The 8th International Workshop on Performability Modeling of Computer and Communication Systems
Edinburgh, Scotland, Sept. 20-21, 2007.

Recurrence-Relation-Based Reward Model for Performability Evaluation of Embedded Systems

Ann T. Tai Kam S. Tso
IA Tech, Inc.
Los Angeles, CA 90094, USA

William H. Sanders
University of Illinois
Urbana, IL 61801, USA

Abstract

Many embedded systems behave as discrete-time semi-Markov processes (DTSMPs). For those systems, performability measures, especially when specified as an accumulated reward, are often difficult to evaluate analytically. In this article, we informally describe an approach that uses a recurrence-relation-based (RRB) reward model for performability evaluation of systems exhibiting DTSMP behavior. We explain how an RRB reward model is constructed and how such a model can be solved analytically.

I. Motivation

Performability evaluation of embedded systems is important because 1) their performance is often gracefully degradable, and 2) a value or timing failure may result in severe consequences if such a system hosts a mission-critical application. Moreover, interval-of-time accumulated reward measures are well suited to performability modeling of such systems, because 1) embedded systems typically execute in open or closed loops or cycles[1] each of which corresponds to a unit of time and accommodates at most one state transition, and 2) it is meaningful to quantify a mission's worth in terms of accomplished duty cycles.

However, performability modeling for those systems can be difficult. In particular, while embedded systems have relatively simple architectures and functionalities, the behavior of an embedded system is often non-Markovian in nature. For example, an application may require its host to take a particular action with a specified frequency or to be engaged in a specific operation through a pre-designated period of time, which implies a nonhomogeneous transition probability and a deterministic sojourn time, respectively.

While those non-Markovian properties can be circumvented using the notion of embedded Markov chain (as such a process behaves just like an ordinary Markov process at the instants of state transition), performability measures based on accumulated reward can yet be difficult to solve analytically. In addition, embedded applications may involve path-dependent behavior, which prevents a reward model from being analytically manageable. Although simulation-based performability modeling tools are flexible and powerful, using simulation is usually time-consuming and may lose its advantages when a modeler desires to get insights from analytic results such as a reachability graph or symbolic solutions which are unlikely to be obtained from a simulation.

With the above observations, we develop an approach that uses a recurrence-relation-based reward model to represent a system's DTSMP behavior and to obtain performability-measure solutions. Previously, we leveraged recurrence relations for reliability assessment of a fault-tolerant bus architecture for an avionics system [1]. In addition, we built and solved a recurrence-relation-based reward model to evaluate performability in terms of expected accumulated reward for a distributed embedded system [2]. More recently, we have been further investigating the idea and attempting to generalize it to a certain degree.

The remainder of the article is organized as follows. Section II informally describes the basic elements of an RRB reward model and how it is constructed. Section III explains how such a

[1] In the remainder of the text, the words "loop," "cycle," "frame," and "iteration" are used interchangeably.

model can be solved analytically for performability measures based on an example. Section IV briefly presents an application to exemplify the applicability of RRB reward models to embedded systems. Section V concludes this extended abstract.

II. RRB Reward Model

As mentioned in Section I, a major class of embedded systems can be represented by DTSMPs. More specifically, such systems have the following characteristics which traditional Markov reward models may not be able to handle:

- C1) *Path- and time-dependent transitions*: A transition from S_j in the i^{th} cycle will be dependent of the path via which the system entered to S_j (prior to cycle i) or dependent of the elapsed time since the most recent traversal of a specific path.
- C2) *Deterministic or nondeterministic time-triggered transitions*: A transition $\mathcal{T}$ will be enabled in the i^{th} cycle only if it is a pre-designated cycle during which $\mathcal{T}$ is allowed to fire. If $\mathcal{T}$ will fire with probability 1 in a pre-designated cycle or frequency, the transition is a deterministic time-triggered transition, otherwise it is a nondeterministic time-triggered transition.
- C3) *Deterministic sojourn times*: The duration (quantified in cycles) through which the system in question will be in a particular state S_k is a constant.

Accordingly, a reward model for the type of semi-Markov process we are concerned with should be able to 1) allow an impulse reward to be accrued at the end of each cycle (which is also the epoch of a new cycle), and 2) support interval-of-time performability measures such as expected accumulated reward and instant-of-time measures such as the probability that the system will be in S_k at the end of cycle i.

In order to construct recurrence-relation-based reward models, we use a state diagram to specify a DTSMP and introduce the following basic elements to represent the system characteristics described above: 1) indexed transition probability, 2) state-entry probability, 3) colored arc, and 4) indicator variable. The role of each basic element is described in the following subsections.

A. Indexed Transition Probabilities

An indexed state transition probability is expressed as $\gamma^{(j,k)}[i]$, where i refers to the i^{th} execution cycle, j and k are state identification numbers, and (j,k) means a transition from state S_j to state S_k (that occurs in cycle i).

By labeling arcs using $\gamma^{(j,k)}[i]$, we are able to 1) draw a state diagram such as the one shown in Figure 1, and 2) derive path- and/or elapsed-time-dependent transition probabilities. Furthermore, coupled with $P_k[i]$, the probability that the system in question will be in S_k at the end of cycle i, $\gamma^{(j,k)}[i]$ allows us to derive recurrence relations. In turn, the recurrence relations enable us to obtain an analytic reward-model solution, as explained in Section III.

B. State-Entry Probability

A state-entry probability, denoted as $\hat{P}_{(j,k)}[i]$, is the likelihood that the system will enter into state k from state j, $k \neq j$, in cycle i. This probability enables us to simplify the construction and solution of a DTSMP model in which one or more states are characterized by deterministic sojourn times.

In particular, from model construction perspective, $\hat{P}_{(j,k)}[i]$ is important since it helps leverage the embedded Markov process in an RRB model (recall that each cycle accommodates one and only one transition). From model solution perspective, $\hat{P}_{(j,k)}[i]$ simplifies the derivation of recurrence relations and computation of accumulated reward for such a DTSMP, as shown in Sections III-B and III-C.

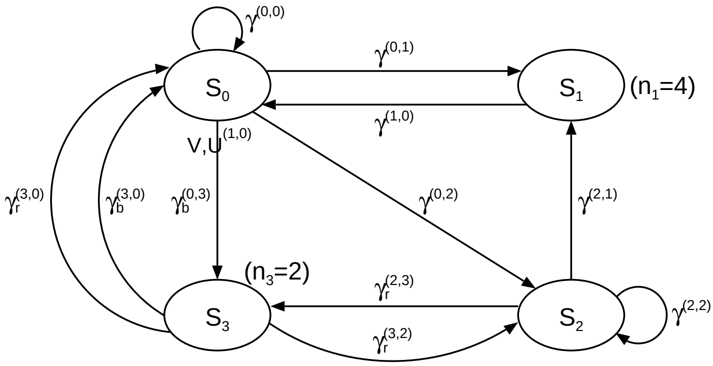

Fig. 1. An Example DTSMP

C. *Colored Arc*

A colored arc is an arc that represents a transition from one state to another and that is a segment of a path consisting of arcs of the same color. Colored arcs in a state diagram enable the specification and computation of path-dependent transition probabilities. As depicted in Figure 1, two different paths (or path sets) can be distinguished by their colors which are denoted by the subscripts of transition probabilities (i.e., b and r mean blue and red colors, respectively).

Note also that in Figure 1 the blue and red paths merge at state S_3 and then branch out again from that state. The values of the branch probabilities, namely $\gamma_b^{(3,0)}[i]$, $\gamma_r^{(3,0)}[i]$, and $\gamma_r^{(3,2)}[i]$, are dependent of the transition probabilities associated with the arcs of the same colors before merging at S_3.

D. *Indicator Variable*

An indicate variable is a function which defines an enabling or disabling predicate for a particular state transition. One or more indicator variables can be used to specify a transition. RRB reward models accommodate two types of indicator variable as described in the following.

With an indicator variable of the first type, the outcome of an enabling predicate indicates the possibility of a deterministic transition. For example, in Figure 1, the transition from S_0 to S_3 is labeled by an indicator variable $V[i, \phi]$ which characterizes a time-triggered transition $\mathcal{T}$ which is allowed to fire per a specified frequency $\frac{1}{\phi}$ (i.e., $\mathcal{T}$ can be activated at most once every ϕ cycles). More succinctly, such an indicator variable is a function which returns 1 when the next epoch is reached and returns zero otherwise:

$$V[i, \phi] = \begin{cases} 1 & \text{if } i \bmod \phi = 0 \\ 0 & \text{otherwise} \end{cases}$$

where ϕ is the time interval[2] between two consecutive transition activations.

With an indicator variable of the second type, a disabling predicate is defined based upon 1) a path through which the system in question $\mathcal{M}$ reaches the current state through a transition $\mathcal{T}$, and 2) the elapsed time since $\mathcal{T}$. Note that the firing of $\mathcal{T}$ itself can be deterministic or probabilistic. Accordingly, a positive outcome of the predicate indicates that a deterministic or probabilistic transition is forbidden. When the transition is probabilistic, the corresponding disabling predicate will be translated into probabilistic terms for computing transition probabilities (see Section III).

In Figure 1, the indicator variable $U^{(1,0)}[i, j, \theta]$ attached to the transition from S_0 to S_3 means that the transition is disallowed to fire if the most recent transition from S_1 to S_0 occurred less

[2]In the remainder of the text, all parameters involving time presume that time is quantified in number of cycles.

than or equal to θ cycles ago with respect to the current cycle i. More precisely,

$$U^{(1,0)}[i,j,\theta] = \begin{cases} 1 & \text{if } i-j \leq \theta \\ 0 & \text{otherwise} \end{cases}$$

Table I summarizes the basic elements discussed in this section and their roles with respect to model construction.

TABLE I
BASIC ELEMENTS OF RRB REWARD MODEL

	Path-dependent transition	*Deterministic sojourn time*	*Time-dependent transition*	*Recurrence relation*	*State-space reduction*
Indexed transition probability	✔	✔	✔	✔	
State-entry probability		✔		✔	✔
Colored arc	✔			✔	✔
Indicator variable	✔		✔	✔	

III. ANALYTIC SOLUTION

Clearly, the key to the analytic solution of an RRB reward model is the derivation of recurrence relations. However in order to derive those relations, we must first compute the transition probabilities, as described in the following subsection.

A. *Computation of Transition Probabilities*

Before proceeding to discuss the computation, we define the following additional notation:

$W_k[i]$: A recursive function which yields a value equal to the product of the expected reward accumulated up to the end of cycle i (given that the system is in S_k then) and $P_k[i]$.

$\gamma_c^{(j,k)}[i]$: Probability of a transition that occurs in cycle i from S_j to S_k and that leads or follows a path with the same color[3] c, given that at the end of cycle $(i-1)$ the system is in S_j.

For a transition that is neither time nor path dependent, the transition probability will simply be a constant, similar to the case of a discrete-time Markov chain (DTMC). Otherwise, analytic work is required for computing the transition probability. For example, for a deterministic time-triggered transition from S_j to S_k with a frequency $\frac{1}{\phi}$, the probability will simply equal the value of the indicator variable $V[i,\phi]$. If a time-triggered transition $\mathcal{T}$ is nondeterministic (meaning that while $\mathcal{T}$ fires only at an epoch implied by ϕ, $\mathcal{T}$ may skip a firing according to a probability), then the transition probability will be formulated as $\gamma_c^{(j,k)}[i]V[i,\phi]$.

For the case where the source state of a transition $\mathcal{T}$ has a constant sojourn time n, the formulation of its transition probability will involve a state-entry probability. Consider the transition from S_1 to S_0 in Figure 1. Since the source state S_1 has a deterministic sojourn time n_1 and since the transition will fire with a probability of 1 at the end of the ${n_1}^{th}$ cycle after entering S_1, the transition probability is formulated as $\gamma^{(1,0)}[i] = \hat{P}_1[i-n_1]$.

When a transition is time and path dependent, the corresponding indicator variable(s) will enable a systematic computation of the transition probability. Consider the following transition sequence in Figure 1.

$$S_1 \xrightarrow{\mathcal{T}_1} S_0 \xrightarrow{\mathcal{T}_2} S_3.$$

Note that the indicator variable $V[i,\phi]$ means that $\mathcal{T}_2$ will be activated only if the current cycle falls on an epoch implied by the specified frequency $\frac{1}{\phi}$. Whereas the other indicator variable

[3]The subscript will be absent when the transition probability is not path-dependent.

$U^{(1,0)}[i,j,\theta]$ (a disabling predicate) implies that $\mathcal{T}_2$ is not allowed to fire unless the most recent activation of $\mathcal{T}_1$ occurred more than θ cycles ago. Then by including $V[i,\phi]$ directly into the formulation of $\gamma^{(0,3)}[i]$ and by translating $U^{(1,0)}[i,j,\theta]$ into a term which helps exclude the probability that $\mathcal{T}_1$ was activated less than θ cycles ago, we have

$$\gamma^{(0,3)}[i] = \left(1 - \sum_{j_1=1}^{\min\{\theta,i-1\}} \gamma^{(1,0)}[i-j_1] \prod_{j_2=1}^{j_1-1} \gamma^{(0,0)}[i-j_2]\right) V[i,\phi].$$

B. Derivation of Recurrence Relations

When all the state transition probabilities are solved, we will be ready to derive a complete set of recurrence relations which collectively enable us to solve all the state probabilities. This step is fairly close to the solution process for solving a DTMC, but we have to deal with more boundary conditions due to deterministic sojourn times and path-/time-dependent transitions.

Consider S_2 in the example DTSMP in Figure 1. Based on indexed transition probabilities and state-entry probabilities, we formulate $P_2[i]$ as follows:

$$P_2[i] = \begin{cases} \sum_{k\in\{0,2\}} \gamma^{(k,2)}[i]P_k[i-1] + \gamma_r^{(3,2)}[i]\hat{P}_{(2,3)}[i-n_3] & \text{if } i \geq n_3 \\ \sum_{k\in\{0,2\}} \gamma^{(k,2)}[i]P_k[i-1] & \text{otherwise} \end{cases}$$

where $\hat{P}_{(2,3)}[i-n_3]$ (instead of $P_3[i-1]$) is used because S_3 has a deterministic sojourn time n_3. Note also since S_1 has a deterministic sojourn time n_1, we have

$$P_1[i] = \sum_{j=0}^{\min\{n_1-1,i-1\}} \sum_{k\in\{0,2\}} \hat{P}_{(k,1)}[i-j].$$

C. Accumulated Reward Computation

The state probabilities derived from recurrence relations in turn enable us to solve the expected reward accumulated through a mission period τ. Letting this performability measure be denoted as $W[\tau]$ and using the theorem of total expectation, we have $W[\tau] = \sum_k W_k[\tau]$, where $W_k[\tau]$ is a recursive function which will yield a value equal to the product of the expected reward accumulated up to the end of cycle τ (given that the system is in S_k then) and $P_k[\tau]$.

The solution process for $W_k[i]$ resembles that for $P_k[i]$. Nonetheless the formulation must ensure that an impulse reward (with a correct magnitude) will be accrued in each iteration. Consider again the example shown in Figure 1. Let us assume that S_1 is a state in which the system makes progress and an impulse reward with a unity magnitude should be collected for each cycle in S_1. Then we have (recall that S_1 has a deterministic sojourn time n_1),

$$W_1[i] = \sum_{j=0}^{\min\{n_1-1,i-1\}} \sum_{k\in\{0,2\}} \gamma^{(k,1)}[i-j](W_k[i-j-1] + P_k[i-j-1]\, j)$$

However if we assume that S_1 is a state during which the system makes no contribution to its mission (so that a reward impulse that has a magnitude of zero will be accrued), then we have

$$W_1[i] = \sum_{j=0}^{\min\{n_1-1,i-1\}} \sum_{k\in\{0,2\}} \gamma^{(k,1)}[i-j]W_k[i-j-1]$$

Note that for those states that do not have a deterministic sojourn time (e.g., S_0 and S_2), reward is accumulated in an "incremental" fashion, meaning that a single impulse reward will be collected through each iteration of a recursive reward function $W_k[i]$. Further solution details for the example model are omitted here due to limited space, but important formulas for a general RRB model can be found in the Appendix.

IV. A Demonstrative Application

Thus far we have used solely the running example DTSMP shown in Figure 1 to illustrate the notion of RRB reward model without interpreting such a DTSMP. In this section we briefly exemplify the applicability of RRB reward models to embedded systems, through discussing an extended version of the running example DTSMP (see Figure 2).

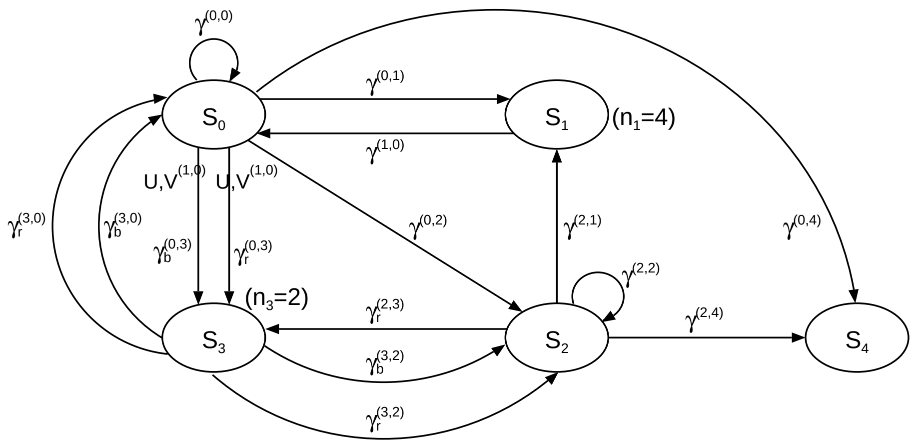

Fig. 2. Application for an Image-Based Tracking System

The DTSMP in Figure 2 represents an image-based tracking system that is typically embedded in an unmanned vehicle. The system tracks a moving target $\mathcal{G}$ by image taking and predicting $\mathcal{G}$'s subsequent position according to $\mathcal{G}$'s trajectory that is updated per the most recent image. Toward the end of an iteration i_{n-1}, the camera takes yaw and pitch adjustments per the prediction, followed by another image taking; and then the system goes to the next iteration i_n. Clearly this is a typical embedded-system application that executes in a closed loop. Note that such a closed-loop computation may allow errors to manifest and accumulate to cause improper camera positioning (S_2) or target missing (S_4). In addition, due to the complexity of the embedded software which involves image processing and pattern recognition, the system is vulnerable to errors caused by memory exhaustion.

Accordingly, proactive process restart is used to periodically clean up memory buffers to avoid problems. Moreover, in order to reduce the performance cost of the preventive measure, a policy is enforced such that a proactive restart can be exempted if a reactive reboot (S_1) due to a detected error occurred at most θ cycles ago. This policy is represented by the indicator variable $U^{(1,0)}[i, j, \theta]$ (see Figure 2).

Finally, by using colored arcs to specify path-dependent transition probabilities, we are able to distinguish the cases in which the system has and has not been affected by an error before undergoing a proactive restart (indicated by the red and blue colors, respectively). This distinction is necessary because some value errors in target tracking may not be self-compensated via the computation cycle following a proactive restart. We obtained some preliminary results from this model and will discuss the results in a future paper when more quantitative analyses are done.

V. Concluding Remarks

We have developed an initial performability-evaluation framework for an important class of embedded systems, namely open- and closed-loop applications. Although prior research efforts resulted in a number of formalisms for modeling real-time and embedded systems, their primary objective was system property verification and validation. In contrast, the emphasis of our work is an approach to systematically constructing and analytically solving RRB reward models for quantitative performability evaluation.

It is worth noting that colored arcs and enabling/disabling predicates specified in the indicator variables enable us to reduce the state space of an RRB model. Moreover, the notation and graphical primitives for DTSMP specifications and the recurrence-relation-based solution mechanisms are intended to facilitate RRB model specification and to enable solution automation, respectively. Currently, we are working toward a complete and improved framework. In particular, we are attempting to simplify the sets of mathematical notation and DTSMP graphical primitives. Finally, as the underlying model is a DTSMP, the framework fits well for hierarchical and hybrid performability modeling. In particular, state transition probabilities can be derived and computed, when necessary, using high-level modeling formalisms such as stochastic activity networks, as we studied in [2]. In the subsequent work, we will further investigate the feasibility of combining RRB reward models with other modeling formalisms.

REFERENCES

[1] A. T. Tai, S. N. Chau, and L. Alkalai, "COTS-based fault tolerance in deep space: A case study on IEEE 1394 application," *International Journal of Reliability, Quality and Safety Engineering*, vol. 9, pp. 17–40, Mar. 2002.

[2] A. T. Tai and W. H. Sanders, "Performability modeling of coordinated software and hardware fault tolerance," in *Proceedings of the Sixth International Workshop on Performability Modeling of Computer and Communication Systems (PMCCS-6)*, (Monticello, Illinois), pp. 31–34, Sept. 2003.

APPENDIX
GENERAL FORMULAS FOR RRB MODEL SOLUTIONS

By definition

$$\hat{P}_{(\hat{k},k)}[i] = P_{\hat{k}}[i-1]\gamma^{(\hat{k},k)}[i], \quad k \neq \hat{k} \tag{1}$$

Then letting $\tilde{K}$ be the set of identification numbers of all the states that have deterministic sojourn times (each of which is quantified in number of cycles, $n_{\hat{k}}$), we have

$$P_k[i] = \sum_{\hat{k}\in\{K-\tilde{K}\}} \gamma^{(\hat{k},k)}[i]P_{\hat{k}}[i-1] + \sum_{\hat{k}\in\tilde{K}} \gamma^{(\hat{k},k)}[i]\hat{P}_{(\hat{k},k)}[i-n_{\hat{k}}]\; I[i,n_{\hat{k}}] \tag{2}$$

where

$$I[i,n_{\hat{k}}] = \begin{cases} 1 & \text{if } i \geq n_{\hat{k}} \\ 0 & \text{otherwise} \end{cases}$$

We let $\omega_{(\hat{k},k)}$ denote the reward impulse for a cycle which begins at $S_{\hat{k}}$ and ends at S_k. Further, we let K and $\gamma^{(\hat{k},k)}$ denote, respectively, the complete set of the state identification numbers and the "merged" (from the arcs of different colors) transition probability. Then, when S_k has a probabilistic sojourn time,

$$W_k[i] = \sum_{\hat{k}\in K} \gamma^{(\hat{k},k)}[i] \left(W_{\hat{k}}[i-1] + P_{\hat{k}}[i-1]\; \omega_{(\hat{k},k)} \right) \tag{3}$$

but when S_k has a deterministic sojourn time,

$$W_k[i] = \sum_{j=0}^{\min\{n-1,i-1\}} \sum_{\hat{k}\in\{K-k\}} \gamma^{(\hat{k},k)}[i-j] \left(W_{\hat{k}}[i-j-1] + P_{\hat{k}}[i-j-1] \left(\omega_{(\hat{k},k)} + (j-1)\omega_{(k,k)} \right) \right) \tag{4}$$

where j is interpreted as the elapsed time since the system $\mathcal{M}$ enters S_k.