - \left(\frac{D_i}{D_1}\right)e_1 - \left(\frac{D_i}{D_2}\right)e_2 - \cdots - \left(\frac{D_i}{D_i}\right)e_i - I_i \tag{4}$$

Note that $S_i \geq S_{i-1} \geq \cdots \geq S_2 \geq S_1$.

3.1 Approximated Failure Probability

It is very difficult to derive the *exact* failure probability for $r \geq 3$. Even for a simple example of two tasks, the exact equation requires very complex calculations. However, in an environment where assumption A6 holds, we can derive the failure probability approximately with a reasonable computational complexity. A system is called failed when any one among r tasks is not executed successfully within its deadline. We define $\phi_{ij}(\Delta)$ $(i, j \in \{0,1\})$ as the probability of fault lying in state j at $t = \Delta$ given the initial state i at $t = 0$, and $\phi_0(\Delta)$ as the probability of no fault within an interval $[0, \Delta]$. Here, state *0* and *1* represent fault-free and fault-active, respectively. Figure 4 describes the probabilities of $\phi_{ij}(\Delta)$ $(i, j \in \{0,1\})$ graphically.

From the assumption A4, we can find probabilities $\phi_{ij}(\Delta)$ and $\phi_0(\Delta)$ as follows [2,10].

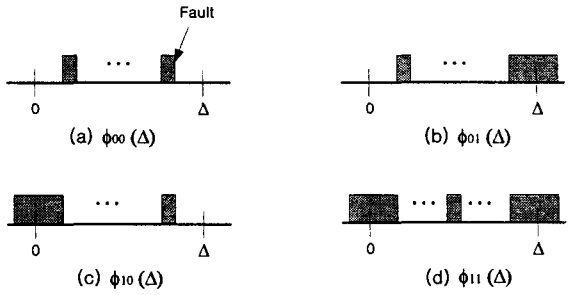


Fig. 4. Graphical description of $\phi_{ij}(\Delta)$

$$\phi_{00}(\Delta) = \frac{\lambda}{\mu+\lambda}e^{-(\mu+\lambda)\Delta} + \frac{\mu}{\mu+\lambda}, \tag{5}$$

$$\phi_{01}(\Delta) = -\frac{\lambda}{\mu+\lambda}e^{-(\mu+\lambda)\Delta} + \frac{\lambda}{\mu+\lambda}, \tag{6}$$

$$\phi_{10}(\Delta) = -\frac{\mu}{\mu+\lambda}e^{-(\mu+\lambda)\Delta} + \frac{\mu}{\mu+\lambda}, \tag{7}$$

$$\phi_{11}(\Delta) = \frac{\mu}{\mu+\lambda}e^{-(\mu+\lambda)\Delta} + \frac{\lambda}{\mu+\lambda}, \tag{8}$$

$$\phi_0(\Delta) = e^{-\lambda\Delta}. \tag{9}$$

Let V_x be the transition probability matrix of an interval x containing no fault, and U_x be the transition matrix containing faults. V_x and U_x can be derived by

$$V_x = \begin{bmatrix} \phi_0(x) & 0 \\ 0 & 0 \end{bmatrix}, \quad U_x = \begin{bmatrix} \phi_{00}(x)-\phi_0(x) & \phi_{01}(x) \\ \phi_{10}(x) & \phi_{11}(x) \end{bmatrix}. \tag{10}$$

Since the probability of more than one fault within the interval of the largest period is sufficiently small, we can neglect the probability of multiple faults in deriving the failure probability. Let Q_i be a transition probability matrix of one period interval for τ_i where τ_i is successfully executed within its deadline even though a fault may occur during the execution of τ_i. For $i=1$, that is the highest priority task (the shortest period), one period interval with n_1 checkpoints can be sketched as in Figure 5. Here, the shaded region represents the extra time needed for each checkpoint, that is the checkpoint overhead. Δ_1 is one checkpoint interval for τ_1 including the checkpoint overhead.

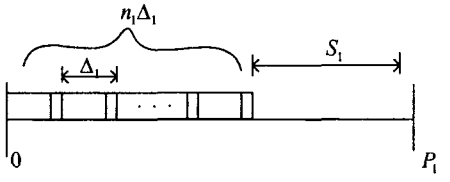


Fig. 5. One period interval for task τ_1

Considering the slack time and the checkpoint interval in Figure 5, we can find

$$Q_1 = \sum_{k=0}^{n_1-1} \sum_{j=1}^{\left\lfloor \frac{S_1}{\Delta_1} \right\rfloor} (V_{\Delta_1})^k (U_{\Delta_1})^j (V_{\Delta_1})^{n_1-k} \tag{11}$$

For τ_i $(i>1)$, the higher priority task $\tau_j (j<i)$ occasionally preempts τ_i. Execution of τ_i begins at the end of τ_{i-1}. Figure 6 shows one period interval and the slack time for τ_i. Note that at first, only the highest priority task τ_1 preempts the execution of τ_i due to the assumption A2, and at the end of τ_1 execution the next highest priority task waiting is executed, and so on until all tasks higher than τ_i are executed. Oblique regions represent the preemption of τ_1 and other higher tasks waiting for execution.

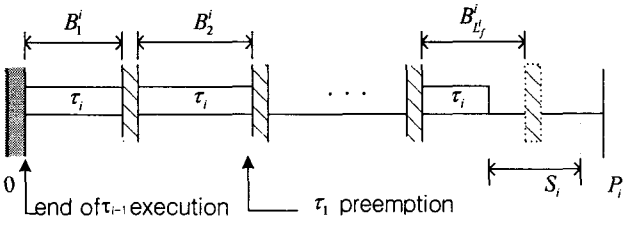
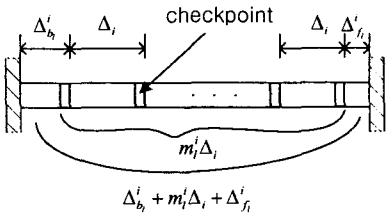


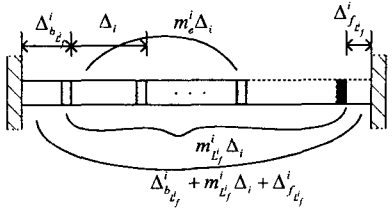
Fig. 6. One period interval for τ_i, $i>1$

One period interval for τ_i is divided into $L^i{}_f$ blocks. Figure 7 (a) and (b) show detailed depiction of l-th and the final block, respectively. The l-th block is denoted by B_l^i.

For each block B_l^i $(l < L_f^i)$, there are one beginning incomplete checkpoint interval $\Delta_{b_l}^i$, m_l^i complete checkpoint intervals of length Δ_i, and the final incomplete checkpoint interval $\Delta_{f_l}^i$ as shown in Figure 7 (a). The incomplete checkpoint intervals are made by the preemption of τ_1 and other higher tasks during the execution of τ_i. Thus both the final incomplete interval of block B_{l-1}^i and the beginning incomplete interval of block B_l^i make one complete checkpoint interval, that is $\Delta_i = \Delta_{b_l}^i + \Delta_{f_{l-1}}^i$.



(a) B_l^i



(b) $B_{L_f^i}^i$

Fig. 7. l-th and the final block

The final block $B_{L_f^i}^i$ has $m_{L_f^i}^i$ checkpoint intervals of length Δ_i and one beginning incomplete interval of $\Delta_{b_{L_f^i}}^i$. For the final block, the execution of τ_i finishes when m_e^i complete intervals among $m_{L_f^i}^i$ intervals are executed successfully in the block $B_{L_f^i}^i$ as shown in Figure 7 (b). Define $C_l^i = \sum_{j=1}^{l} B_j^i$ for $l \geq 1$, and $C_0^i = 0$. The length of each block (B_l^i), and the number of blocks (L_f^i) for τ_i can be derived as follows.

$$B_l^i = K_l^i \cdot b - \{A_{i-1}(K_l^i) + \sum_{d=1}^{l-1} B_d^i\} \tag{12}$$

$$L_f^i = \min_{C_l^i \geq e_i} \{l\} \tag{13}$$

where,

$b = D_1 - e_1$,

$$A_{i-1}(K_l^i) = \left\lceil \frac{K_l^i D_1}{D_2} \right\rceil e_2 + \left\lceil \frac{K_l^i D_1}{D_3} \right\rceil e_3 + \cdots + \left\lceil \frac{K_l^i D_1}{D_{i-1}} \right\rceil e_{i-1},$$

$$K_l^i = \begin{cases} \min\limits_{k \cdot b > A_{i-1}(k)} \{k\} & if \ \ l=1 \\ K_{l-1}^i + h_{l-1}^i & otherwise \end{cases}$$

$$h_l^i{}_1 = \min_{(K_{l-1}^i + h) b > A_{i-1}(K_{l-1}^i + h) + \sum_{d=1}^{l-1} B_d^i} \{h\}$$

where

Here, K_l^i, h_{l-1}^i are positive integers. K_1^i indicates the number of periods with respect to τ_1 where τ_i finishes its first period execution.

Parameters for each block shown in Figure 6 can be derived as follows.

$$\Delta_i = \frac{e_i}{n_i} = \frac{\bar{e}_i}{n_i} + t_{cp} \tag{14}$$

$$\Delta_{b_l}^i = \Delta_i - \left(C_{l-1}^i - \Delta_i \left\lfloor \frac{C_{l-1}^i}{\Delta_i} \right\rfloor \right), \tag{15}$$

$$\Delta_{f_l}^i = \begin{cases} C_l^i - \Delta_i \left\lfloor \frac{C_l^i}{\Delta_i} \right\rfloor & \text{if } C_l^i - \Delta_i \left\lfloor \frac{C_l^i}{\Delta_i} \right\rfloor \neq 0 \\ \Delta_i & \text{if } C_l^i - \Delta_i \left\lfloor \frac{C_l^i}{\Delta_i} \right\rfloor = 0 \end{cases}, \tag{16}$$

$$m_l^i = \frac{B_l^i - \Delta_{b_l}^i - \Delta_{f_l}^i}{\Delta_i}, \quad m_{L_f^i}^i = \left\lfloor \frac{B_{L_f^i}^i - \Delta_{b_{L_f^i}}^i}{\Delta_i} \right\rfloor,$$

$$m_e^i = n_i - \left\lfloor \frac{C_{L_f^i-1}^i + \Delta_{b_{L_f^i}}^i}{\Delta_i} \right\rfloor. \tag{17}$$

Here, n_i represents the number of checkpoints in τ_i, t_{cp} is the checkpoint overhead, and $\bar{e}_i$ is the execution of τ_i before n_i inserting checkpoints.

Let $'\alpha','\beta' (\in\{0,1\})$ represent corruption state of an interval. An interval is called in corruption state '*1*' when there is any fault within the interval, and called in corruption state '*0*' when there is no fault. Define *l*-th block transition probability matrices (TPM) $\gamma_{(\alpha,\beta)}^l(\Delta_{b_l}^i, \Delta_{f_l}^i, \Delta_i, m_l^i, m_{m_l}^i)$, $\gamma_\alpha^f(\Delta_{b_{L_f^i}}^i, \Delta_i, m_{L_f^i}^i, m_e^i)$, and $\gamma_1^l(B_l^i, \Delta_i, m_l^i)$ as follows.

$\gamma_{(\alpha,\beta)}^l(\Delta_{b_l}^i, \Delta_{f_l}^i, \Delta_i, m_l^i, m_{m_l}^i)$ $(for\ m_l^i \geq 0\ or\ \Delta_i < B_l^i) \equiv$ TPM{A transient fault occurs at the *l*-th block of τ_i with parameters of $\Delta_{b_l}^i, \Delta_{f_l}^i, \Delta_i$ and m_l^i, but τ_i is executed successfully within its period with no fault in other blocks, where at least $m_l^i - m_{m_l}^i$ complete checkpoint intervals (which will also be called time-slots) are corrupted by the fault, the number of corrupted time-slots in τ_i and τ_1 by the fault is within the bound of not causing a system failure, and the beginning interval of $\Delta_{b_l}^i$ is in corruption state 'α' and the final interval of $\Delta_{f_l}^i$ is in corruption state 'β'}

$\gamma_\alpha^f(\Delta_{b_{L_f^i}}^i, \Delta_i, m_{L_f^i}^i, m_e^i) \equiv$ TPM{A transient fault occurs at the final block of τ_i with parameters of $\Delta_{b_{L_f^i}}^i, \Delta_i, m_{L_f^i}^i$ and m_e^i, but τ_i is executed successfully within its period and finished at the final block with no fault in other blocks, where the number of corrupted time-slots in τ_i by the fault is within the bound of not causing a system failure, and the beginning interval of $\Delta_{b_{L_f^i}}^i$ is in corruption state 'α'}

$\gamma_1^l(B_l^i, \Delta_i, m_l^i)$ $(for\ m_l^i < 0\ or\ \Delta_i \geq B_l^i) \equiv$ TPM{A transient fault occurs at the *l*-th block of τ_i characterized by $B_l^i \leq \Delta_i$, but τ_i is executed successfully within its period with no fault in other blocks, where the number of corrupted time-slots in τ_i and τ_1 by the fault is within the bound of not causing a system failure}

Note that when a fault occurs at the *l*-th block of τ_i and continues until the preemption of τ_1 as shown in Figure 8, execution of τ_1 as well as execution of τ_i is affected by the fault.

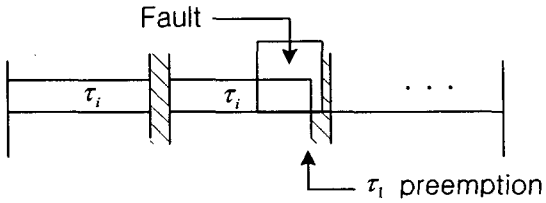


Fig. 8. Continuation of a fault until τ_1 preemption

$\gamma_{(\alpha,\beta)}^l(\Delta_{b_l}^i, \Delta_{f_l}^i, \Delta_i, m_l^i, m_{m_l}^i)$ are derived as follows.

$$\gamma_{(0,0)}^l(\Delta_{b_l}^i, \Delta_{f_l}^i, \Delta_i, m_l^i, m_{m_l}^i) = \sum_{k_{11}=0}^{m_{m_l}^i} \sum_{k_{12}=\max\left\{\left\lfloor \frac{(m_l^i - k_{11})\Delta_i - S_1}{\lambda_1} \right\rfloor, 0\right\}}^{m_{m_l}^i - k_{11}} V_{\lambda_1}^{k_{11}} \cdot U_{\lambda_1}^{m_l^i - (k_{11}+k_{12})} \cdot V_{\lambda_1}^{k_{12}} \cdot V_{\lambda_1}^{n_1 - (k_{11}+k_{12})} \tag{18}$$

$$\gamma_{(0,1)}^l(\Delta_{b_l}^i, \Delta_{f_l}^i, \Delta_i, m_l^i, m_{m_l}^i) = \sum_{k=0}^{m_{m_l}^i} \sum_{j=0}^{\min\left\{\left\lfloor \frac{S_1}{\lambda_1} \right\rfloor, \left\lfloor \frac{S_1 - (m_l^i - k+1)\lambda_1}{\lambda_1} \right\rfloor\right\}} V_{\lambda_1}^k \cdot U_{\lambda_1}^{m_l^i - k} \cdot U_{\lambda_{f_l}} \cdot U_{\lambda_1}^j \cdot V_{\lambda_1 - \Delta_{f_l}} \cdot V_{\lambda_1}^{n_1 - k} \tag{19}$$

$$\gamma_{(1,0)}^l(\Delta_{b_l}^i, \Delta_{f_l}^i, \Delta_i, m_l^i, m_{m_l}^i) = \sum_{k=\max\left\{\left\lfloor \frac{(m_l^i+1)\Delta_i - S_1}{\Delta_i} \right\rfloor, 0\right\}}^{m_{m_l}^i} U_{\Delta_{f_l}} \cdot U_{\Delta_i}^{m_l^i - k} \cdot V_{\Delta_i}^k \cdot V_{\Delta_i - \Delta_{f_l}} \cdot V_{\Delta_i}^{n_i - k} \tag{20}$$

$$\gamma^{l}_{(1,1)}(\Delta^{i}_{b_l},\Delta^{i}_{f_l},\Delta_i,m^{i}_{l},m^{i}_{m_l}) = \sum_{j=0}^{\min\left\{\left\lfloor\frac{S_l}{\Delta_l}\right\rfloor,\left\lfloor\frac{S_l-(m^{i}_{l}-k+1)\Delta_l}{\Delta_l}\right\rfloor\right\}} U_{\Delta_{b_l}}\cdot U^{m^{i}_{l}}_{\Delta_l}\cdot U_{\Delta_{f_l}}\cdot U^{j}_{\Delta_l}\cdot V_{\Delta_l-\Delta_{b_l}}\cdot V_{\Delta_l-\Delta_{f_l}}\cdot V^{n_i}_{\Delta_l} \quad (21)$$

In $\gamma^{l}_{(0,0)}(\cdot)$, the term $V^{k_{l1}}_{\Delta_i}\cdot U^{m^{i}_{l}-(k_{l1}+k_{l2})}_{\Delta_i}\cdot V^{k_{l2}}_{\Delta_i}$ is transition probability matrix of l-th block where $m^{i}_{l}-(k_{l1}+k_{l2})$ among m^{i}_{l} time-slots are corrupted by a transient fault, and the term $V^{n_i-(k_{l1}+k_{l2})}_{\Delta_i}$ is the transition probability matrix of no fault in $n_i-(k_{l1}+k_{l2})$ time-slots of the other blocks of τ_i. Note that the number of V_{Δ_i} in $\gamma^{l}_{(0,0)}(\cdot)$ is n_i, which represents the successful execution of τ_i. The constraint on k_{l2}, that is $\max\left\{\left\lceil\frac{(m^{i}_{l}-k_{l1})\Delta_i-S_l}{\Delta_i}\right\rceil,0\right\}\le k_{l2}\le m^{i}_{m_l}-k_{l1}$, is to ensure the whole execution of τ_i within its period. In case of $\gamma^{l}_{(0,1)}(\cdot)$, a fault may continue until the preemption of τ_1, and corrupt task τ_1. The transition probability matrix $U_{\Delta_1})^{j}$ included in the equation $\gamma^{l}_{(0,1)}(\cdot)$ represents this. Other transition probabilities can be obtained similarly [10].

$\gamma^{f}_{\alpha}(\Delta^{i}_{b_{L^{i}_{f}}},\Delta_i,m^{i}_{L^{i}_{f}},m^{i}_{e})$ and $\gamma^{l}_{1}(B^{i}_{l},\Delta_i,m^{i}_{l})$ are as follows.

$$\gamma^{f}_{0}(\Delta^{i}_{b_{L^{i}_{f}}},\Delta_i,m^{i}_{L^{i}_{f}},m^{i}_{e}) = \sum_{k=0}^{m^{i}_{e}-1}\sum_{j=1}^{\min\left\{m^{i}_{L^{i}_{f}}-m^{i}_{e},\left\lfloor\frac{S_l}{\Delta_i}\right\rfloor\right\}} V^{k}_{\Delta_i}\cdot U^{j}_{\Delta_i}\cdot V^{m^{i}_{e}-k}_{\Delta_i}\cdot V^{n_i-m^{i}_{e}}_{\Delta_i} \quad (22)$$

$$\gamma^{f}_{1}(\Delta^{i}_{b_{L^{i}_{f}}},\Delta_i,m^{i}_{L^{i}_{f}},m^{i}_{e}) = \sum_{j=0}^{\min\left\{m^{i}_{L^{i}_{f}}-(m^{i}_{e}+1),\left\lfloor\frac{S_l-\Delta_i}{\Delta_i}\right\rfloor\right\}} U_{\Delta^{i}_{b_{L^{i}_{f}}}}\cdot U^{j}_{\Delta_i}\cdot V_{\Delta_i-\Delta^{i}_{b_{L^{i}_{f}}}}\cdot V^{n_i}_{\Delta_i} \quad (23)$$

$$\gamma^{l}_{1}(B^{i}_{l},\Delta_i,m^{i}_{l}) = \sum_{j=0}^{\min\left\{\left\lfloor\frac{S_l}{\Delta_l}\right\rfloor,\left\lfloor\frac{S_l-\Delta_i}{\Delta_l}\right\rfloor\right\}} U_{B^{i}_{l}}\cdot U^{j}_{\Delta_1}\cdot V_{\Delta_i-B^{i}_{l}}\cdot V^{n_i}_{\Delta_i} \quad (24)$$

Using the above equations (Eqs. (18) to (24)), Q_i for $i\ge 2$ can be derived as follows.

$$Q_i = \sum_{l=1}^{L^{i}_{f}}\left[\sum_{m^{i}_{l}\ge 0}\sum_{\alpha,\beta\in\{0,1\}}\gamma^{l}_{(\alpha,\beta)}(\Delta^{i}_{b_l},\Delta^{i}_{f_l},\Delta_i,m^{i}_{l},m^{i}_{m_l})+\sum_{m^{i}_{l}<0}\gamma^{f}_{1}(B^{i}_{l},\Delta_i,m^{i}_{l})\right] + \gamma^{f}_{0}(\Delta^{i}_{b_{L^{i}_{f}}},\Delta_i,m^{i}_{L^{i}_{f}},m^{i}_{e}) + \gamma^{f}_{1}(\Delta^{i}_{b_{L^{i}_{f}}},\Delta_i,m^{i}_{L^{i}_{f}},m^{i}_{e}) \quad (25)$$

where

$$m^{i}_{m_l} = \begin{cases} m^{i}_{l}-1 & if\ \ \alpha=\beta=0 \\ 0 & if\ \ \alpha=\beta=1 \\ m^{i}_{l} & otherwise \end{cases},$$

$$m^{i}_{m_{L^{i}_{f}}} = \begin{cases} m^{i}_{e}-1-\delta(\Delta^{i}_{f_{L^{i}_{f}}}-\Delta_i) & if\ \ \alpha=\beta=0 \\ m^{i}_{e}-1 & if\ \ \alpha=0,\ \beta=1 \\ m^{i}_{e}-\delta(\Delta^{r}_{f_{L^{i}_{f}}}-\Delta_i) & if\ \ \alpha=1,\ \beta=0 \\ 0 & if\ \ \alpha=\beta=1 \end{cases},$$

$$\delta(x) = \begin{cases} 1 & if\ \ x=0 \\ 0 & otherwise \end{cases}.$$

Define $q_i(n_1,n_2,\cdots,n_r)$ as the probability that only one fault occurs in task τ_i, and other tasks $\tau_1,\tau_2,\cdots,\tau_r$, which are checkpointed with $n_1,n_2,\cdots,n_r$ respectively, are executed successfully within their periods during the interval of the largest period, then

$$q_1(n_1,n_2,\cdots,n_r) \cong \left(\frac{D_r}{D_1}\right)\cdot E\cdot Q_1\cdot V^{n_1\left(\frac{D_r}{D_1}\right)-1}_{\Delta_1}\cdot V^{n_2\left(\frac{D_r}{D_2}\right)}_{\Delta_2}\cdots\ V^{n_r\left(\frac{D_r}{D_r}\right)}_{\Delta_r}[1\ \ 0]^{T} \quad (26)$$

$$q_i(n_1,n_2,\cdots,n_r) \cong \left(\frac{D_r}{D_i}\right)\cdot E\cdot Q_i\cdot V^{n_1\left(\frac{D_r}{D_1}\right)}_{\Delta_1}\cdots V^{n_{i-1}\left(\frac{D_r}{D_{i-1}}\right)}_{\Delta_{i-1}}\cdot V^{n_i\left(\frac{D_r}{D_i}\right)-1}_{\Delta_i}\cdot V^{n_{i+1}\left(\frac{D_r}{D_{i+1}}\right)}_{\Delta_{i+1}}\cdots\ V^{n_r\left(\frac{D_r}{D_r}\right)}_{\Delta_r}\cdot[1\ \ 0]^{T} \quad (27)$$

where $E=\left[\frac{\mu}{\lambda+\mu}\ \ \frac{\lambda}{\lambda+\mu}\right]$, which is the initial probability at the beginning of the largest period. Define $q(n_1,n_2,\cdots,n_r)$ as follows.

$q(n_1,n_2,\cdots,n_r)\equiv$P{tasks $\tau_1,\tau_2,\cdots,\tau_r$, which are checkpointed with $n_1,n_2,\cdots,n_r$ respectively, are executed successfully within their periods during the interval of the largest period in an environment characterized by the assumption A6}

From Eqs. (26) and (27), $q(n_1,n_2,\cdots,n_r)$ can be obtained as

$$q(n_1,n_2,\cdots,n_r) = \sum_{i=1}^{r}q_i(n_1,n_2,\cdots,n_r) + E\cdot V^{n_1\left(\frac{D_r}{D_1}\right)}_{\Delta_1}\cdot V^{n_2\left(\frac{D_r}{D_2}\right)}_{\Delta_2}\cdots\ V^{n_r\left(\frac{D_r}{D_r}\right)}_{\Delta_r}[1\ \ 0]^{T} \quad (28)$$

The second term in Eq. (28) is the probability of no

fault over the largest period. Note that the failure probability is $p(n_1,n_2,\cdots,n_r)=1-q(n_1,n_2,\cdots,n_r)$.

4. EstimationofErrorBound

Although derivation of the exact failure probabilities is very difficult, the bound of error between the exact failure probability and the approximated failure probability can be estimated easily. From the Poisson distribution and the Taylor's theorem with remainder, the probability of more than one fault in t is given by

$$p(n\geq 2,\lambda t)=\frac{(\lambda t)^2}{2!}e^{-\lambda t}+\frac{(\lambda t)^3}{3!}e^{-\lambda t}+$$
$$p(n\geq 2,\lambda t)=\frac{(\lambda t)^2}{2!}e^{-\lambda t}+\frac{(\lambda t)^3}{3!}e^{-\lambda t}+\cdots\leq\frac{(\lambda t)^2}{2}e^{-\lambda t}e^{\lambda t}=\frac{(\lambda t)^2}{2}. \quad (29)$$

Suppose that the probability of more than one fault is fully reflected in the failure probability, then the following equation holds.

$$(1-q^*(\cdot))-(1-q(\cdot)) = (q(\cdot)-q^*(\cdot))\leq\frac{(\lambda t)^2}{2} \quad (30)$$

where $q^*(\cdot)$ is the exact probability corresponding to $q(\cdot)$. Thus the approximated failure probability well represents the exact failure probability as λt goes small, where t is the largest period.

5. NumericalResults

Consider a simple example of the case of two tasks with $E=[e_1\ e_2]=[0.5\ 0.56]$ and $D=[D_1\ D_2]=[1\ 2]$. We assume $\lambda=0.01$, $\mu=10$, and $t_{cp}=0.01$. Figure 9 shows the probability $q(n_1,n_2)$ according to n_1 (the number of checkpoints in task τ_1) and n_2 (the number of checkpoints in task τ_2). The maximum $q(n_1,n_2)$ is achieved at $(n_1,n_2)=(4,5)$. It means that checkpoint placement in tasks τ_1 and τ_2 with the number of $(n_1,n_2)=(4,5)$ is the best. Figure 10 shows a result for parameters of $e_1=0.52$, $e_2=0.6$, $\lambda=0.001$, $\mu=10$, and other parameters are the same as those in Figure 9. The maximum $q(n_1,n_2)$ is achieved at $(n_1,n_2)=(5,3)$ in this case.

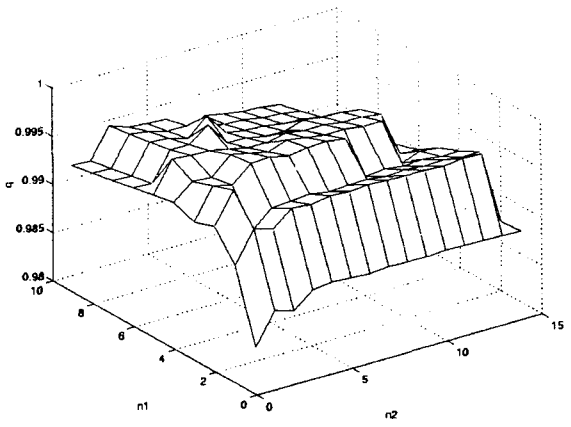


Fig. 9. Plot of $q(n_1,n_2)$
($e_1=0.5$, $e_2=0.56$, $\lambda=0.01$, $\mu=10$, $T=1$, $t_{cp}=0.01$)

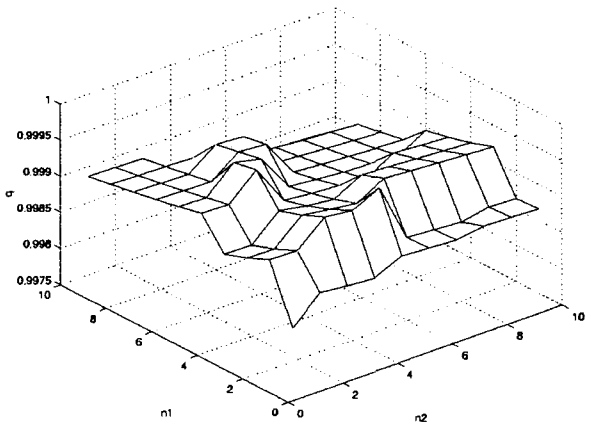


Fig. 10. Plot of $q(n_1,n_2)$
($e_1=0.52$, $e_2=0.6$, $\lambda=0.001$, $\mu=10$, $T=1$, $t_{cp}=0.01$)

6. Conclusion

In this paper, we provided a checkpoint placement strategy for multiple tasks. Our strategy determines checkpoint interval for each task. Assuming that periods of tasks are in a set of harmonics of a basic period and tasks are scheduled by the RM algorithm, we derived an approximated failure probability over the interval of the largest period. The number of checkpoints for each task is selected to minimize the approximated failure probability. Error bound between the exact and the approximated failure probability is estimated to show usefulness of our strategy, which is in reasonable range for small λD_r ($\lambda D_r \ll 1$) where D_r is the largest period. Moreover, the approximated failure probability needs quite a small amount of computational complexity compared with that of the exact probability that is nearly intractable for more than 3 tasks.

References

[1] C. M. Krishna and A. D. Singh, "Optimal configuration of redundant real-time systems in the face of correlated failure," IEEE Trans. on Reliability, vol. 44, pp. 587-594. Dec. 1995.

[2] Seong Woo Kwak and Byung Kook Kim, "Task Scheduling Strategies for Reliable TMR Controller-susing Task Grouping and Assignment", IEEE Tr. Reliability, vol. 49, no.4, pp. 355-362, Dec. 2000.

[3] C. M. Krishina and Kang G. Shin, Real-Time Systems, New York: McGraw-Hill, 1997.

[4] D. P. Siewiorek and R. S. Swarz, Reliable Computer Systems, Digital Press, 1992.

[5] Avi Ziv and Jehoshua Bruck, "An on-line algorithm for checkpoint placement," IEEE Trans. on Computers, vol. 46, pp. 976-984, Sep. 1997.

[6] R. Geist, R. Reynolds, and J. Westall, "Selection of a checkpoint interval in a critical-task environment," IEEE Trans. on Reliability, vol. 37, pp. 395-400, Oct. 1988.

[7] Kang G. Shin, Tein-Hsiang Lin, and Yann-Hang Lee, "Optimal checkpointing of real-time tasks," IEEE Trans. on Computers, vol. C-36, pp. 1328-1341, Nov. 1987.

[8] C. M. Krishna and A. D. Singh, "Reliability of checkpointed real-time systems using time redundancy," IEEE Trans. on Reliability, vol. 42, pp. 427-435, Sep. 1993.

[9] Seong Woo Kwak, Byung Jae Choi and Byung Kook Kim, "Optimal Checkpointing Strategy for Real-Time Control Systems under Faults with Exponential Duration", IEEE Tr. Reliability, vol. 50, no. 3, pp. 293-301, Sep. 2001.

[10] Seong Woo, Kwak, "Reliability Analysis and Design of Real-time Fault Tolerant Control Systems under Transient Faults", Ph.D thesis, KAIST, 2000.

[11] John W. Young, "A first order approximation to the optimal checkpoint intervals," Comm. of the ACM, vol. 17, pp.530-531, Nov. 1974.

[12] Hagbae Kim and Kang G. Shin, "Modeling of externally-induced/common-cause faults in fault-tolerant systems," IEEE/AIAA Digital Avionics System Conference, pp. 402-407, Oct. 1994.

저 자 소 개

곽 성 우 (郭 成 祐)

1970년 3월 10일생. 1993년 한국과학기술원 전기 및 전자공학과 졸업. 2000년 동대학원 전기 및 전자공학과 졸업(공박). 2003년~현재 계명대학교 전자공학과 전임강사.

Tel : 053-580-5926

Fax : 053-580-5165

E-mail : ksw@kmu.ac.kr