Control Eng. Practice, Vol. 1, No. 4, pp. 623-628, 1993
Printed in Great Britain. All rights reserved.

0967-0661/93 $6.00 + 0.00
© 1993 Pergamon Press Ltd

HARD DEADLINES IN REAL-TIME CONTROL SYSTEMS

K.G. Shin and H. Kim

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

Abstract. In a real-time control system, transient computer failures caused by electromagnetic interference(s) may seriously affect system stability due to not updating the control input properly. This paper presents a method for deriving hard deadlines by examining the stability of the state difference equations modified with an assumed maximum delay and several random sequences that represent the effects of stationary occurrences of disturbances to, as well as random delays of, the control input. A one-shot event model, in which a single long-lasting fault causes a dynamic failure, is also presented based on state trajectories and allowed state spaces.

Key Words. Real-time control system; controller computer; electromagnetic interference; computation time delay; control input disturbance; hard deadline; system stability; allowed state-space.

1. INTRODUCTION

Most real-time control systems consist of two synergistic parts: the *controlled process* or *environment*, and the *controller computer*. Digital computers are commonly used in real-time control systems due mainly to their improved performance and reliability in dealing with increasingly complex controlled processes. The control programs, which are executed by a controller computer residing in the feedback loop, perform a set of functions using sensor readings from the controlled process and/or the environment at regular time intervals.

Since the controller computer is highly susceptible to transient electromagnetic interferences inducing functional errors (perhaps without damaging any components), it is usually equipped with some fault-tolerance mechanisms especially for life- or safety-critical systems like aircraft or nuclear reactors. When the abnormality (component failure or environmental interference) of the controller computer occurs, the computation-time delay increases significantly, thus either failing to update the control input during the time taken for error detection, fault location, and recovery; or updating the control input incorrectly until the failure is handled successfully (i.e., detected and recovered). The stationary occurrences of these abnormalities may lead to the loss of system stability if their active duration exceeds a certain limit called the *hard deadline* (Shin *et al.*, 1985). Even one occurrence of the abnormality for a long period — called a *one-shot event model* by Shin and Kim (1992) — may drive the controlled process out of its allowed state space, or a *dynamic failure* occurs.

Most conventional analyses of computation-time delay effects have been based on the assumption that the feedback delay is fixed or constant (Gosiewski and Olbrot, 1980; Hirai and Satoh, 1980). Shin and Kim (1992) derived hard deadlines for linear time-invariant control systems based on the fact that computation-time delays are stochastic in both their occurrence times and magnitudes reflecting the nature of computer failures. However, Shin and Kim (1992) did not consider control input disturbances under the assumption of perfect fault detection.

In this paper, the hard deadlines are derived by examining the stability of the state difference equations modified with random sequences that represent the stationary occurrences of computer failures and the imperfect error coverage (with binomial distributions), the duration of failures/interferences (with multinomial distributions), and the magnitude of disturbances to

The work reported was supported in part by the Office of Naval Research under Grant N00014-91-J-1115 and by the NASA under Grant NAG-1-1120.

the control input (with a normal distribution). The system dynamics are modified first according to the assumed maximum delay, NT_s, and the probability distribution of delays whose occurrence periods $\leq NT_s$, where N is changed from 1 to the actual maximum delay (or hard deadline), denoted by DT_s. In addition, the state and input constraints are used to derive the allowed state space from which the hard deadline is derived as a function of time and the system state. This analysis is useful for the one-shot event model, where a single event — a long-lasting failure — may cause a dynamic failure.

Section 2 addresses the generic problem for analyzing the effects of computation-time delays and input disturbances. Section 3 presents the basic assumptions and the random sequences that characterize controller failures and input disturbances. Then, hard deadlines are derived with the modified state difference equations for linear time-invariant control systems for both stationary and one-shot event models. In Section 4, simple linear systems are examined to demonstrate the proposed approach. Section 5 deals with the application of the hard-deadline information to the design of a reliable controller computer. The paper concludes with Section 6.

2. EFFECTS OF CONTROLLER COMPUTER FAILURES

Linear time-invariant controlled processes are generally represented by state-space models as shown in (1) and are equipped with well-designed controllers that stabilize the overall control system and optimize the control objectives:

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

where k is the time index, one unit of time represents the sampling interval T_s, and $\mathbf{x} \in \mathcal{R}^n$ and $\mathbf{u} \in \mathcal{R}^\ell$ are the state and input vectors, respectively. The (digital) controller computer reads sensors, and calculates the control input once every T_s seconds according to a programmed control strategy. The control input, which is held constant within each sampling interval by a latch circuit, is applied to the controlled process.

When a fault[1] occurs in the controller computer, it will trigger functional error modes — which are computer failures — perhaps, without component damages. Suppose a computer failure is detected upon its occurrence at time k_0, and its recovery takes n sampling intervals. The control inputs during this period (i.e., from time k_0 to $k_0 + n$) will be held constant at $\mathbf{u}(k_0)$ by the D/A converter and latch circuits. Suppose a computer failure is detected n_1 sampling intervals after its occurrence at time k_0 and the subsequent recovery takes n_2 sampling intervals. The control inputs during this period are: (i) $\mathbf{u}(k_0)\mathbf{I}_\Delta, \mathbf{u}(k_0+1)\mathbf{I}_\Delta, \cdots, \mathbf{u}(k_0+n_1)\mathbf{I}_\Delta$ from time k_0 to $k_0 + n_1$, (ii) held constant at $\mathbf{u}(k_0 + n_1)\mathbf{I}_\Delta$ from time $k_0 + n_1 + 1$ to $k_0 + n_1 + n_2$, and (iii) $\mathbf{u}(k_0+n_1+n_2+1), \mathbf{u}(k_0+n_1+n_2+2) \cdots$ from time $k_0 + n_1 + n_2 + 1$, where $\mathbf{I}_\Delta$ is a diagonal matrix with $Diag[\mathbf{I}_\Delta]_i = 1 + \Delta u_i$ and Δu_i is a random sequence modeled as the output of a dynamic system with a white-noise input. Since faults occur randomly during the mission lifetime, they are considered to be random disturbances to the controlled process, which can be modeled based on the fault characteristics.

The hard deadline of a stationary model is defined as the maximum duration of the controller computer's failure without losing system stability. Thus, in linear time-invariant systems, the hard deadline is defined as:

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

where $\lambda(N)$ is an 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 computer failures. Let $\mathbf{X}_A(k)$ and $\mathbf{U}_A$ be the allowed state space at time kT_s and the admissible input space, respectively. Suppose the state is evolved from time k_0 in the presence of a computer failure (disturbance/delay) which occurred at k_1T_s, was detected N_1 sample intervals later and is recovered within N_2 sample intervals of its detection, where $N = N_1 + N_2$, $0 \leq N_1, N_2 \leq N$. The control input during this period ($k_1 \leq k \leq k_1 + N$) is:

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

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

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

where the state trajectory is governed by:

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

3. DERIVATION OF HARD DEADLINES

[1] Transient electromagnetic interferences (EMI) such as lighting, high intensity radio frequency fields (HIRF) and nuclear electromagnetic pulses (NEMP) are considered as the possible sources of the fault.

Let NT_s and DT_s be the assumed maximum and actual maximum delays, respectively. Then, the hard deadline can be obtained by iteratively testing the necessary conditions for system stability and state residence in the allowed state space while changing N from 1 to D.

3.1. Hard Deadline of Stationary Model

The controlled processes described by (1) are unstable without any state feedback control. Thus, the state feedback control input necessary to stabilize such unstable systems can be calculated in the usual form of $\mathbf{u}(k) = -\mathbf{F}\mathbf{x}(k)$, depending on the control objectives, e.g., time-optimal control with energy constraint, optimal state tracking, and optimal linear regulator.

To derive hard deadlines, the given state equation is modified to include all the stochastic behaviors of computer failures based on the following random sequences and the basic assumptions for tractability.

Definition of Random Sequences:

1. p: the probability of a computer failure at each sampling instant.
2. d: the conditional probability of successful failure detection given that a computer failure had occurred.
3. q_i^d: the conditional probability of a delay (recovery duration) for i sampling intervals ($\sum_{i=1}^{N} q_i^d = 1$) if a computer failure occurred and is detected before generating any incorrect control input.
4. q_i^w: the conditional probability of a control input disturbance for i sampling intervals ($\sum_{i=1}^{N} q_i^w = 1$) if a computer failure occurred and is not detected till its disappearance.
5. $q_{\Delta u}$: the probability density function (pdf) of the magnitude, Δu, of the control input disturbance at time kT_s, i.e., $\mathbf{u}_{actual}(k) = \mathbf{u}_{desired}(k)\mathbf{I}_{\Delta}$. The mean and variance of $q_{\Delta u}$ are given *a priori* as $\mu_{\Delta u}$ and $\sigma^2_{\Delta u}$.

Basic Assumptions:

1. The control inputs calculated after recovering computer failures are always correct.
2. The probability that two transient failures occur sequentially within a small number, $N - i$, of sample intervals, where the delay (recovery duration) or duration of incorrect control inputs (active duration of a transient failure) is i sample intervals and N is the assumed maximum value of such intervals, i.e., $1 \leq i \leq N$ — is small enough to be ignored. That is, we consider only one computer failure possible during N sample intervals.
3. Every random sequence considered here is independent identically distributed ($i.i.d$) with respect to the time index k.

Suppose the control input has been updated correctly at $t = mNT_s$. In case an abnormality (delay/disturbance) is active for i sampling intervals since time $t = mNT_s$ as a result of a controller computer failure, where $1 \leq i \leq N$, let the control input at $(mN+i)T_s$ be denoted as $\mathbf{u}_a(mN+i)$ which is equal to either $\mathbf{u}(mN+i)\mathbf{I}_{\Delta}$ for disturbance or $\mathbf{u}(mN)$ for delay. The corresponding state equations for the group of intervals during which the system failed to update the control input correctly become:

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

where m is the time index for the 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 the group of sampling intervals of NT_s. Then, the following augmented state equations are obtained:

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

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, $[\mathbf{B}^{d1}_{D}, \mathbf{B}^{d2}_{D_i}]$ for delay, and

$[\mathbf{B}_{D_i}^{w1}, \mathbf{B}_{D_i}^{w2}]$ for disturbance, respectively. In the above equations, $\mathbf{A}_D, \mathbf{B}_{D_0}^{nk}, \mathbf{B}_{D_0}^{wk}, \mathbf{B}_D^{d1}$, and $\mathbf{B}_{D_i}^{d2}$ are the augmented state and input transition matrices, where $k \in \{1,2\}$, and $\mathbf{F}_D$ is an augmented feedback gain matrix. Then, the state difference equation is modified to:

$$\mathbf{X}(m+1) = \mathbf{A}_D\mathbf{X}(m) + \left((1-\psi)\mathbf{B}_{D_0}^{n1} + \psi(1-\varphi)\mathbf{B}_D^{w1} + \psi\varphi\sum_{i=1}^{N}\xi_i\mathbf{B}_{D_i}^{d1}\right)\mathbf{U}(m) + \left(((1-\psi)\mathbf{B}_{D_0}^{n2} + \psi(1-\varphi)\sum_{i=1}^{N}\zeta_i\mathbf{B}_{D_i}^{w2} + \psi\varphi\sum_{i=1}^{N}\xi_i\mathbf{B}_{D_i}^{d2}\right)\mathbf{U}(m+1) \qquad (2)$$

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^w, q_i^d, i.e., $\Pr[\xi_i = 1] = q_i^w$.

Similarly to the method used by Shin and Kim (1992), the deterministic value of the hard deadline is determined by examining the pole positions of the first moment (ensemble average) of (2). Although the resulting hard deadline has little practical meaning, it indicates the trend of the ensemble system behavior with the uncertainty (in the state and output) that can be measured by the second moment of (2). In addition, one 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 $q_{\Delta u}$. The mapping between the hard deadlines and the magnitudes of disturbances (Δu's) is not one–to–one and hard deadlines can be derived iteratively using a numerical method with respect to each sample value of Δu's. The sample values of Δu's are obtained by uniformly quantizing the $q_{\Delta u}$ continuum of 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), where a, b, and M depend on the required accuracy of analysis. Then, points are allocated to the quantized intervals (cells). Let the point of the ith cell ($[a + (i-1)\frac{M}{b-a}, a + i\frac{M}{b-a}]$) be Δu_i which corresponds to the value of the midpoint of the cell, i.e., $\Delta u_i = a + \frac{(2i-1)M}{2(b-a)}$, then the probability of being at this point is calculated as $q_{\Delta u}^i = \int_{a+(i-1)\frac{M}{b-a}}^{a+i\frac{M}{b-a}} q_{\Delta u}(s)ds$. A hard deadline is derived for each Δu_i, and let it be D_i whose probablity is equal to that of the ith cell (i.e., $q_{\Delta u}^i$). Finally, the pmf of the hard deadline is derived numerically by multiplying D_i and $q_{\Delta u}^i$, $1 \leq i \leq M$.

3.2. One–Shot Event Model

The pole locations do not change in case of only one single failure with a relatively long ($> T_s$) active period. The (asymptotic or global) stability condition discussed thus far is therefore no longer applicable. Instead, the terminal state constraints can be used to test whether or not the system leaves its allowed state space. Note that every critical process must operate within the state space circumscribed by given constraints, i.e., the allowed state space. When the control input is not updated for a period exceeding the hard deadline, the system may leave the allowed state space, thus causing a dynamic failure. The allowed state space consists of two sets of states $\mathbf{X}_A^1$ and $\mathbf{X}_A^2$ defined as follows:

- $\mathbf{X}_A^1$: the set of states in which the system must stay to avoid an *immediate* dynamic failure, e.g., a civilian aircraft flying upside down is viewed as an immediate dynamic failure. This set can usually be derived *a priori* from the physical constraints.
- $\mathbf{X}_A^2$: the set of states that can lead to meeting the terminal constraints with appropriate control inputs. This set is determined by the terminal constraints, the dynamic equation, and the control algorithm used.

The system must not leave $\mathbf{X}_A^1$ nor $\mathbf{X}_A^2$ in order to prevent catastrophic failure.

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 intervals, respectively, where $N = N_1 + N_2$, $0 \leq N_1, N_2 \leq N$. The dynamic equation of a one-shot event model is:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\left[\mathbf{u}(k) + (\mathbf{u}(k_0) - \mathbf{u}(k))\Pi_{k_0}(N_1) + \mathbf{u}(k)(\mathbf{I}_\Delta - \mathbf{I})\Pi_{k_0+N_1}(N_2)\right] \quad (3)$$

where $\Pi_{k_0}(N)$ is a rectangular function defined in Section 2, and N_1 and N_2 are random variables whose probabilities are determined by q_i^d and q_i^w. Then, the deterministic value or the pmf of hard deadline can be derived, similarly to the stationary model, by using the first moment or the samples of (3).

The state trajectory, which is tested for the given constraints, can be obtained by using the first moment or the samples of (3). $\mathbf{x}_f \in \mathbf{X}_A^f$ can be tested indirectly by the following relation:

$$\mathbf{x}(k_f) \in \mathbf{X}_A^f \Longleftrightarrow \mathbf{x}(k_0 + N) \in \mathbf{X}_A^2,$$

where

$$\mathbf{X}_A^2 = \{ \mathbf{x} \mid [\mathbf{A}^{k_f-k_0-N}\mathbf{x} + \textstyle\sum_{i=k_0+N}^{k_f-1} \mathbf{A}^{k_f-i-1}\mathbf{B}\mathbf{u}(i)] \in \mathbf{X}_A^f \}.$$

Table 1 $|\lambda_{max}|$

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

In practice, it is difficult to obtain $\mathbf{X}_A^2$. Although there may be a one-to-one mapping between $\mathbf{x}(k_0+N)$ and $\mathbf{x}(k_f)$, $\mathbf{X}_A^f$ is usually a continuum, which requires an excessive amount of computation due to the curse of dimensionality. The size of $\mathbf{X}_A^2$ will decrease as either N increases or k approaches k_f, but the size of $\mathbf{X}_A^2$ is usually larger than that of $\mathbf{X}_A^f$ due to the (asymptotic) stability of a controlled process.

4. EXAMPLES

To demonstrate the concept of hard deadline, the hard deadlines are derived for several simple example control systems, two of which are described for the stationary and one-shot event models.

Example 1: Consider a simple controlled process:

$$x_1(k+1) = 11.02x_1(k) + 1.08x_2(k) + 10u_1(k)$$
$$x_2(k+1) = 0.95x_2(k) + 10u_2(k),$$

where the coefficient matrices of a quadratic performance index and the corresponding optimal (feedback) control gain matrix stabilizing the controlled process by discrete Riccati equation are given by:

$$\begin{aligned} \mathbf{R}_{xx} &= 2\mathbf{R}_{uu} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \quad \text{and} \\ \mathbf{F} &= \begin{bmatrix} 3.1251 & 0.3090 \\ -1.0791 & 0.5512 \end{bmatrix}. \end{aligned}$$

This feedback control changes the eigenvalues from $\{0.95, 11.02\}$ to $\{0.0777, 0.2101\}$. Then, the change of poles as a result of incrementing N is derived deterministically for the occurrence of the largest delay possible ($p = q_N = 0.045$) and is given in Table 1, where the first case is for the perfect coverage ($d = 1$) and the second case represents the existence of input disturbances ($d = 0.9$ and $\mu_{\Delta u} = -5$). The deterministic value of the hard deadline is $D = 6T_s$ in the absence of input disturbances with an instant failure detection, whereas it decreases to $D = 5T_s$ with some (infrequent) input disturbances. The *pmf* of hard deadline is given in Table 2 with the *pmf* of the magnitude of disturbances to the control input.

Table 2 *Pmf* of D and $q_{\Delta u}$

D	3	4	5	7	9
Pr[D]	0.12	0.28	0.48	0.08	0.04
Δu	-5 ± 4	-5 ± 3	-5 ± 2	-5 ± 1	-5
$q_{\Delta u}$	0.0667	0.1334	0.2	0.2667	0.3334

Example 2: The hard deadline of a one-shot event model is derived for the system of a double integrator which was also used for a one-shot delay model by Shin and Kim (1992). The state difference equation of the discretized process with the sampling rate, $T_s = 0.01s$, is:

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

With the same (state/terminal) constaints and the same feedback control input as those of Shin and Kim (1992), the *pmf* of hard deadline 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 Table 3.

Table 3 *Pmf* of hard deadline

D	2	3	4	5	6
Pr[D]	0.3295	0.0905	0.1326	0.0152	0.0678
D	8	10	11	20	21
Pr[D]	0.0248	0.0159	0.0093	0.0096	0.3048

5. APPLICATION OF HARD DEADLINE INFORMATION

The information on hard deadlines is very useful for modeling system reliability and designing both the hardware and software of a controller computer. When designing a controller computer, one has to make many design decisions in the context of controlled processes that are characterized by their hard deadlines and cost functions (Shin *et al.*, 1985), including:

- hardware design issues dealing with the number of processors and the type of interconnection network to be used, and the synchronization of processors,
- software design issues related to the implementation of control algorithms, task assignment and scheduling, redundancy management, error detection and recovery.

From the hard-deadline information, one can deduce the knowledge of system inertia, which can, in turn, be used to specify the fault-tolerance requirement of a real-time control system. This knowledge is required to estimate the system's ability of meeting timing constraints in the presence of controller-computer failures, which was characterized as the ***probability of dynamic failure***, P_{dyn} by Shin *et al.* (1985).

To illustrate the application of the knowledge of system inertia, consider a simple example of a triple modular redundant (TMR) controller computer in which three identical processors execute the same set of cyclic tasks. The TMR controller computer updates, once every T_s seconds or every sampling period, the control input to the controlled process (plant). That is, the period of each cyclic task is equal to T_s. The input of the cyclic task is a discretized output of the plant and the output of the cyclic task will be used to control the plant during the next sampling interval. The output of the TMR controller is correct for each task only if at least two of the three processors in the TMR controller produce correct outputs. A *TMR failure* is said to occur if more than one processor in the TMR controller fail during T_s. Thus, the output of the TMR controller would not be changed in case of a TMR failure. The condition for a system (dynamic) failure resulting from controller-computer failures[2] is derived from the hard deadline, which is the allowable maximum computation-time delay. In other words, this condition gives the knowledge about the controlled system's inertia against controller-computer failures.

More than 90% of computer failures have been reported to be transient, especially with short active durations. Thus, the controller computer may recover from most failures in a few sampling intervals, and it can correctly update the control input without causing any dynamic failure, if the active duration of controller-computer failure is smaller than the hard deadline.

Suppose the hard deadline derived from the controlled system is three sampling periods and a TMR controller computer is used. That is, no dynamic failure occurs if the faults inducing computer failures disappear (or are recovered by a fault-tolerance mechanism) within three sampling intervals. Then, the reliability model for this controller computer is built by extending a Markov chain model with two additional states before the state of a dynamic failure, where the parameters of the Markov chain model are to be estimated at a given level of confidence from empirical data. The additional states account for the system inertia, i.e., a dynamic failure results from only three consecutive incorrect (missing the update of) outputs of the controller computer or for a period of $3T_s$, not immediately from one or two incorrect (missing the update of) outputs. Without the information of hard deadline, one can over-estimate the probability of a system failure under the assumption that the system has no delay-tolerance, i.e., one incorrect output can lead to a dynamic failure.

6. CONCLUSION

The hard deadline for a critical control task is usually assumed to be given *a priori*. This presupposes the existence of a precise definition of the hard deadline and a method to derive it, which, however, have not been addressed in detail. In this paper, the hard deadlines were derived for linear time-invariant control systems in the presence of input disturbances due to imperfect detection coverage based on consideration of the intrinsic nature of computer failures. The knowledge of hard deadlines, which must be derived from real control applications, is very important for task assignment and scheduling, specification and evaluation of fault-tolerant controller computers.

Acknowledgement. The authors would like to thank Allan White, Chuck Meissner, Felix Pitts of NASA Langley Research Center and James Smith of the Office of Naval Research for their support of the work reported in this paper.

7. REFERENCE

Gosiewski, A. and A .Olbrot (1980). The effect of feedback delays on the performance of multivariable linear control systems. In: *IEEE Trans. on Automat. Contr.*, Vol. AC-25, No. 4, pp. 729–734, August.

Hirai, K. and Y. Satoh (1980). Stability of a system with variable time delay. In: *IEEE Trans. on Automat. Contr.*, Vol. AC-25, No. 3, pp. 552–554, June.

Shin, K.G. and H. Kim (1992). Derivation and application of hard deadlines for real–time control systems. In: *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 22, No. 6, pp. 1403-1413, Nobember.

Shin, K.G., C.M. Krishna, and Y.-H. Lee (1985). A unified method for evaliating real–time computer controller and its application. In: *IEEE Trans. on Automat. Contr.*, Vol. AC–30, No. 4, pp. 357-366, April.

[2] The other sources of system failure(s), such as failures in actuators or sensors or mechanical parts and failures of A/D and D/A converters, are not considered in this paper, because its main intent is to analyze the coupling between a controlled process and a (fault-tolerant) controller computer.