Submitted to 20th IEEE Conference on Decision and Control, San Diego, Ca. Dec. 1981

20th CDC

Analytic Models of Multitask Processes*

Timothy L. Johnson

Lecturer
Department of Electrical
Engineering & Computer Science
M.I.T., Rm. 35-205B
Cambridge, Mass. 02139

Senior Scientist
Bolt, Beranek and Newman, Inc.
50 Moulton Street
Cambridge, Mass. 02238

Abstract

Asynchronous multitask processes occur in a wide variety of control applications ranging from industrial control to computer operating systems, yet no analytical methods are available for studying their detailed behavior. The preliminary results reported here illustrate that a very general class of such processes can be represented by discontinuous hybrid-state discrete-time systems.

---

*This research has been performed at the M.I.T. Laboratory for Information and Decision Systems with support provided by the U.S. Air Force Office of Scientific Research under Contract F49620-80-C-0002. The results presented here do not necessarily represent the views of the U.S. Government.

Background and Motivation

A multitask process is characterized by a number of tasks which operate concurrently or sequentially, on an external resource or data base. The timing of the tasks is generally asynchronous in that new task execution is initiated by the completion of previous tasks. If necessary, synchrony and sequential ordering of tasks can be enforced in a number of ways through the task definitions themselves. However in this research no such constraints are imposed: rather, the general qualitative behaviors which may arise in such systems are analyzed. Only two basic assumptions are imposed: (1) a task requires a finite amount of time and storage to execute, and (2) task descriptions are fixed, in that the execution of a task cannot alter its own nature nor the number or nature of any other tasks.

The range of possible behavior of such systems is so large that the problem of conceptualizing, analyzing and "debugging" multitask processes is very common and enormously complex. Two approaches are presently in use: stochastic queueing analysis [1],[2] and simulation [3],[4]. Queueing analysis is most useful for evaluating the average performance properties of an operational multitasking system, while simulation allows certain undesirable properties of a planned system to be discovered and corrected during the design process. Neither of these methods provides very much insight about generic problems in the design of such systems, nor do they provide ideas about how to remedy or detect flaws. The results reported here constitute a modest step in that direction.

Model Development

Let $t\varepsilon[t_o,\infty)$ denote time. Three sets of state variables will be identified:

x^1 - those states which vary continuously with time and take on real values.

x^2 - those states which are real-valued but change only at discrete instants of time

x^3 - those states which are discrete-valued and (necessarily) change only at discrete instants of time.

The state set is denoted $X = \{X^1, X^2, X^3\}$. For present purposes, it will be assumed that these sub sets of states are finite-dimensional and recognizable; an example will be provided below. Let the increasing sequence $\{t_k\}$ denote the set of all values of t for which changes in at least one element of x^2 or x^3 occur, and let the values of the states prior to and following t_k be denoted $x_k^{i^-}$, $x_k^{i^+}$, respectively, for i = 1,2,3. In the sequel, $x_k^{i^+}$ will be identified with x_k^i.

The instants $\{t_k\}$ will be identified with task initiation or termination times. Let the set of tasks in the system be denoted $G \doteq \{G_1 \ldots G_n\}$. Associated with each task is an iniation function, a termination function and a state-update function[1]:

$g_j^I: X \to \{0,1\}$ - initiation function for task j

$g_j^T: X \to \{0,1\}$ - termination function for task j

$f_j: X \to X$ - state-update function for task j

Each task is either "on" or "off": let $\hat{G}$ denote those tasks which are "on" and $\check{G}$ denote those which are off, so that $G = \hat{G} \cup \check{G}$ and $\hat{G} \cup \check{G} = \phi$ (the null set). The subscript $\hat{j}$ will be used to denote tasks which are "on" and $\check{j}$

---

[1]To simplify this exposition, these are assumed to be time-invariant; however, this assumption may be relaxed.

to denote tasks which are "off". The task succession rule is as follows: A transition time, t_k, is declared whenever

(a) For some $j \varepsilon \{\check{j}\}$, $g_j^I(x^1, x^2, x^3)$ undergoes a 0→1 transition

or

(b) For some $j \varepsilon \{\hat{j}\}$, $g_j^T(x^1, x^2, x^3)$ undergoes a 0→1 transition

Between task transition times, only the states x^1 can change, according to a state equation

$$\dot{x}^1(t) = f(x^1(t), x_k^{2^+}, x_k^{3^+}) \tag{1}$$

with $x^1(t_k) = x_k^{1^+}$. At the completion time t_k^- of a task j, the transformation

$$\begin{bmatrix} x_k^{1^+} \\ x_k^{2^+} \\ x_k^{3^+} \end{bmatrix} = f_j \left(\begin{bmatrix} x_k^{1-} \\ x_k^{2-} \\ x_k^{3-} \end{bmatrix} \right) \tag{2}$$

is applied, with $x_k^{2-} = x_{k-1}^{2^+}$ and $x_k^{3-} = x_{k-1}^{3^+}$.

At a transition time, it is possible that more than one task terminates and/or more than one task is initiated. This produces an inherent conflict situation which must be resolved in a consistent manner. For instance, if tasks j_1 and j_2 terminate together, it is not necessarily true that $f_{j1} \circ f_{j2} = f_{j2} \circ f_{j1}$ (functional composition may not be commutative). Or if task j_1 is initiated when j_2 terminates, then up-dating with f_{j2} may turn off j_1, while terminating j_1 may turn on j_2 again, etc. In this preliminary abstract, it will be assumed that

• there is a fixed priority among task completions (e.g. 1 > 2 > 3 > j > ...> n)

- all completions are performed first according to priority, and then initiation functions are re-evaluated to redetermine which tasks (if any) should be initiated at the transition times.

Other conflict-resolution methods, such as imposed sequential orderings, are also possible.

Let $\hat{j}_k \in 2^n$ be the set of tasks active at t_k^+. Let the transition mapping of (1) be given by ϕ: $[0,\infty) \times X^1 \to X^1$, so that the solution of

$$\dot{x}^1(t) = f(x^1, x_k^2, x_k^3) \; ; \; x^1(t_k) = x_k^1 \tag{3}$$

is

$$x^1(t) = \phi(t-t_k, x_k^1; \; x_k^2, x_k^3) \tag{4}$$

where x_k^2, x_k^3 are viewed as parameters. Define the function τ: $2^n \times X \to R^+$ to be the first transition-time encountered with processes $\hat{j} \in 2^n$ active at $t = t_o$, with initial state $x = (x^1, x^2, x^3) \in X$. This can be tabulated by integrating (1) and applying rules (a) and (b). Let the function σ: $2^n \times X \to 2^n$ define the next set of active tasks, determined from the preceding priority rules, at the transition time defined by . In other words,

$$t_{k+1} = t_k + \tau(\hat{j}_k, x_k^1, x_k^2, x_k^3) \tag{5}$$

$$\hat{j}_{k+1} = (\hat{j}_k, x_k^1, x_k^2, x_k^3) \tag{6}$$

The important point to observe is that, in principle, it is not necessary to include the continuous-time part of the dynamics, since τ and σ can be pre-computed from f, $\{g_j^I\}$, and $\{g_j^T\}$.

In summary, the dynamics of the asynchonous multitask system can always be represented in the form

$$\left.\begin{aligned} t_{k+1} &= t_k + \tau(\hat{j}_k, x_k) \\ x_{k+1} &= f_{\hat{j}_k}(x_k) \\ j_{k+1} &= \sigma(\hat{j}_k, x_k) \end{aligned}\right\} \tag{7}$$

where $f_{\hat{j}_k}$ is the composition, according to priority, of the transition functions (2) of the tasks completing at t_{k+1}^-. It is then clear that t_{k+1} may be combined with x^1 and x^2, and that x^3 may be combined with $\hat{j}$ to yield a general discontinuous hybrid discrete-time system. Extensions to stochastic behavior of $f, \{f_j\}$, $\{g_j^I\}$ and $\{g_j^T\}$ are readily accomodated.

Qualitative Properties

The finite-state part of (7) may be further aggregated to produce an equivalent real-state discrete-time system with discontinuous transition function. Systems of this general class have been discussed by Johnson [5] and Kaliski and Lemone [6]. Their behavior may roughly approximate the behavior of discontinuous systems discussed in Utkin [7] and Johnson [8]. The pertinent properties of such systems will be described more fully in the final version of this paper. Here it is merely noted that problems may arise if $\lim_{k\to\infty} t_k$ is finite. A possible behavior in this situation is an approximation to sliding mode behavior, which is closely akin to the phenomenon of "thrashing" observed in heavily-loaded multitasking systems.

Examples

Realistic examples will be provided in the conference version of this paper.

References

[1] M.G. Kienzle and K.C. Sevcik, "Survey of Analytic Queueing Network Models of Computer Systems," Proc. 1979 Conf. on Simulation, Measurement and Modelling of Computer Systems, pp. 113-129.

[2] G.K. Hutchinson and J.J. Hughes, "A Generalized Model of Flexible Manufacturing Systems", Proc. Multi-Station Digitally-Controlled Mfg. Systems Workshop, Milwaukee, Wisc. Jan. 1977; pp. 88-115.

[3] Shannon, R.E., Systems Simulation: The Art and Science, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.

[4] Pritsker, A.A.B., The GASP IV Simulation Language, J. Wiley & Sons, New York, 1974.

[5] Johnson, T.L., "Finite-State Compensators for Continuuous Processes", Proc. 7th IFAC Congress, Helsinki, Finland, June 1978, pp. 1823-1828.

[6] Kaliski, M.E. and Lemone, K., "Discrete Codings of Continuous Valued Signals", Proc. 14th Ann. Conf. on Info. Sci. and Sys., Johns Hopkins Univ., March 1980.

[7] Utkin, V.I., Sliding Modes and Their Application to Valuable-Structure Systems, MIR Publishers, Moscow, 1978.

[8] Johnson, T.L., "Stability of Diced Systems", Proc. 19th IEEE Conf. on Decision and Control, Albuquerque, N.M., Dec. 1980.