To Appear in IEEE Transactions on Control Systems Technology,
vol. 2, no. 2, pp. 110–122, June, 1994

On the Maximum Feedback Delay in a Linear/Nonlinear Control System with Input Disturbances Caused by Controller-Computer Failures[1]

Hagbae Kim and Kang G. Shin

Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109–2122

Abstract

Electromagnetic interferences or other environmental disturbances may cause transient failures to the controller computer of a real-time control system. Such a faulty controller either fails to update the control input for one or more sampling periods, or generates erroneous control inputs until the failure is handled properly or disappears.

The goal of this paper is to derive the maximum duration of controller's faulty behavior, called the *hard deadline*, a real-time control system can tolerate without losing stability or leaving its allowed state space. For linear time-invariant control systems, one can derive hard deadlines by testing the stability of their state difference equations which account for the effects of stationary occurrences of disturbances to, as well as the random delays in, the control input. Similarly, one can derive deadlines for nonlinear time-invariant control systems by linearizing their nonlinear state equations and using the Lyapunov's first method. In addition to this stationary model, a one-shot event model is considered for linear/nonlinear time-invariant control systems by using their state trajectories and allowed state spaces. The hard deadline information that represents the knowledge of the controlled process's inertia and timing constraints is applied to the design and evaluation of controller computers.

Index Terms — Real-time control systems, fault-tolerance, computation-time delay, input disturbances, controller computer, hard deadline, stationary and one-shot event models.

[1]The work reported was supported in part by the Office of Naval Research under Contract No. N00014–91–J–1115 and by the NASA under Grant No. NAG–1–1120. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the view of the funding agencies.

1 Introduction

Most real-time control systems consist of two synergistic parts: the *controlled process* or *environment*, and the *controller computer*. The control programs, which are executed by a controller computer residing in the feedback loop, realize a set of functions using sensory data from the controlled process and/or from the environment at regular time intervals.

Since the controller computer is susceptible to transient electromagnetic interferences inducing mainly functional errors, often without damaging any of its components, it is usually equipped with some fault-tolerance mechanisms, especially for life-, or safety- critical systems like aircraft or nuclear reactors. When an abnormality (component failure or environmental interference) of the controller computer occurs, there are two possible outcomes:

- The controller generates an erroneous control input or an input *disturbance* due to erroneous computations, and
- The controller fails to update the control input until the abnormality is detected and handled properly. That is, there will be a *delay* in the feedback control loop.

The stationary occurrences of these two types of abnormality — which depend upon the stochastic nature of the environment — may lead to the loss of system stability if their active duration exceeds a certain limit called the *hard deadline* [17]. Even one occurrence of this abnormality for a long period may drive the controlled process out of its allowed state space, i.e., a *dynamic failure* occurs, which is called the *one-shot event model*. Some failures in actuators or sensors or mechanical parts and failures of A/D and D/A converters may also induce a system failure. In [11], a mathematical framework was presented to describe the interactions between the detection and isolation device for component (actuator, sensor, or computers) failures and the reconfiguration of the control algorithms. The authors of [21] focused on the design of fault-tolerant control systems to enhance system reliability. By contrast, our main intent is to analyze the coupling between a controlled process and its (fault-tolerant) controller computer (rather than such interfaces as an actuator and a sensor) and to formally specify the deadline information of the controlled process useful for the design and evaluation of the controller computer.

Several researchers qualitatively analyzed the effect of feedback delay resulting from the unexpected delay of data flow or the temporary unavailability of a controller computer by obtaining the stability conditions or proposing means to reduce the delay effects [2, 4, 5, 13, 12, 23]. The quantitative analysis of the effects of computation-time delay was made by deriving hard deadlines for prototypical real-time control systems such as the robot trajectory tracking problem [15] and the aircraft landing problem [17]. In [16], we numerically derived hard deadlines for linear time-invariant control systems based on the fact that computation-time delays are stochastic in both their occurrence times (frequency) and magnitudes (duration) reflecting the nature of computer failures, without using the common assumption that the feedback delay is fixed or periodic. However, we assumed there all computer failures are detected upon their occurrence and appropriate recovery mechanisms are invoked immediately. Thus, the results in [16] considered neither the effects of erroneous control inputs nor nonlinear control systems. The problem associated with disturbances to the control input was treated by analyzing the observability of a linear system with a dynamic feedback controller under unknown disturbances in the control input [6] and experimentally testing the functional error modes of computer-based control systems in a "harsh" operating environment [1].

In this paper, we analyze the effects of stationary occurrences of disturbances to, as well as the delays in, the control input for linear time-invariant control systems. For this purpose, hard deadlines are derived by examining the stability of state difference equations modified with random sequences representing: (i) stationary occurrences of computer failures, (ii) imperfect error coverage [18] (with binomial distributions), (iii) the duration of failures/interferences (with multinomial distributions), and (iv) the magnitude of disturbances to the control input (with a normal distribution). The system dynamics are modified with these random sequences and the augmented transition matrices for a group of N sampling intervals where N is the assumed maximum delay [16]. System stability is then examined for the samples and the ensemble average of the modified equations in order to derive hard deadlines stochastically as well as deterministically. In addition, the hard deadline of a one-shot event model, where a single event (disturbance/delay) — a long-lasting failure/interference — may cause a dynamic failure, is derived by using the state trajectory and the allowed state space. For nonlinear time-invariant control systems, we first linearize the nonlinear dynamics around an operating point and then use well-developed linear systems methods to derive an optimally stabilizing control input and examine system stability (Lyapunov's indirect method) in the presence of additional feedback delays.

Section 2 reviews the basic definitions related to real-time control, and states the importance of the deadline information in a real-time control system. In Section 3, we introduce the basic assumptions and the random sequences that characterize system failures and input disturbances. Then, hard deadlines are derived deterministically and stochastically with the modified state difference equations for linear time-invariant control systems. Two different approaches for stationary and one-shot event models are presented there. Section 4 treats the hard deadlines of nonlinear time-invariant control systems, without considering control input disturbances. The linearization method is applied for this analysis. Section 5 presents four examples of deriving hard deadlines from both the stationary and one-shot event models for linear and nonlinear time-invariant control systems. Moreover, we present not only a design example that optimizes time redundancy such as retry or rollback, but also an evaluation example that assesses the system reliability by using the derived deadline informations. The paper concludes with Section 6.

2 Generic Problems Related to Controller-Computer Failures

Controlled processes are generally represented by a state-space model as shown in Eq. (2.1) and are equipped with well-designed controllers that stabilize the overall control system and optimize the given control objectives:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{f}(k, \mathbf{x}(k), \mathbf{u}(k)), \\ \mathbf{u}(k) &= \mathbf{g}(k, \mathbf{x}(k)), \end{aligned} \tag{2.1}$$

where k is the time index, one unit of time represents one sampling interval T_s, $\mathbf{f} : \mathcal{R}^{n+\ell+1} \to \mathcal{R}^n$ and $\mathbf{g} : \mathcal{R}^{n+1} \to \mathcal{R}^\ell$ are smooth and continuously differentiable functions, and $\mathbf{x} \in \mathcal{R}^n$ and $\mathbf{u} \in \mathcal{R}^\ell$ are the state and input vectors, respectively. The controllers are implemented with digital computers in the feedback loop, and A/D and D/A converters [16], since (i) fast, accurate, and consistent controls are required of most real-time control systems with increasing sophistication of the controlled processes, and (ii) the capability and reliability of digital computers have been steadily improving at a low cost. The control input is computed by the controller computer at regular time intervals without allowing job pipelining, i.e., a control job should be completed or abandoned before its successor in the next sampling interval is invoked.

As mentioned earlier, digital computers are highly susceptible to transient electromagnetic interferences (EMI) such as lightening, high intensity radio frequency fields (HIRF), and nuclear electromagnetic pulses (NEMP). The main problem caused by EMI is functional error modes — or computer failures — often without component damages. When a computer failure occurs and it is detected upon its occurrence (i.e., with a zero *error latency* [18]), a certain recovery process is invoked.[2] Suppose a computer failure is detected on its occurrence at time k_0, and its recovery takes n sampling intervals. The control inputs during these intervals, $\mathbf{u}(k_0+1), \cdots, \mathbf{u}(k_0+n)$, will be held constant at $\mathbf{u}(k_0)$ by the D/A converter and latch circuits. When a computer failure occurs and it is not detected immediately, the recovery process will not be called in immediately: instead, the control input may be updated erroneously until the fault inducing the computer failure disappears or the computer failure is detected and handled properly. This may degrade the system's performance and reliability more significantly than the case of missing control updates. Suppose a computer failure occurs at time k_0, it is detected n_1 sampling intervals after its occurrence, and the subsequent recovery takes n_2 sampling intervals. The control inputs during this period are $\underbrace{\mathbf{u}(k_0+1)\mathbf{I}_{\boldsymbol{\Delta}}, \mathbf{u}(k_0+2)\mathbf{I}_{\boldsymbol{\Delta}}, \cdots, \mathbf{u}(k_0+n_1)\mathbf{I}_{\boldsymbol{\Delta}}}_{n_1}, \cdots, \underbrace{\mathbf{u}(k_0+n_1+1), \mathbf{u}(k_0+n_1+1), \cdots, \mathbf{u}(k_0+n_1+1)}_{n_2}$, $\mathbf{u}(k_0+n_1+n_2+1), \cdots$, where $\mathbf{I}_{\boldsymbol{\Delta}}$ is a diagonal matrix with $Diag[\mathbf{I}_{\boldsymbol{\Delta}}]_i = 1 + \Delta u_i$ and Δu_i is a random sequence which is modeled by the output of a dynamic system with a white-noise input. Since faults/interferences occur randomly during the mission lifetime, their occurrences are considered stochastic perturbations to the controlled process, which can be modeled depending on the fault characteristics. When the environment is assumed to be stochastically stationary, the occurrence and duration of computer failure(s), and the magnitude of disturbances in the control input can be represented by several probability density functions. The relative frequency of disturbance and delay due to such computer failure(s) depends upon the *coverage* (the probability of detecting an error induced by an arbitrary fault), which is determined by failure-detection mechanisms [18].

Stationary occurrences of controller-computer failures/interferences not only degrade the performance of the controlled process but may also lead to loss of system stability if their active durations exceed the hard deadline. Even one occurrence of this abnormality with a long active duration in a one-shot event model may make the system leave its allowed state space [16].

The hard deadline of the stationary model is defined as the maximum duration of the controller computer's failure without losing system stability [16]. More formally, we have the following definition.

Definition 1: Let $\mathbf{x}_e$ denote an equilibrium state of the system represented by Eq. (2.1).

[2] A fault-tolerance mechanism consists of error detection, fault location, system reconfiguration, and recovery. General recovery processes are retry, rollback, and reconfiguration which take a finite time. See [20] for a more detailed account of fault-tolerance mechanisms.

Then, $\mathbf{x}_e$ is said to be *stable* at time k_0 if for each $\epsilon > 0$ there exists a $\delta(\epsilon, k_0) > 0$ such that

$$\|\mathbf{x}(k_0) - \mathbf{x}_e\| \leq \delta \Longrightarrow \|\mathbf{x}(k) - \mathbf{x}_e\| \leq \epsilon, \quad \forall k \geq k_0. \tag{2.2}$$

The equilibrium state $\mathbf{x}_e$ is said to be *asymptotically stable* at time k_0, if it is stable at time k_0, and there exists a $\delta_1(k_0) > 0$

$$\|\mathbf{x}(k_0) - \mathbf{x}_e\| \leq \delta_1(k_0) \Longrightarrow \|\mathbf{x}(k) - \mathbf{x}_e\| \longrightarrow 0 \quad \text{as } k \longrightarrow \infty. \tag{2.3}$$

In linear time-invariant systems, stability can be checked simply by using the pole positions of the controlled process in the presence of random computer failures. Using this information one can derive hard deadlines stochastically or deterministically with the sample(s) and the ensemble average of the controlled process:

$$D(N) = \inf_{C_{env}} \sup\{N \ : \ \|\lambda(N)\| < 1\}, \tag{2.4}$$

where $\lambda(N)$ is the eigenvalue of the controlled process in the presence of computer failures of the maximum duration NT_s and C_{env} represents all the environmental characteristics that cause controller-computer failures.

Consider a state trajectory evolved from time k_0 till k_f. Let $\mathbf{X}_A(k)$ and $\mathbf{U}_A$ be the allowed state space at time k and the admissible input space, respectively. If a computer failure, which occurred at k_1 ($k_0 \leq k_1 \leq k_f$) and was detected N_1T_s later, is recovered within N_2T_s, where $N = N_1 + N_2, \ \ 0 \leq N_1, N_2 \leq N$, then the control input during this period (NT_s) is:

$$\mathbf{u}_a^N(k) = \mathbf{u}(k)\mathbf{I}_{\mathbf{\Delta}}\Pi_{k_1}(N_1) + \mathbf{u}(k_1 + N_1)\Pi_{k_1+N_1}(N_2), \quad k_1 \leq k \leq k_1 + N$$

where $\Pi_m(n)$ is a rectangular function from m to $m+n$, i.e., $\Pi_m(n) = \xi(k-m) - \xi(k-m-n)$ where ξ is the unit step function. Then, the hard deadline of a task during the time interval $[k_0T_s, \ k_fT_s]$ is also defined as:

$$D(N, \mathbf{x}(k_0)) = \inf_{\mathbf{u}_a^N(k) \in \mathbf{U}_A} \sup\{N \ : \ \phi(k, k_0, \mathbf{x}(k_0), \mathbf{u}_a^N(k)) \in \mathbf{X}_A(k), \ k_0 \leq k \leq k_f\}, \tag{2.5}$$

where the state trajectory is assumed to evolve as:

$$\mathbf{x}(k) = \phi(k, k_0, \mathbf{x}(k_0), \mathbf{u}_a^N(k)). \tag{2.6}$$

The hard-deadline information allows us to deduce the timing constraints of the controlled process. It can be used to evaluate the fault-tolerance requirements of a real-time

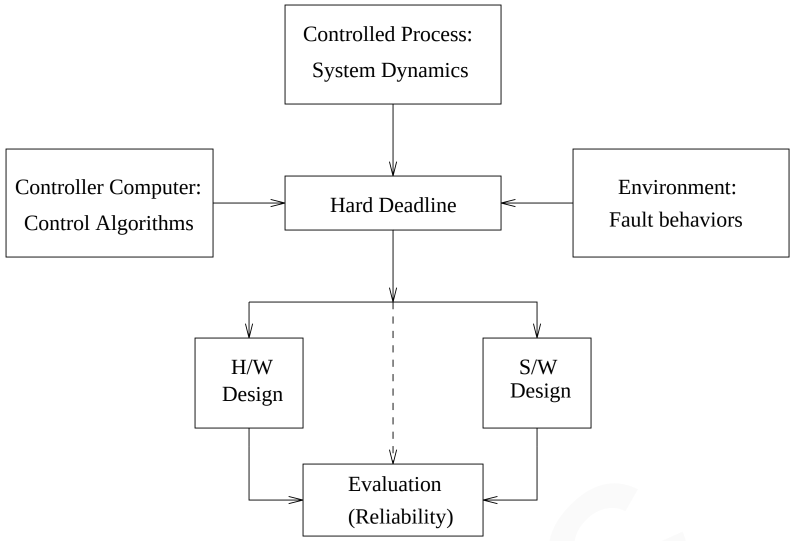

Figure 1: The source and application of hard deadline in a real-time control system.

control system in terms of its timing constraints. Fig. 1 shows the source and application of hard deadline. As we shall see, this deadline information about the controlled process is quite useful for the design and evaluation of the controller computer.

Using the requirements (e.g., deadlines) of the specific control application in hand, one can make decisions on (i) hardware design issues such as the number of processors, the type of interconnection network, the number of power supplies, and the means of synchronizing processors, and (ii) software design issues such as implementation of control algorithms, assignment and scheduling of tasks, handling of interrupts, management of redundancy, and detection and recovery of component failure(s). In a real-time control system, a *system failure* may result either because of not responding fast enough to meet the timing constraints or because of massive component failures. Since the timing constraints of the controlled processes are manifested as hard deadlines, the deadline information is also essential to evaluate the system reliability, an important yardstick to measure the goodness of the controller computer.

3 Hard Deadlines for Time-Invariant Linear Systems with Control Input Disturbances

For a linear time-invariant controlled process, the general state difference equation corresponding to Eq. (2.1) is described by:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \tag{3.1}$$

where $\mathbf{A} \in \mathcal{R}^{n \times n}$ and $\mathbf{B} \in \mathcal{R}^{n \times \ell}$ are the state and input transition matrices with constant elements. The hard deadlines of this model, which are defined by Eq. (2.4) and Eq. (2.5) for the stationary and one-shot event models, respectively, are derived as the maximum delay/disturbance durations which can satisfy the conditions of asymptotic system stability and keep the state trajectory in the allowed state space. The iterative method used in [16] is applied to numerically derive hard deadlines, i.e., the system stability and the state residence of the modified dynamic equation are tested iteratively while changing the assumed maximum duration (NT_s) from T_s to DT_s, where DT_s is the actual maximum duration or the hard deadline.

3.1 Hard Deadline of the Stationary Model

In our model, the controlled processes represented by Eq. (3.1) are usually unstable in the absence of any feedback control, i.e., the real part of at least one eigenvalue of A is greater than one.[3] Some state feedback control inputs stabilizing such unstable systems can be calculated by using the observed (or estimated) states according to their own control objectives such as time-optimal control with an energy constraint, optimal state-tracking, and optimal linear regulator [10]. Suppose the feedback control input is computed by $\mathbf{u}(k) = \mathbf{F}\mathbf{x}(k)$, where the feedback matrix $\mathbf{F}$ depends upon the control objective used. When computer failures occur according to the random environmental characteristics, the control input calculated in the absence of failures is no longer optimal because of additional stochastic input delays and disturbances. The performance of the resulting system may thus be degraded. Furthermore, the asymptotic stability of the sample or ensemble average of the system with respect to the stochastic nature of the environment may be lost if the duration of such a failure exceeds a certain value. To derive this value, the given state equation is modified to include all stochastic behaviors of computer failures based on the

[3] For example, the aircraft designer must push his design to the edge of instability to improve the fuel-efficiency of a future aircraft, where the fast, accurate, and consistent control is required [17].

following random sequences and the assumptions for tractability.

- Definition of random sequences ($i.i.d.$ for the time index k)

  1. p is the probability of a computer failure at each sampling instance.
  2. d is the conditional probability of successful detection if computer failure(s) has occurred.
  3. q_i^d is the conditional probability of delay (recovery duration) for i sampling intervals ($\sum_{i=1}^{N} q_i^d = 1$) if a computer failure occurs and is detected without generating any erroneous input, or input disturbance.
  4. q_i^w is the conditional probability of an input disturbance for i sampling intervals ($\sum_{i=1}^{N} q_i^w = 1$) if a computer failure occurs and is not detected till its disappearance.
  5. $q_{\Delta u}$ is the probability density function (pdf) of Δu which is the magnitude of a control input disturbance at time kT_s, i.e., $\mathbf{u}_{actual}(k) = \mathbf{u}_{desired}(k)\mathbf{I}_{\boldsymbol{\Delta}}$. The mean and variance of $q_{\Delta u}$ are given as $\mu_{\Delta u}$ and $\sigma^2_{\Delta u}$, respectively.

- Basic assumptions

  1. The control inputs calculated after recovering from computer failure(s) are always correct. That is, the probability of successful recovery is assumed to be 1.
  2. The probability that two transient failures occur sequentially within a small number of sampling intervals, $(N-i)T_s$, where the delay (recovery duration) or duration of erroneous inputs (active duration of a transient failure) is i sampling intervals and NT_s is the assumed maximum value of such intervals — is so small as to be ignored. So, we consider only one computer failure possible during a group of N intervals.
  3. Any random sequence will be identically independent distributed ($i.i.d$) for the time index k.

Let the control input have been updated correctly at $k = mNT_s$. If an abnormality (delay/disturbance) is active for i $(1 \leq i \leq N)$ sampling periods from that time due to a computer failure, let the control input at $(mN+i)T_s$ be $\mathbf{u}_a(mN+i)$ which is actually equal to $\mathbf{u}(mN+i)\mathbf{I}_{\boldsymbol{\Delta}}$ for the disturbance case or $\mathbf{u}(mN)$ for the delay case. The corresponding state equations for the group of intervals during which the system failed to update control inputs become:

$$
\begin{aligned}
\mathbf{x}(mN+1) &= \mathbf{A}\mathbf{x}(mN) + \mathbf{B}\mathbf{u}_a(mN) \\
\mathbf{x}(mN+2) &= \mathbf{A}\mathbf{x}(mN+1) + \mathbf{B}\mathbf{u}_a(mN+1) \\
&= \mathbf{A}^2\mathbf{x}(mN) + \mathbf{A}\mathbf{B}\mathbf{u}_a(mN) + \mathbf{B}\mathbf{u}_a(mN+1) \\
&\quad \vdots \\
\mathbf{x}(mN+i) &= \mathbf{A}^i\mathbf{x}(mN) + \sum_{j=0}^{i-1} \mathbf{A}^{i-1-j}\mathbf{B}\mathbf{u}_a(mN+j) \\
\mathbf{x}(mN+i+1) &= \mathbf{A}^{i+1}\mathbf{x}(mN) + \sum_{j=1}^{i} \mathbf{A}^{i-j}\mathbf{B}\mathbf{u}_a(mN+j) + \mathbf{B}\mathbf{u}(mN+i) \\
&\quad \vdots \\
\mathbf{x}((m+1)N) &= \mathbf{A}^N\mathbf{x}(mN) + \sum_{j=N-i}^{N-1} \mathbf{A}^j\mathbf{B}\mathbf{u}_a(mN+N-1-j) \\
&\quad + \sum_{j=0}^{N-i-1} \mathbf{A}^j\mathbf{B}\mathbf{u}(mN+N-1-j),
\end{aligned}
$$

where m is the time index for groups of N sampling intervals each. Let $\mathbf{X}(m) = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T \equiv [\mathbf{x}(mN+1), \mathbf{x}(mN+2), \ldots, \mathbf{x}((m+1)N)]^T$ and $\mathbf{U}(m) = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N]^T \equiv [\mathbf{u}(mN+1), \mathbf{u}(mN+2), \ldots, \mathbf{u}((m+1)N)]^T$. That is, $\mathbf{X}(m)$ and $\mathbf{U}(m)$ are respectively the augmented state and control vectors for a group of N sampling intervals. Then, we get the following augmented state equations:

$$
\begin{aligned}
\mathbf{X}(m+1) &= \mathbf{A}_D\mathbf{X}(m) + \mathbf{B}^1_{D_i}\mathbf{U}(m) + \mathbf{B}^2_{D_i}\mathbf{U}(m+1), && (3.2) \\
\mathbf{U}(m) &= -\mathbf{F}_D\mathbf{X}(m), && (3.3)
\end{aligned}
$$

where $[\mathbf{B}^1_{D_i}, \mathbf{B}^2_{D_i}]$ becomes $[\mathbf{B}^{n1}_{D_0}, \mathbf{B}^{n2}_{D_0}]$ for the normal behavior, $[\mathbf{B}^{d1}_{D_i}, \mathbf{B}^{d2}_{D_i}]$ for delay, and $[\mathbf{B}^{w1}_{D_i}, \mathbf{B}^{w2}_{D_i}]$ for disturbance, respectively. From Eqs. (3.2) and (3.3), the augmented transition matrices are:

$$
\mathbf{A}_D = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}^2 \\ \vdots & & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}^N \end{bmatrix}, \qquad
\mathbf{F}_D = \begin{bmatrix} \mathbf{F} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{F} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{F} \end{bmatrix},
$$

$$
\mathbf{B}_{D_0}^{n1} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{B} \\ \mathbf{0} & \cdots & \mathbf{AB} \\ \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{A}^{N-1}\mathbf{B} \end{bmatrix}, \quad
\mathbf{B}_{D_i}^{d1} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \cdots & \mathbf{0} & (\mathbf{AB}+\mathbf{B}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \sum_{j=0}^{i-1} \mathbf{A}^j\mathbf{B} \\ \mathbf{0} & \cdots & \mathbf{0} & \sum_{j=1}^{i} \mathbf{A}^j\mathbf{B} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \sum_{j=N-i}^{N-1} \mathbf{A}^j\mathbf{B} \end{bmatrix},
$$

$$
\mathbf{B}_{D_0}^{n2} = \begin{bmatrix} \mathbf{0} & \cdots & & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \cdots & & \mathbf{0} & \mathbf{0} \\ \mathbf{AB} & \cdots & & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{A}^{N-2}\mathbf{B} & \cdots & \mathbf{AB} & \mathbf{B} & \mathbf{0} \end{bmatrix}, \quad
\mathbf{B}_{D_i}^{d2} = \begin{bmatrix} \mathbf{0} & & \cdots & & & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \cdots & & & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \cdots & & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{B} & \cdots & & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{AB} & \cdots & & \mathbf{0} & \mathbf{0} \\ \vdots & \cdots & \vdots & \ddots & \ddots & & \vdots \\ \mathbf{0} & \cdots & \mathbf{A}^{N-i-1}\mathbf{B} & \cdots & \mathbf{AB} & \mathbf{B} & \mathbf{0} \end{bmatrix}.
$$

$$
\mathbf{B}_{D_i}^{w1} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{B} \\ \mathbf{0} & \cdots & \mathbf{AB} \\ \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{A}^{N-1}\mathbf{B} \end{bmatrix}, \quad
\mathbf{B}_{D_i}^{w2} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & & \mathbf{0} & \mathbf{0} \\ \mathbf{BI}_{\boldsymbol{\Delta}} & \ddots & \vdots & \vdots & & \mathbf{0} & \mathbf{0} \\ \mathbf{ABI}_{\boldsymbol{\Delta}} & \ddots & & & & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \mathbf{0} & \mathbf{0} & & \vdots & \vdots \\ \mathbf{A}^{i-2}\mathbf{BI}_{\boldsymbol{\Delta}} & \cdots & \mathbf{BI}_{\boldsymbol{\Delta}} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{A}^{i-1}\mathbf{BI}_{\boldsymbol{\Delta}} & \cdots & \mathbf{ABI}_{\boldsymbol{\Delta}} & \mathbf{B} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots \\ \mathbf{A}^{N-2}\mathbf{BI}_{\boldsymbol{\Delta}} & \cdots & \mathbf{A}^{N-i}\mathbf{BI}_{\boldsymbol{\Delta}} & \mathbf{A}^{N-i-1}\mathbf{B} & \cdots & \mathbf{B} & \mathbf{0} \end{bmatrix}.
$$

The state difference equation is modified by using these augmented transition matrices and the random squences representing the behavior of computer failures:

$$
\begin{aligned}
\mathbf{X}(m+1) \quad = \quad & \mathbf{A}_D\mathbf{X}(m) + \left((1-\psi)\mathbf{B}_{D_0}^{n1} + \psi(1-\varphi)\mathbf{B}_{D_i}^{w1} + \psi\varphi\sum_{i=1}^{N}\xi_i\mathbf{B}_{D_i}^{d1}\right)\mathbf{U}(m) + \\
& \left((1-\psi)\mathbf{B}_{D_0}^{n2} + \psi(1-\varphi)\sum_{i=1}^{N}\zeta_i\mathbf{B}_{D_i}^{w2} + \psi\varphi\sum_{i=1}^{N}\xi_i\mathbf{B}_{D_i}^{d2}\right)\mathbf{U}(m+1) \qquad (3.4)
\end{aligned}
$$

where $\psi, \varphi \in \{0,1\}$ are binomially-distributed random sequences with probabilities p, d, and $\xi_i, \zeta_i \in \{0,1\}$ are multinomially-distributed random sequences with probabilities q_i^d, q_i^w, i.e., $\Pr[\xi_i = 1] = q_i^d$.

The asymptotic stability of Eq. (3.4) can be examined deterministically or stochastically.

A. Deterministic Approach

Similar to the method used in [16], the deterministic value of the hard deadline is obtained by examining the pole positions of the first moment (ensemble average) of Eq. (3.4). Although the resulting hard deadline has little practical meaning, it can show the trend of the ensemble system behavior with an uncertainty (in the state and output) which can be characterized by the second moment of Eq. (3.4). The first moment of Eq. (3.4) is:

$$\begin{aligned}\bar{\mathbf{X}}(m+1) \;=\;& \mathbf{A}_D\bar{\mathbf{X}}(m) + \left((1-p)\mathbf{B}^{n1}_{D_0} + p(1-d)\bar{\mathbf{B}}^{w1}_{D_i} + pd\sum_{i=1}^{N} q_i^d \mathbf{B}^{d1}_{D_i}\right)\mathbf{U}(m) + \\ & \left((1-p)\mathbf{B}^{n2}_{D_0} + p(1-d)\sum_{i=1}^{N} q_i^w \bar{\mathbf{B}}^{w2}_{D_i} + pd\sum_{i=1}^{N} q_i^d \mathbf{B}^{d2}_{D_i}\right)\mathbf{U}(m+1). \qquad (3.5)\end{aligned}$$

Using Eqs. (3.3) and (3.5), one can get the characteristic equation of Eq. (3.5):

$$\begin{aligned}\det \quad & \Bigg[\left(\mathbf{I} + [(1-p)\mathbf{B}^{n2}_{D_0} + p(1-d)\sum_{i=1}^{N} q_i^w \bar{\mathbf{B}}^{w2}_{D_i} + pd\sum_{i=1}^{N} q_i^d \mathbf{B}^{d2}_{D_i}]\mathbf{F}_D\right) z^N - \\ & \left(\mathbf{A}_D - [(1-p)\mathbf{B}^{n1}_{D_0} + p(1-d)\bar{\mathbf{B}}^{w1}_{D_i} + pd\sum_{i=1}^{N} q_i^d \mathbf{B}^{d1}_{D_i}]\mathbf{F}_D\right)\Bigg] = 0. \qquad (3.6)\end{aligned}$$

The characteristic equation of the zero-delay case (i.e., no computer failure, or $p = 0$) is:

$$\det\left[z^N\mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{F})^N\right] = 0, \qquad (3.7)$$

where the magnitudes of eigenvalues are equal to those obtained from:

$$\det\left[z\mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{F})\right] = 0 \qquad (3.8)$$

which is the characteristic equation of the original state equation (Eq. (3.1)) in the absence of computer failures.

B. Stochastic Approach

The effectiveness of the deterministic approach decreases as the variance of $q_{\Delta u}$ gets large. In such a case, we can derive the probability mass function (pmf) of the hard deadline with respect to $q_{\Delta u}$ rather than the deterministic value of the hard deadline based on the mean of $q_{\Delta u}$. Now, the mapping between the hard deadlines and the magnitudes of disturbances (Δu's) is not one-to-one, and the hard deadlines can be derived numerically for each sample value of Δu's. In all but simplest cases it is impossible to derive a closed-form expression for the *pmf* of the hard deadline or the exact relation between the hard deadline and Δu. The method we use is therefore to quantize uniformly the $q_{\Delta u}$ continuum in the interval $[a, b]$, where $\int_a^b q_{\Delta u} d\Delta u = \gamma$. Let this quantization result in M equal-length subintervals (cells). There is a tradeoff between the accuracy and the amount of computation in determining γ and M, and a and b are determined appropriately according to γ. Then, points are allocated to the quantized cells, and let the point of the i-th cell $([a+(i-1)\frac{M}{b-a}, a+i\frac{M}{b-a}])$ be Δu_i which corresponds to the midpoint of the cell, i.e., $\Delta u_i = a + \frac{(2i-1)M}{2(b-a)}$, then the probability of the point is calculated as $q^i_{\Delta u} = \int_{a+(i-1)\frac{M}{b-a}}^{a+i\frac{M}{b-a}} q_{\Delta u}(s)ds$. A hard deadline is derived for each Δu_i, and let it be D_i whose probablity is equal to that of $q^i_{\Delta u}$. Finally, the *pmf* of the hard deadline is derived numerically by multiplying D_i and $q^i_{\Delta u}$, $1 \le i \le M$. The accuracy of the resulting *pmf* depends on a, b, and M, which must be determined by considering the controlled-process state equations, the environment, and the amount of computation.

Although the above method uses a stochastic approach in deriving the *pmf* of the hard deadline, it is still based on the mean values of binomially- and multinomially- distributed random sequences since the hard deadlines of all possible samples cannot be derived due to the excessive number of possible samples. However, the stability of each individual system (i.e., samples) has more practical meaning than the stability of the average system or the ensemble of all possible systems which might be built. Thus, in addition to the deteministic analysis (or combined with the stochastic analysis) of the averaged system stability, we will attempt to stochastically analyze system stability by using the *almost-sure* sample stability concept (which is actually almost deterministic).

Definition 2:

- Probabilistic Stability : For every pair of positive numbers a and b, there exists a positive number $d(a, b, t_0)$ such that

$$P[\sup_{t \ge t_0} \|x_t\| > a] < b \quad \text{for } x_{t_0} \text{ such that } \|x_{t_0}\| < d, \tag{3.9}$$

  where $x_t = \{x(s) : t_0 \le s \le t\}$ is a segment of the past history.

- Almost-Sure Stability: For every pair of positive numbers a and b, there exists a positive number $d(a, b, t_0)$ such that

$$P[\sup_{\|x_{t_0}\| < d} \left(\sup_{t \ge t_0} \|x_t\| \right) > a] < b. \tag{3.10}$$

In fact, the almost-sure sample stability means that almost every possible difference equation for a given ensemble of such systems has a state which is stable in the Lyapunov sense.

3.2 Hard Deadline of the One-Shot Event Model

Although one long-lasting computer failure cannot move the pole position of the stationary model, it may lead to a dynamic failure by driving the system out of the allowed state space when there are critical immediate or terminal constraints on system states [16]. When the effects of erroneous control inputs are included, this phenomenon may be more pronounced than the case of perfect failure detection. Assuming that some computer failure may not be detected upon its occurrence but every detected failure can always be recovered successfully, we can consider three cases for the analysis of the effects of computer failures: (i) delay: when a computer failure is detected upon its occurrence, (ii) disturbance: when a computer failure is not detected till its disappearance, and (iii) disturbance and delay: when a computer failure is detected at some time after its occurrence but before its disappearance.

Let k_0, k_f, N_1, and N_2 denote the indices for the failure occurrence time, the mission completion time, and the period of disturbance, the period of delay measured in sampling periods, respectively, where $N = N_1 + N_2,\ \ 0 \le N_1, N_2 \le N$. The dynamic equation for a one-shot event model is:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\left[\mathbf{u}(k) + (\mathbf{u}(k_0) - \mathbf{u}(k))\Pi_{k_0}(N_1) + \mathbf{u}(k)(\mathbf{I}_{\boldsymbol{\Delta}} - \mathbf{I})\Pi_{k_0+N_1}(N_2)\right], \quad (3.11)$$

where $\Pi_{k_0}(N)$ is the rectangular function as defined in Section 2, and N_1 and N_2 are random variables and determined by the conditional probability of successful detection (d) if N is given:

$$\begin{aligned} \Pr[N_1 = i] &= d(1-d)^i && 0 \le i \le N-1 \\ \Pr[N_2 = i] &= d(1-d)^{N-i} && 1 \le i \le N \\ \Pr[N_1 = N] &= \Pr[N_1 = 0] = 1 - d(1-d)^{N-1}. && \end{aligned} \quad (3.12)$$

Thus, the first moment of Eq. (3.11) is:

$$\bar{\mathbf{x}}(k+1) = \mathbf{A}\bar{\mathbf{x}}(k) + \mathbf{B}\left(\mathbf{u}(k) + \sum_{i=0}^{N}[q_i^d(\mathbf{u}(k_0) - \mathbf{u}(k))\Pi_{k_0}(N_1) + q_i^w\mathbf{u}(k)(\mathbf{I}_{\boldsymbol{\Delta}} - \mathbf{I})\Pi_{k_0+N_1}(N_2)]\right). \quad (3.13)$$

Using Eq. (3.13), one can derive the states at time $k_0 + N$ and k_f, and examine whether or not the state trajectory satisfies the immediate and terminal constraints, iteratively for each N, as was done in [16].

$$\bar{\mathbf{x}}(k_0+N) = \mathbf{A}^N\bar{\mathbf{x}}(k_0) + \sum_{i=k_0}^{k_0+N-1} \mathbf{A}^{k_0+N-i-1}\mathbf{B}[q_i^d\mathbf{u}(k_0) + q_i^w\mathbf{u}(i)\mathbf{I}_{\mathbf{\Delta}}], \tag{3.14}$$

$$\begin{aligned}\bar{\mathbf{x}}(k_f) &= \mathbf{A}^{k_f-k_0}\bar{\mathbf{x}}(k_0) + \sum_{i=k_0}^{k_0+N-1} \mathbf{A}^{k_f-i-1}\mathbf{B}[q_i^d\mathbf{u}(k_0) + q_i^w\mathbf{u}(i)\mathbf{I}_{\mathbf{\Delta}}] + \sum_{i=k_0+N}^{k_f-1} \mathbf{A}^{k_f-i-1}\mathbf{B}\mathbf{u}(i) \\ &= \mathbf{A}^{k_f-k_0-N}\bar{\mathbf{x}}(k_0+N) + \sum_{i=k_0+N}^{k_f-1} \mathbf{A}^{k_f-i-1}\mathbf{B}\mathbf{u}(i). \end{aligned} \tag{3.15}$$

In addition to this deterministic approach, the pmf of the hard deadline that depends on $q_{\Delta u}$ is also derived by using the same stochastic approach employed in the stationary model. Without using the first moment of the state equation (3.13) the probability of the hard deadline being N (i.e., $\Pr[N = D]$) is thus obtained as the sum of the probabilities[4] of the sample equations (3.11) in which a dynamic failure occurs at time N. One must continue this process until a dynamic failure occurs for all samples (i.e., $\Pr[N = D]$=1).

4 Hard Deadlines for Time-Invariant Nonlinear Systems

Nonlinear control systems generally differ from linear systems in two important aspects:

1. It is not always possible to obtain closed-form solutions for nonlinear systems, where the sequences of approximating functions converging to (or estimates for) the true solution are mostly satisfying forms of the solution.
2. The analysis requires more complex and difficult mathematics.

In spite of these difficulties, hard deadlines are derived for nonlinear control systems since the dynamic equations of most control systems consist of nonlinear properties such as nonlinear gain, saturation, deadband, backlash, hysteresis, and nonlinear characteristic curves. The nonlinear differential equation of the continuous-time domain is generally given by:

$$\dot{\mathbf{x}}(t) = \mathbf{h}[t, \mathbf{x}(t), \mathbf{u}(t)], \qquad \mathbf{x}(0) = \mathbf{x}_0. \tag{4.1}$$

Assuming that the function $h(t,\cdot)$ is globally Lipschitz-continuous, the state at the sampling time $(k+1)T_s$ is represented by using a Taylor series:

$$\mathbf{x}((k+1)T_s) = \mathbf{x}(kT_s) + T_s\dot{\mathbf{x}}(kT_s) + \frac{T_s^2}{2}\ddot{\mathbf{x}}(kT_s) + \cdots, \tag{4.2}$$

[4] N^2 testings are required for each N.

where the first-order derivative term can be calculated using Eq. (4.1), i.e., $\dot{\mathbf{x}}(kT_s) = \mathbf{h}[kT_s, \mathbf{x}(kT_s), \mathbf{u}(kT_s)]$. Since higher-order methods require the calculation of many partial derivatives, the first-order method is applied as a useful starting point in understanding more sophsticated methods. Then, the nonlinear discrete-time state equation is approximated by:

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{f}[k, \mathbf{x}(k), \mathbf{u}(k)], \quad \forall k \geq k_0, \tag{4.3}$$

where $\mathbf{f}(\cdot) = T_s\mathbf{h}(\cdot)$ and $\mathbf{x}(k)$ corresponds to $\mathbf{x}(kT_s)$.

4.1 Hard Deadline of the Stationary Model

The effect of the stationary occurrences of computation-time delay due to the failure(s) of the controller computer can also be analyzed by examining the stability of nonlinear systems like linear systems. In this analysis, every failure is assumed to be detected upon its occurrence and call for a recovery mechanism, i.e., $d = 1$. This assumption ignores the effects of erroneous control inputs. However, one cannot simply modify the dynamic equations, nor can he calculate pole positions efficiently. The stability of nonlinear systems is generally analyzed by the Lyapunov's second method. The drawback of this method, which seriously limits its use in practice, is that it is not easy to find the required Lyapunov function and it gives only sufficient conditions for stability or instability. Thus, the Lyapunov's first method, which linearizes the nonlinear system around an equilibrium point and examines the stability of the resulting linearized system, is used for general nonlinear systems.

The state difference equation of a nonlinear control system is assumed to be given as in Eq. (4.3) and a suitable feedback control input is calculated to stabilize the system. Let the control input have been updated at $t = mNT_s$. If the control inputs were not updated for i sampling periods from that time due to a long computation-time delay, where $0 \leq i \leq N$, the corresponding state equations for the group of intervals during which the system failed to update the control inputs become:

$$\begin{aligned}
\mathbf{x}(mN+1) &= \mathbf{x}(mN) + \mathbf{f}[mN, \mathbf{x}(mN), \mathbf{u}(mN)] \\
\mathbf{x}(mN+2) &= \mathbf{x}(mN+1) + \mathbf{f}[mN+1, \mathbf{x}(mN+1), \mathbf{u}(mN)] \\
&= \mathbf{x}(mN) + \mathbf{f}[mN, \mathbf{x}(mN), \mathbf{u}(mN)] + \mathbf{f}[mN+1, \mathbf{x}(mN+1), \mathbf{u}(mN)] \\
&\ \ \vdots \\
\mathbf{x}(mN+i) &= \mathbf{x}(mN) + \sum_{j=0}^{i-1} \mathbf{f}[mN+j, \mathbf{x}(mN+j), \mathbf{u}(mN)]
\end{aligned}$$

$$
\begin{aligned}
\mathbf{x}(mN+i+1) &= \mathbf{x}(mN) + \sum_{j=0}^{i-1} \mathbf{f}[mN+j, \mathbf{x}(mN+j), \mathbf{u}(mN)] \\
&\quad + \mathbf{f}[mN+i, \mathbf{x}(mN+i), \mathbf{u}(mN+i)] \\
&\quad \vdots \\
\mathbf{x}((m+1)N) &= \mathbf{x}(mN) + \sum_{j=0}^{i-1} \mathbf{f}[mN+j, \mathbf{x}(mN+j), \mathbf{u}(mN)] \\
&\quad + \sum_{j=i}^{N-1} \mathbf{f}[mN+j, \mathbf{x}(mN+j), \mathbf{u}(mN+j)]
\end{aligned}
$$

where m is the time index for the groups of N sampling intervals each. Then, we get the following augmented state difference equation:

$$
\mathbf{X}(m+1) = \mathbf{X}(m) + \mathbf{F}_i[mN, \cdots, mN+N-1, \mathbf{X}(m), \mathbf{X}(m+1), \mathbf{U}(m), \mathbf{U}(m+1)], \quad (4.4)
$$

where

$$
\mathbf{F}_i[\cdot] = \begin{bmatrix}
\mathbf{f}[mN, \mathbf{E}_N\mathbf{X}(m), \mathbf{E}_N\mathbf{U}(m)] \\
\mathbf{f}[mN, \mathbf{E}_N\mathbf{X}(m), \mathbf{E}_N\mathbf{U}(m)] + \mathbf{f}[mN+1, \mathbf{E}_1\mathbf{X}(m+1), \mathbf{E}_N\mathbf{U}(m)] \\
\vdots \\
\sum_{j=0}^{i-1} \mathbf{f}[mN+j, \mathbf{E}_j\mathbf{X}(m+1), \mathbf{E}_N\mathbf{U}(m)] \\
\sum_{j=0}^{i-1} \mathbf{f}[mN+j, \mathbf{E}_j\mathbf{X}(m+1), \mathbf{E}_N\mathbf{U}(m)] + \mathbf{f}[mN+i, \mathbf{E}_i\mathbf{X}(m+1), \mathbf{E}_i\mathbf{U}(m+1)] \\
\vdots \\
\sum_{j=0}^{i-1} \mathbf{f}[mN+j, \mathbf{E}_j\mathbf{X}(m+i), \mathbf{E}_N\mathbf{U}(m)] + \sum_{j=i}^{N-1} \mathbf{f}[mN+j, \mathbf{E}_j\mathbf{X}(m+i), \mathbf{E}_j\mathbf{U}(m+1)]
\end{bmatrix},
$$

where $\mathbf{E}_i = [\overbrace{\underbrace{0, \cdots, 0, 1}_{i}, 0, \cdots, 0}^{N}]$.

Using the probabilities of delays $(q_0, q_1 \cdots, q_N)$, we get a final form of the state difference equation:

$$
\mathbf{X}(m+1) = \mathbf{X}(m) + \sum_{i=0}^{N} \xi_i \mathbf{F}_i[mN, \cdots, mN+N-1, \mathbf{X}(m), \mathbf{X}(m+1), \mathbf{U}(m), \mathbf{U}(m+1)] \quad (4.5)
$$

where $\xi_i \in \{0, 1\}$ is a binomially-distributed random variable with parameter q_i, i.e., $\Pr[\xi_i = 1] = q_i$. Then, the first moment of the above equation is:

$$
\bar{\mathbf{X}}(m+1) = \bar{\mathbf{X}}(m) + \sum_{i=0}^{N} q_i \mathbf{F}_i[mN, \cdots, mN+N-1, \bar{\mathbf{X}}(m), \bar{\mathbf{X}}(m+1), \mathbf{U}(m), \mathbf{U}(m+1)]. \quad (4.6)
$$

To derive the maximum value of N which maintains local stability, the stability of Eq. (4.6) is examined by linearizing Eq. (4.6) around an equilibrilum point. Let the equilibrilum point be $\mathbf{0}$ without loss of generality. Then, the linearized form of Eq. (4.6) is:

$$\mathbf{A}_D^2\bar{\mathbf{X}}(m+1) = \mathbf{A}_D^1\bar{\mathbf{X}}(m) + \mathbf{B}_D^1\mathbf{U}(m) + \mathbf{B}_D^2\mathbf{U}(m+1), \tag{4.7}$$

where $\mathbf{F}_i[\mathbf{0},\cdots,\mathbf{0}] = \mathbf{0}$, $\mathbf{A}_D^1 = \mathbf{I} - \left[\frac{\partial\sum_{i=0}^N q_i\mathbf{F}_i}{\partial\bar{\mathbf{X}}(m)}\right]_{\mathbf{X}=\mathbf{0},\mathbf{U}=\mathbf{0}}$, $\mathbf{A}_D^2 = \mathbf{I} + \left[\frac{\partial\sum_{i=0}^N q_i\mathbf{F}_i}{\partial\bar{\mathbf{X}}(m+1)}\right]_{\mathbf{X}=\mathbf{0},\mathbf{U}=\mathbf{0}}$, $\mathbf{B}_D^1 = \mathbf{I} - \left[\frac{\partial\sum_{i=0}^N q_i\mathbf{F}_i}{\partial\bar{\mathbf{U}}(m)}\right]_{\mathbf{X}=\mathbf{0},\mathbf{U}=\mathbf{0}}$, and $\mathbf{B}_D^2 = \mathbf{I} - \left[\frac{\partial\sum_{i=0}^N q_i\mathbf{F}_i}{\partial\bar{\mathbf{U}}(m+1)}\right]_{\mathbf{X}=\mathbf{0},\mathbf{U}=\mathbf{0}}$.

When a state feedback controller — which is also obtained from the linearization method — is used (i.e., $\mathbf{U}(m) = -\mathbf{P}_D\mathbf{X}(m)$ where $\mathbf{P}_D = \frac{\partial\mathbf{G}}{\partial\mathbf{X}(m)}$ and $\mathbf{U}(m) = \mathbf{G}[\mathbf{X}(m)]$), Eq. (4.7) becomes:

$$(\mathbf{A}_D^2 + \mathbf{B}_D^2\mathbf{P}_D)\bar{\mathbf{X}}(m+1) = (\mathbf{A}_D^1 - \mathbf{B}_D^1\mathbf{P}_D)\bar{\mathbf{X}}(m). \tag{4.8}$$

Using Eq. (4.8), we examine local stability by calculating the pole positions of the following characteristic equation:

$$\det\left[(\mathbf{A}_D^2 + \mathbf{B}_D^2\mathbf{P}_D)z^N - (\mathbf{A}_D^1 - \mathbf{B}_D^1\mathbf{P}_D)\right] = 0. \tag{4.9}$$

This method has two limitations: (i) the conclusions based on linearization are purely local, i.e., it is effective only in the vincinity of the equilibrium point, and (ii) if some poles are located on the unit circle and the others are located within the unit circle, the result is inconclusive, which is called a *critical problem*.

4.2 Hard Deadline of the One-Shot Delay Model

The trajectory of Eq. (4.3) is determined by the following equations if the conditions for the existence and uniqueness of solutions for the nonlinear difference equation are met:

$$\mathbf{x}(k) = \mathbf{x}_0 + \sum_{i=0}^{k-1}\mathbf{f}(i,\mathbf{x}(i),\mathbf{u}(i)). \tag{4.10}$$

For the existence of a unique trajectory there must exist finite constants T, r, h, k such that

$$||\mathbf{f}(k,\mathbf{x}(k),\mathbf{u}(k)) - \mathbf{f}(k,\mathbf{y}(k),\mathbf{u}(k))|| \le k||\mathbf{x}-\mathbf{y}||, \quad \forall\mathbf{x},\mathbf{y}\in B,\ \forall k\in[0,K]$$

$$||\mathbf{f}(k,\mathbf{x}_0(k),\mathbf{u}(k))|| \le h, \quad \forall k\in[0,K]$$

where $B = \{\mathbf{x}\in R^n : ||\mathbf{x}-\mathbf{x}_0|| \le r\}$.

Then, Eq. (4.3) has exactly one solution over $[0,\delta]$ whenever

$$h\delta e^{k\delta} \le r \quad\text{and}\quad \delta \le \min[T, \frac{\rho}{h}, \frac{r}{h+kr}] \quad \text{for some constant } \rho < 1.$$

Let k_0, k_f, and N denote the indices of delay/failure occurrence time, the mission completion time, and the period of delay measured in sampling periods, respectively. Then, the dynamic equation of a one-shot delay model for nonlinear control systems described by Eq. (4.1) is:

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{f}\,[\, k, \mathbf{x}(k), \{(\mathbf{u}(k) + (\mathbf{u}(k_0) - \mathbf{u}(k))\Pi_{k_0}(N)\}\,]\,.$$

To test if the constraints at time $k_0 + N$ and k_f are met, one must derive $\mathbf{x}(k_0 + N)$ and $\mathbf{x}(k_f)$ as:

$$\begin{aligned}
\mathbf{x}(k_0+N) &= \mathbf{x}(k_0) + \sum_{i=k_0}^{k_0+N-1} \mathbf{f}(i, \mathbf{x}(i), \mathbf{u}(k_0)) \\
\mathbf{x}(k_f) &= \mathbf{x}(k_0) + \sum_{i=k_0}^{k_0+N-1} \mathbf{f}(i, \mathbf{x}(i), \mathbf{u}(k_0)) + \sum_{i=k_0+N}^{k_f-1} \mathbf{f}(i, \mathbf{x}(i), \mathbf{u}(i)).
\end{aligned}$$

5 Examples and Numerical Results

Example 5.1-1: To show the hard deadline of a linear time-invariant control system in the presence of stationary occurrences of input delays/disturbances due to computer failures, we consider a simple controlled process:

$$\begin{aligned}
x_1(k+1) &= 11.02x_1(k) + 1.08x_2(k) + 3.5u_1(k) \\
x_2(k+1) &= 0.95x_2(k) + 0.5u_1(k) + 1.07u_2(k),
\end{aligned} \tag{5.1}$$

where the coefficient matrices of a quadratic performance index are given as:

$$\mathbf{R}_{xx} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}; \quad \mathbf{R}_{uu} = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}.$$

One can derive the optimal feedback control gain matrix $\mathbf{F}$ that stabilizes the controlled process by solving a discrete Riccati equation as:

$$\mathbf{F} = \begin{bmatrix} 3.1251 & 0.3090 \\ -1.0791 & 0.5512 \end{bmatrix}.$$

This feedback control changes the system eigenvalues from $\{0.95, 11.02\}$ to $\{0.0777, 0.2101\}$. We then derived deterministically the change of poles as a result of iteratively incrementing N for the occurrence of the largest delay possible ($p = q_N = 0.045$). The results are given in Table 1, where the first case represents the perfect coverage ($d = 1$) and the second case

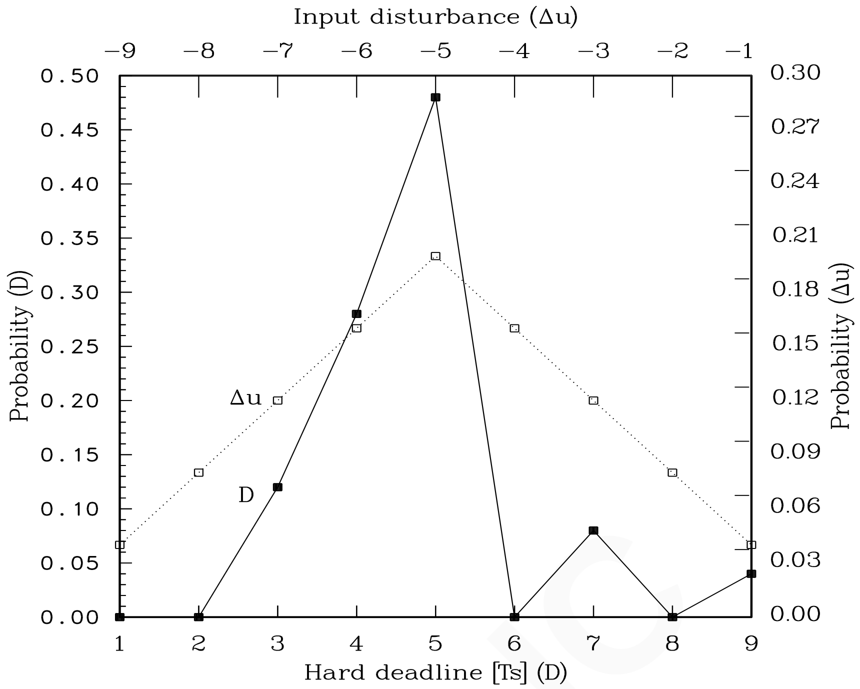

Figure 2: Probability mass functions of Δu, which is given *a priori*, and D, which is derived.

represents the presence of an input disturbance ($d = 0.9$ and $\mu_{\Delta u} = -5$).

N	3	4	5	6	7
$d = 1$	0.7537	0.8945	0.9659	1.0021	1.0204
$d = 0.9$	0.7579	0.9985	1.1698	1.2922	1.3798

Table 1: Maximum magnitude of eigenvalues, $|\lambda_{max}|$.

The deterministic value of the hard deadline is $D = 6T_s$ in the absence of input disturbances under instant failure detection, whereas it decreases to $D = 5T_s$ with some (infrequent) input disturbances. The *pmf* of hard deadline is plotted in Fig. 2 along with the *pmf* of the magnitude of input disturbances.

Example 5.1-2: The hard deadline of a one-shot event model is derived for a double-integrator system which was also used for a one-shot delay model in [16]. The state difference

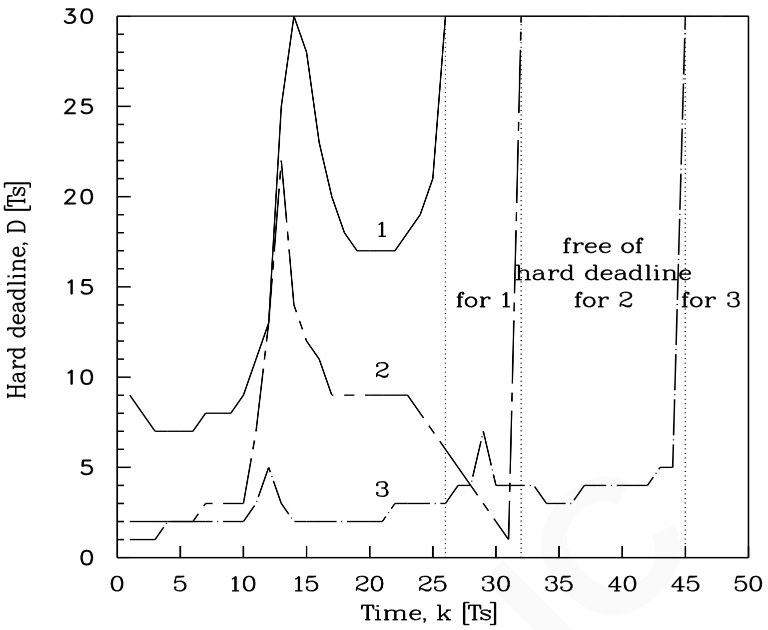

Figure 3: Hard deadlines of one-shot event model in the absence/presence of input disturbances.

equation of the discretized process with sampling rate, $T_s = 0.01s$, is:

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u(k). \tag{5.2}$$

With the same (state/terminal) constaints and the same feedback control input as those in [16], hard deadlines are deterministically derived in the absence (curve 1)/presence (curve 2: $\mu_{\Delta u} = 10$, curve 3: $\mu_{\Delta u} = -10$) of input disturbances, which are shown in Fig. 3.

The *pmf* of hard deadlines at time $T = 15T_s$ is derived for a Gaussian probability density function of Δu, $q_{\Delta u} = \frac{1}{\sqrt{2\pi}10} e^{-\frac{(\Delta u - 10)^2}{200}}$, and is given in Fig. 4. Since the disturbance to the control input is significant, the hard deadline is likely to be small as shown in the calculation except for $D = 21T_s$, which is the hard deadline in the absence of input disturbances.

Example 5.1-3: The hard deadline of a nonlinear control system in the presence of stationary occurrence of delays is derived for a second-order system described by:

$$\begin{aligned} x_1(k+1) &= 3x_1(k) + x_2(k)^2 - \mathrm{sat}(2x_2 + u(k)) \\ x_2(k+1) &= \sin x_1(k) - x_2(k) + u(k), \end{aligned} \tag{5.3}$$

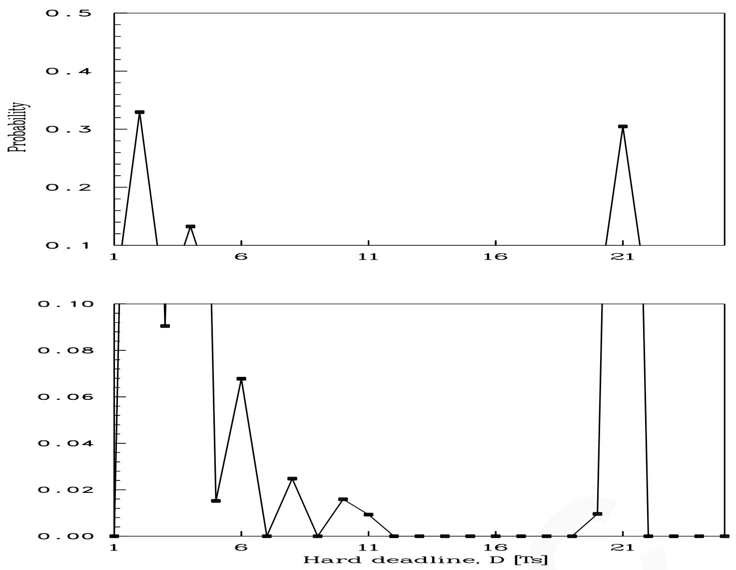

Figure 4: Pmf of hard deadline of one-shot event model when $\mu_{\Delta u} = 10$ and $\sigma_{\Delta u} = 10$.

where the *sat* function is defined as

$$\text{sat}(\rho) = \begin{cases} \rho & |\rho| \leq 1 \\ \text{sign}(\rho) & |\rho| > 1. \end{cases}$$

Then, $\mathbf{f}[k, \mathbf{x}(k), \mathbf{u}(k)]$ in Eq. (4.3) is equal to $[2x_1(k) + x_2(k)^2 - \text{sat}(2x_2 + u(k)), \sin x_1(k) - 2x_2(k) + u(k)]^T$. Hard deadlines around some operating points are given in Table 2, for which the optimal feedback control inputs are calculated by a linearization method, and local asymptotical stability around such operating points is examined by using the eigenvalues of linearized equations in the presence of random occurrence of feedback delays (Lyapunov's first method).

(x_{1o}, x_{2o})	$(0,0)$	$(\pi,0)$	$(\frac{\pi}{2},0)$	$(0,1)$	$(0,-0.24)$
D $[T_s]$	5	2	3	3	7

Table 2: Hard deadlines around some operating points ($u_0 = 0$).

Example 5.1-4: As an example of the one-shot delay models for nonlinear control systems, we consider the brachistochrone problem with an inequality constraint on the admissible state space. Specifically, a particle is falling in a constant gravitational field g for a fixed time t_f with a given initial speed $x_3(t_0) = x_{30}$. Then, we wish to find a path maximizing the final value of the horizontal coordinate $x_1(t_f)$ with unspecified final values of vertical coordinate $x_2(t_f)$ and the velocity $x_3(t_f)$. The continuous-time system dynamic equations, which were treated in [3], are modified by using a certain sampling period T_s to obtain the following difference equations:

$$
\begin{aligned}
x_1(k+1) &= x_1(k) + x_3(k)\cos u(k), & x_1(0) &= 0 \\
x_2(k+1) &= x_1(k) + x_3(k)\sin u(k), & x_2(0) &= 0 \\
x_3(k+1) &= x_3(k) + g\cos u(k), & x_3(0) &= 0.05,
\end{aligned}
\tag{5.4}
$$

where $u(k)$ is the control input to drive the particle to an optimal path at $kT_s(0 \le k \le 100)$, and g is given by 0.02. The state constraint is described by the state variable inequality, $x_2(k) - 0.4x_1(k) - 20 \le 0$, $\forall k$, which is converted to a difference equation by introducing a dummy variable x_4:

$$x_4(k+1) = x_4(k) + [x_2(k) - 0.4x_1(k) - 20]^2 W(x_2(k) - 0.4x_1(k) - 20), \qquad x_4(0) = 0$$

where $W(g) = 0$ if $g \le 0$ and 1 if $g > 0$. The performance index is represented by a modified cost function by including the effect of $x_4(k_f)$ as:

$$J = -x_1(k_f) + \frac{1}{2}Sx_4^2(k_f) \qquad k_f = 100.$$

The optimal control input minimizing J is derived by using the gradient method for multistage decision processes, where the Hamiltonian H and the adjoint equation are defined by the system dynamic equation $\mathbf{f}$ and a new vector λ:

$$
\begin{aligned}
H &= \lambda^T(K+1)\mathbf{f}[\mathbf{x}(k), u(k), k], \\
\frac{\partial H}{\partial \mathbf{x}(k)} &= -\lambda(k) = \frac{\partial \mathbf{f}^T}{\partial \mathbf{x}(k)}\lambda(k+1),
\end{aligned}
$$

where the terminal condition on the adjoint equation is $\lambda^T(k_f) = [-1, 0, 0, Sx_4(k_f)]$. The control input is updated by an iterative equation (for more detailed derivation of $u_{opt}(k)$, see [14]):

$$u^{N+1}(k) = u^N(k) + \Delta u^N(k), \quad \text{where } \Delta u^N(k) = -K(k)\frac{\partial H}{\partial u(k)}. \tag{5.5}$$

The state trajectories during $[0, 100T_s]$ are plotted in Fig. 5, where curve 1 is based on an initial control input ($u(k) = \frac{\pi}{6}$) and curve 2 being close to the optimal path is derived by an optimal control input obtained with 11 iterations of Eq. (5.5) and curve 3 indicates a path of the particle when a long controller-computer failure occurring at $k = 50$ (marked by X). The hard deadlines along curve 2 in the presence of a long one-shot delay are derived as a

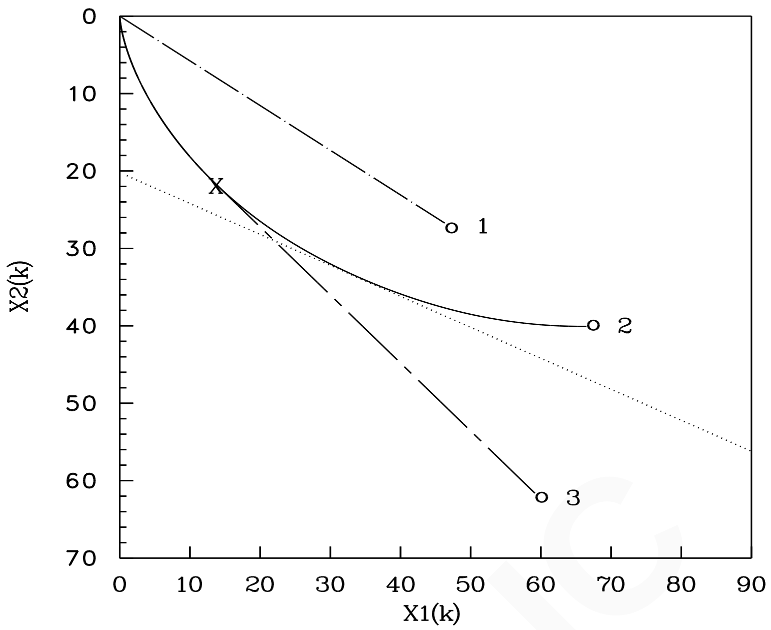

Figure 5: State trajectories of the brachistochrone problem.

function of time index k, which is shown in Fig. 6. As the state gets closer to the boundary of the constraint space, the hard deadline gets reduced significantly ($61 \leq k \leq 75$). When it leaves the boundary by changing the direction, the system (i.e., a falling particle) instantly enters the non-critical region ($76 \leq k$) — which is free of hard deadlines — since the control inputs from that time do not drive the particle close to the constraint space.

The deadline information derived in the previous examples is used for the design and evaluation of controller computers, which are discussed in the following two examples.

Example 5.2-1: When an error is detected the simplest recovery method is to re-execute the previous instruction, called simply *retry*, which is effective in case of immediate error detection [8, 9]. When retrying an instruction, one must determine a retry period, which is long enough for the present fault(s) to die away. If the retry does not succeed in recovering from the error, we have to use an alternative recovery method like rollback or restart. So, the retry period must also be short enough not to miss the deadline by considering the amount of time to be taken by the subsequent recovery method in case of an unsuccessful retry. Let T_t, T_a, and t_r be the "nominal" task execution time in the absence of error, the actual task execution time, and the retry period, respectively. Then,

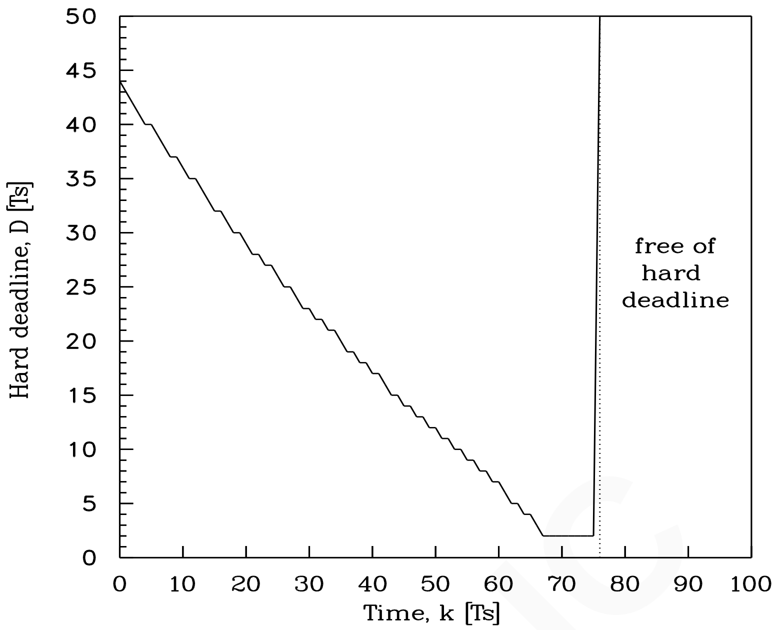

Figure 6: The hard deadline along the optimal path as a function of time.

one can obtain a set of samples of T_a:

$$T_a \in \{T_t, T_t + \frac{1}{\lambda_a}, (\bar{T} + T_s + t_r) + T_t, (\bar{T} + T_s + t_r) + T_t + \frac{1}{\lambda_a}, 2(\bar{T} + T_s + t_r) + T_t, \cdots\}.$$

where T_s, $\bar{T}$, and $\frac{1}{\lambda_a}$ are the resetting time, the mean occurrence time of an error, and the mean active duration of a fault. Since T_a has discrete values, the probability mass function (pmf) of T_a is:

$$\begin{aligned} f_{T_a}^k &= \text{Prob}[T_a = k(\bar{T} + T_s + t_r) + T_t + \delta\frac{1}{\lambda_a}], \quad 0 \le k \le \infty, \quad \delta \in \{0, 1\} \\ &= p_e^{k+\delta}(T_t)(1 - p_s(t_r))^k(1 - p_e(T_t))^{1-\delta} p_s(t_r)^{\delta}, \end{aligned} \tag{5.6}$$

where $p_e(T_t)$ and $p_s(t_r)$ are the probability of the occurrence of an error during T_t (i.e., after restart) and the probability of a successful retry with a retry period t_r.[5] Then, the probability of missing a hard deadline is:

$$p_{mh}(T_t, D) = \int_0^{\infty} \sum_{k > \lfloor (D - T_t)/(\bar{T} + T_s) \rfloor} f_{X_a}^k(x) f_D(y) dy, \tag{5.7}$$

where $f_D(y)$ is the probability density function of the hard deadline. When the hard deadline is deterministic, $f_D(y)$ is a delta function and the corresponding $p_{mh}(T_t, D)$ becomes simpler.

[5] See [7] for the derivation of $p_e(T_t)$ and $p_s(t_r)$ in a TMR system.

Consequently, the optimal retry period can be determined by minimizing $p_{mh}(T_t, D)$ with respect to t_r, using the derived hard deadline information $f_D(y)$.

Similarly, the hard-deadline information is also useful to rollback recovery, where checkpoints must be placed optimally. The checkpoints are usually placed so as to minimize the mean execution cost [22]. However, the mean cost must be minimized while keeping the *probability of dynamic failure* — the probability of missing a deadline [17] — below a prespecified level in a real-time control system [19]. The hard deadline information is necessary to compute the probability of dynamic failure, which can, in turn, be used for the optimal placement of checkpoints.

Example 5.2-2: To have a real sense of using the hard-deadline information, let us consider the Markov reliability model of a Triple Modular Redundant (TMR) controller computer. The TMR controller computer updates the control input to the controlled process (plant) once every T_s seconds. A *TMR failure* is said to occur if more than one processor in the TMR controller fail during T_s. The output of the TMR controller would not be changed in case of a TMR failure, i.e., the control input is not updated. The condition for a system failure resulting from controller-computer failures is derived from the hard deadline, which is the allowable maximum computation-time delay. In other words, this condition gives us the knowledge about the controlled system's resilience against controller-computer failures.

Suppose the hard deadline derived from the controlled system is n sampling periods, where n is a random variable with a probability mass function $f_D(n)$. That is, no dynamic failure occurs if the faults inducing computer failures disappear (or are recovered by a fault-tolerance mechanism) within n sampling intervals. Then, the reliability model is built by extending a Markov chain model with n additional states, as shown in Fig. 7. All samples of the reliability model are obtained by extending the original Markov chain model for all n, whose probabilities are given by $f_D(n)$. The additional states account for the system resilience, i.e., a dynamic failure results only if there are n consecutive incorrect (missing the update of) outputs of the controller computer.

6 Concluding Remarks

Hard deadlines carry important information related to the resilience of the controlled process. Especially, the knowledge of hard deadlines is very useful for modeling system reliability, designing and evaluating fault-tolerant controller computers, which are affected

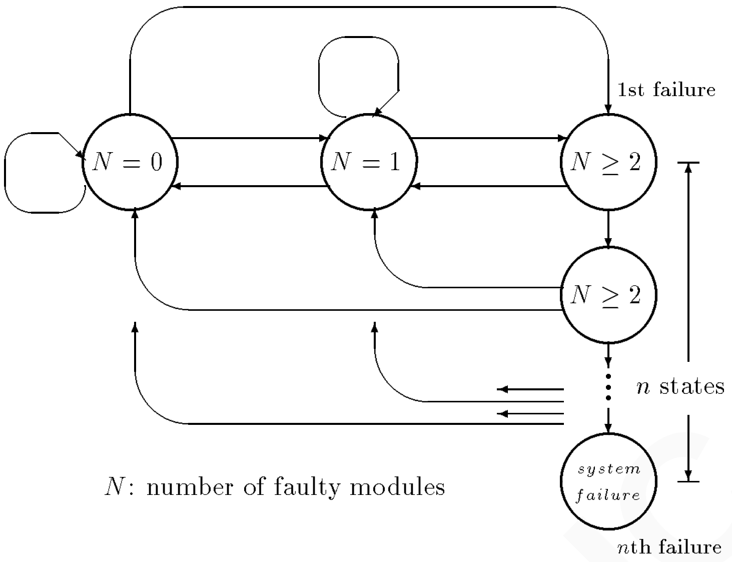

Figure 7: An extended Markov reliability model with a random hard deadline.

by control input disturbances as well as delays caused by faults/interferences. We derived hard deadlines for linear time-invariant control systems in the presence of input disturbances due to imperfect detection and for nonlinear time-invariant control systems by linearizing them around an equilibrium point. Several examples are presented to show the effectiveness of the proposed method in deriving the hard deadlines from the controlled processes and to demonstrate the importance of the deadline information to the design and evaluation of fault-tolerant controller computers.

As an extension of this work, it would be interesting to derive the deadlines of nonlinear time-invariant control systems in the presence of control input disturbances. The derivation of hard deadlines for time-varying systems will also be a challenging task.

Acknowledgement

The authors would like to thank Allan White, Chuck Meissner, Felix Pitts, and Celeste Belcastro of the NASA Langley Research Center, and Jim Smith of the Office of Naval Research for their technical and financial assistance.

References

[1] C. M. Belcastro, "Laboratory test methodology for evaluating the effects of electromagnetic disturbances on fault–tolerant control systems," *NASA TM-101665*, November 1989.

[2] A. P. Belleisle, "Stability of systems with nonlinear feedback through randomly time–varying delays," *IEEE Trans. on Automat. Contr.*, vol. AC–20, no. 1, pp. 67–75, February 1975.

[3] S. Dreyfus, "Variational problems with state variable inequality constraints," *RAND Coporation Paper*, vol. P-2605, pp. 72–85, 1962.

[4] A. Gosiewski and A. W. Olbrot, "The effect of feedback delays on the performance of multivariable linear control systems," *IEEE Trans. on Automat. Contr.*, vol. AC-25, no. 4, pp. 729–734, August 1980.

[5] K. Hirai and Y. Satoh, "Stability of a system with variable time delay," *IEEE Trans. on Automat. Contr.*, vol. AC-25, no. 3, pp. 552–554, June 1980.

[6] G. Hostetter and J. S. Meditch, "Observing systems with unmeasurable inputs," *IEEE Trans. on Automat. Contr.*, vol. AC–18, pp. 306–307, June 1973.

[7] H. Kim and K. G. Shin, "Design and analysis of an optimal instruction-retry policy for TMR controller computer," submitted for publication, 1993.

[8] I. Koren and Z. Koren, "Analysis of a class of recovery procedures," *IEEE Trans. on Computers*, vol. C-35, no. 8, pp. 703–712, August 1986.

[9] Y. H. Lee and K. G. Shin, "Optimal design and use of retry in fault-tolerant computing systems," *Journal of the ACM*, vol. 35, pp. 45–69, January 1988.

[10] G. Leitmann, *An Introduction to Optimal Control*, New York, NY: McGray-Hill, 1969.

[11] M. Mariton, "Detection delays, false alarm rates and the reconfiguration of control systems," *Int. J. Control*, vol. 49, no. 3, pp. 981–992, 1989.

[12] T. Mita, "Optimal digital feedback control systems counting computation time of control laws," *IEEE Trans. on Automat. Contr.*, vol. AC–30, no. 6, pp. 542–548, June 1985.

[13] Z. V. Rekasius, "Stability of digital control with computer interruption," *IEEE Trans. on Automat. Contr.*, vol. AC–31, no. 4, pp. 356–359, April 1986.

[14] A. P. Sage and I. C. C. White, *Optimum systems control*, Prentice–Hall Inc., Englewood Cliffs, New Jersey, 1977.

[15] K. G. Shin and X. Cui, "Effects of computing time delay on real–time control systems," in *Proc. of 1988 American Control Conf.*, pp. 1071–1076, 1988.

[16] K. G. Shin and H. Kim, "Derivation and application of hard deadlines for real–time control systems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1403–1413, Nov./Dec. 1992.

[17] K. G. Shin, C. M. Krishna, and Y.-H. Lee, "A unified method for evaliating real–time computer controller and its application," *IEEE Trans. on Automat. Contr.*, vol. AC–30, no. 4, pp. 357–366, April 1985.

[18] K. G. Shin and Y.-H. Lee, "Error detection process — model, design, and its impact on computer performance," *IEEE Trans. on Computers*, vol. C–33, no. 6, pp. 529–539, June 1984.

[19] K. G. Shin, T.-H. Lin, and Y.-H. Lee, "Optimal checkpointing of real–time tasks," *IEEE Trans. on Computers*, vol. C–36, no. 11, pp. 1328–1341, November 1987.

[20] D. P. Siewiorek and R. S. Swarz, *The Theory and Practice of Reliable System Design*, Digital Equipment Corporation, Bedford, MA, 1982.

[21] D. D. Siljak, "Reliable control using multiple control systems," *Int. J. Control*, vol. 31, no. 2, pp. 303–329, 1980.

[22] A. Tantawi and M. Ruschitzka, "Performance analysis of checkpinting strategies," *ACM Trans. Computer Systems*, vol. 2, pp. 123–1441, 1984.

[23] K. Zahr and C. Slivinsky, "Delay in multivariable computer controlled linear systems," *IEEE Trans. on Automat. Contr.*, vol. AC–19, no. 8, pp. 442–443, August 1974.