ON THE EFFECT OF TRANSIENT DATA-ERRORS IN CONTROLLER IMPLEMENTATIONS

Magnus Gäfvert **Björn Wittenmark**

Department of Automatic Control
Lund Institute of Technology, Sweden
{magnus|bjorn}@control.lth.se

Örjan Askerdal

Department of Computer Engineering
Chalmers University of Technology, Sweden
askerdal@ce.chalmers.se

Abstract Computer-level faults leading to data errors in computations are predicted to occur increasingly frequent in future microprocessors. This work discusses the impact of such errors on closed-loop performance in implementations of digital control systems. A method to render a control system more robust to data errors by introducing artificial signal limits and then combine them with an anti-windup scheme is presented and exemplified.

1. Introduction

As computers, rather than electro-mechanical systems, are increasingly used for implementing control algorithms, control systems become more vulnerable to computer level failures due to faults in semiconductor devices. Such faults are classified as: (i) *transient faults* which are short-duration faults that are induced by neutron and alpha particles, power supply and interconnect noise, and electrostatic discharge, (ii) *intermittent faults* which are re-occurring short-term faults that occur due to marginal hardware or aging effects, and (iii) *permanent faults* which have the same causes as the intermittent faults, but reflect irreversible physical changes. The trend with decreasing transistor and interconnect dimensions, lower power supply, and higher operating frequencies contributes to increasing the occurrence rates of transient and intermittent faults, while improvements in design and manufacturing have led to decreasing permanent failure rates, less than 15 FIT (Failures In Time, or failures in 10^9 hours of operation) for microprocessors in 2001 [4]. This paper will therefore focus on transient faults.

A microprocessor is built from SRAM memory cells, latches, and combinational logic. Faults in the microprocessor may either result in control-flow errors, in which case the instruction execution order is erroneous, or in data errors, in which case the executing program delivers erroneous results. Many control-flow errors may be detected with watchdog timers. Modern microprocessors have built-in protection against transients faults in memory cells (cache) using e.g. error-correcting codes and parity checks, while the remaining parts are essentially unprotected. Methods to implement protection against transient faults in other locations than memory cells often involve redundant micro-controller subsystems (ALU etc.) as in [15], but they result in more complex and more expensive devices. Hence, remaining failures are data errors due to transient faults in the combinational logic or latches. In [14] it is predicted that the transient fault rate per chip of combinational logic circuits will increase nine orders of magnitude from 1992 ($\sim 10^{-7}$ FIT) to 2011 ($\sim 10^2$ FIT), when the failure rate will be comparable to that of unprotected memory elements. The corresponding fault rate for the present generation of microprocessors is ~ 1 FIT. The trend of increasing fault rates is also reported in [8], where it is predicted that a 32 Mbit static memory implemented in a 0.1 μm process will fail on the average each 5.7 years at sea level. A result that is expected to hold also for other logic circuits, such as flip-flops, latches, registers, and combinational circuits.

In safety-critical applications it is imperative to have error detection and recovery functionality to meet the high demands on dependability. In some areas, such as the aircraft industry, fault-tolerance is achieved by redundant hardware and high-end devices. In more cost-sensitive areas, such as the automotive industry, expensive solutions may not be feasible. Assuming a microprocessor with a (constant) transient failure rate of 10 FIT that hosts e.g. a braking functionality in a car, the mean time to computer failure (possibly leading to catastrophic failure for the vehicle) is more than 11,415 years for a single vehicle. For a series of 100,000 cars, the mean time to failure in one of the cars is 42 days. If a microprocessor executes a control algorithm that is likely to tolerate a transient data error, this device may not necessarily have to be as fault-tolerant, and can thus be less expensive. Therefore it is of interest to investigate the impact of transient data errors on hardware hosting implementations of safety-critical control algorithms.

Understanding the effect of data errors on general computer functionality is an intensively researched area, e.g., [12]. Methods for analyzing the effects of timing errors on control systems were presented in, e.g., [9, 19]. Analysis of effects on system stability of data errors caused by EMI bursts was in-

vestigated in [10]. However, catastrophic failures in a safety-critical system may occur before the system reaches instability, e.g. if some constraint on the control error is exceeded. Recent results [5, 18] show that many data errors will have a limited effect on control performance, i.e., control systems often have an inherent resilience or inertia to data errors. This path is pursued further in this paper. It is discussed how the implementation of the control algorithms influences the tolerance to data errors resulting from computer node failures. Related work is published in [2], where classical frequency-domain methods are used to investigate the effects of computer failures on linear feedback control systems.

The paper is organized as follows: In Section 2 a general model for the effect of data errors on a control algorithm is stated. Thereafter, controller realization and scaling is discussed in Section 3. Sections 4 and 5 show how computed signal bounds together with an anti-windup scheme may improve the recovery from data errors. This is illustrated with an example in Section 6. The results are discussed in Section 7, and summarized in Section 8.

2. Modeling Data Errors Caused by Computer Node Faults

Let

$$\begin{aligned} z(k+1) &= \Phi_c z(k) + \Gamma_c^{u_c} u_c(k) + \Gamma_c^{y} y(k) \\ u(k) &= C_c z(k) + D_c^{u_c} u_c(k) + D_c^{y} y(k) \end{aligned} \tag{1}$$

be a state-space realization of a general linear two-degrees-of-freedom dicrete-time controller with internal state z, command signal u_c, process output y, and control signal u. The controlled process is assumed to be a linear time-invariant system

$$Y(s) = G(s)U(s) \tag{2}$$

with zero-order-hold sampling. A pseudo-code implementation of (1) would be

```
repeat:
   uc := read_from_interface();
   y := read_from_sensors();
   u := compute_control_signal(z,uc,y);
   write_to_actuator(u);
   z := update_controller_state(z,uc,y);
   wait_for_next_sample;
```

This algorithm would be implemented and executed in a computer node like in Figure 1. Erroneous computation results due to transient data errors would eventually be stored in u or z, and would propagate to the controlled process, and through the feedback loop. As illustrated in the figure, the internal components of a computer node are: communication controllers, memories, microprocessors, and internal communication buses. All these components may be affected by faults. As the communication controller handles the information exchange with other nodes, quantities measured by sensors may

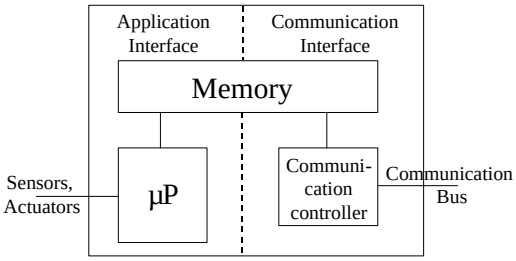

Figure 1 A General architecture of a computer node.

become erroneous due to faults in the communication controller. The data path of a microprocessor consists of caches, registers, buses, and functional units (i.e., ALU, multiplier, etc.). If a fault occurs in any of these parts, it will affect the ongoing calculation if the faulty part is activated, rendering the result of that calculation to be incorrect. The communication inside the computer node is, generally, performed using buses. If a fault occurs on such a bus, the data currently being transferred will be affected and the result may be a data error in any of the calculated data.

Data errors may be regarded as bit-flips in the digital representation of the affected variable or signal. Single-bit errors tend to be more common than multiple-bit errors [1]. In a N-bit fixed-point representation with M fractional bits the number range is $R = [-2^{N-M-1}, 2^{N-M-1} - 2^{-M}]$, with a resolution of $Q = 2^{-M}$. With this representation, the magnitudes of the errors will be in the same range as the control signals (assuming that the data have been properly scaled). In the IEEE floating-point standard with f fraction bits, and e exponent bits the range is $R = [-(2-2^{-f})\cdot 2^{2^{e-1}-1}, (2-2^{-f})\cdot 2^{2^{e-1}-1}]$, and bit flips in the most significant exponent bits may lead to very large errors. A reasonable model of data errors is additive impulse disturbances with magnitude of rectangular distribution within the number representation range. The data error disturbances may then be included in (1) as

$$\begin{aligned} z(k+1) &= \Phi_c z(k) + \Gamma_c^{u_c} u_c(k) + \Gamma_c^{y} y(k) + \eta_z(k) \\ u(k) &= C_c z(k) + D_c^{u_c} u_c(k) + D_c^{y} y(k) + \eta_u(k) \end{aligned} \tag{3}$$

where $\eta_z(k)$ and $\eta_u(k)$ are impulse disturbances due to data errors affecting the computation of the controller state and control signal, respectively. Since transient data errors occur sporadically, it may be assumed that an error manifests in either $\eta_z(k)$ or $\eta_u(k)$.

3. Controller Realizations

With a change of state variables $\bar{z}(k) = Tz(k)$ in (3) the controller realization becomes

$$\begin{aligned} \bar{z}(k+1) &= \bar{\Phi}_c \bar{z}(k) + \bar{\Gamma}_c^{u_c} u_c(k) + \bar{\Gamma}_c^{y} y(k) + \eta_z(k) \\ u(k) &= \bar{C}_c \bar{z}(k) + D_c^{u_c} u_c(k) + D_c^{y} y(k) + \eta_u(k) \end{aligned} \tag{4}$$

with $\bar{\Phi}_c = T\Phi_c T^{-1}$, $\bar{\Gamma}_c^{u_c} = T\Gamma_c^{u_c}$, $\bar{\Gamma}_c^{y} = T\Gamma_c^{y}$ and $\bar{C}_c = C_c T^{-1}$. Note that η_z is not transformed in (4), since it appears

internally in the controller implementation. Combining the controller realization (4) with a state space realization of the sampled controlled process (2)

$$\begin{aligned} x(k+1) &= \Phi x(k) + \Gamma u(k) \\ y(k) &= Cx(k) \end{aligned} \tag{5}$$

results in the closed-loop dynamics

$$\begin{aligned} \xi(k+1) &= \Phi_{cl}\xi(k) + \Gamma_{cl}^{u_c} u_c(k) + \Gamma_{cl}^{\eta_z}\eta_z(k) + \Gamma_{cl}^{\eta_u}\eta_u(k) \\ y(k) &= C_{cl}^{y}\xi(k) \\ u(k) &= C_{cl}^{u_c}\xi(k) + D_c^{u_c}u_c(k) + \eta_u(k) \end{aligned} \tag{6}$$

with $\xi = (x, \bar{z})$ and

$$\Phi_{cl} = \begin{pmatrix} \Phi + \Gamma D_c^y C & \Gamma C_c T^{-1} \\ T\Gamma_c^y C & T\Phi_c T^{-1} \end{pmatrix}$$

$$\Gamma_{cl}^{u_c} = \begin{pmatrix} \Gamma D_c^{u_c} \\ T\Gamma_c^{u_c} \end{pmatrix} \quad \Gamma_{cl}^{\eta_z} = \begin{pmatrix} 0 \\ I \end{pmatrix} \quad \Gamma_{cl}^{\eta_u} = \begin{pmatrix} \Gamma \\ 0 \end{pmatrix}$$

$$C_{cl}^{y} = \begin{pmatrix} C & 0 \end{pmatrix} \quad C_{cl}^{u} = \begin{pmatrix} D_c^y C & C_c T^{-1} \end{pmatrix}$$

The impulse-responses from $\eta_z(k)$ and $\eta_u(k)$ to $y(k)$ are $h_{\eta_z}(k) = C_{cl}^{y}\Phi_{cl}^{k-1}\Gamma_{cl}^{\eta_z}$, $k > 0$, and $h_{\eta_u}(k) = C_{cl}^{y}\Phi_{cl}^{k-1}\Gamma_{cl}^{\eta_u}$, $k > 0$. Inspection of Φ_{cl} reveals that the structure

$$\Phi_{cl}^{k-1} = \begin{pmatrix} * & *T^{-1} \\ T* & T*T^{-1} \end{pmatrix} \tag{7}$$

is preserved for all $k > 1$. Hence, the impulse-responses will have the structures $h_{\eta_z}(k) = C*T^{-1}$ and $h_{\eta_u}(k) = C*\Gamma$ for $k > 1$. Obviously $h_{\eta_u}(k)$ is invariant under the change of state variables, while $h_{\eta_z}(k)$ is not. With a large diagonal state scaling $T = \Lambda$, the output response to impulse errors on the states can be made arbitrary small. However, the state scaling is constrained by the available numeric precision.

The problem to find the controller realization, i.e. find the T, such that the output response $y(k)$ to the disturbance $\eta_z(k)$ is minimized with respect to some measure, under the constraint that the state scaling is kept appropriate for the available numeric representation is treated in e.g. [11, 13]. The results are optimal realizations in the context of round-off errors, which have the same problem structure, but where the disturbances are close to stationary white noise processes. With $y(k) = H_{\eta_z}(q)\eta_z(k)$, $\bar{z}(k) = H_z(q)u_c(k)$, solutions to the minimization of $\|H_{\eta_z}\|_p$ with respect to T, subject to the state scaling constraint $\|H_z\|_q \le \gamma$ are presented for $p, q \in \{2, \infty\}$ in [13]. For data errors it may be more natural to regard $p = q = 1$, i.e. peak-to-peak gain. Even more natural would it be to minimize $\|h_{\eta_z}\|_\infty$, since the disturbances are assumed to be impulses. However, in practice it is likely more feasible to optimize the realization with respect to the all-time present round-off noise than rare and sporadic data errors. In the following it will thus be assumed that the transformation T is given, and we concentrate on the analysis of the influence of η_z and η_u on the closed-loop performance. Hence, we can consider the representations (3) and (5) of the controller and the process.

4. Signal Bounds

If a bound on the command signal u_c is known, then l_∞ bounds on controller states and the control signal may be computed using l_1 norms on impulse responses [7]. The controller state and control signal are given by

$$z_i(k) = \sum_{j=0}^{k} h_i(k-j)u_c(j) \tag{8a}$$

$$u(k) = \sum_{j=0}^{k} h_u(k-j)u_c(j) \tag{8b}$$

where $\{h_i(k)\}$ and $\{h_u(k)\}$ are the impulse-response sequences of the closed-loop system from the command signal $u_c(k)$ to the state variable $z_i(k)$ and control signal $u(k)$. Bounds on z_i may be computed by applying the Hölder inequality on (8a):

$$\begin{aligned} |z_i(k)| &\le \sum_{j=0}^{k} |h_i(k-j)u_c(j)| \\ &\le \Big[\sum_{j=0}^{k} |h_i(j)|^p\Big]^{1/p} \Big[\sum_{j=0}^{k} |u_c(j)|^q\Big]^{1/q} \end{aligned} \tag{9}$$

where $\frac{1}{p} + \frac{1}{q} = 1$. If the maximum absolute value of the command signal is known $|u_c(k)| \le M$, then Equation (9), with $p = 1$ and $q = \infty$, gives

$$|z_i(k)| \le M \sum_{j=0}^{k} |h_i(j)| \tag{10a}$$

Correspondingly a bound on $u(k)$ may be expressed as

$$|u(k)| \le M \sum_{j=0}^{k} |h_u(j)| \tag{10b}$$

It is straightforward to include the effects of other bounded inputs to the closed-loop system, e.g. load disturbances, in the bounds of Equations (10). The bound (10a) may also be used for l_1 state scaling, as described in [7]. If the closed-loop system is well designed with proper damping, the bounds are not expected to be very conservative, which is also confirmed in [7]. If the signal bounds are exceeded it can be concluded that an error has occurred in the system. Hence, the signal bounds may be used on-line to actively detect deviations from normal operation, similarly to the approach in e.g. [16]. Note, however, that in the cited work the bounds appear to be computed on the controller in open-loop, which result in over-estimation of the bounds by factors of magnitudes in comparison with the closed-loop bounds. The approach taken in the present work is to introduce explicit bounds in the controller, that correspond to those of (10), and then use well-known anti-windup methods to handle these signal limitations in a graceful manner. This will give the system an inherent robustness to data errors that exceed the signal bounds.

5. Anti-windup

In the presence of control signal limitations, the control signal actually delivered to the controlled process will be $u(k) = \mathrm{sat}(v(k))$, where $v(k)$ is the linear control signal. When the output signal is saturated the feedback path is broken and the controller states are driven in open-loop, leading to deteriorated performance or even instability. If the controller has integral action this phenomenon is denoted integrator windup. To inhibit this behavior various anti-windup schemes may be applied [6]. Anti-windup should always be implemented in a controller with actuator saturation. In this work an explicit artificial limitation according to (10b) is introduced to make the system robust to data errors. In practice an actuator limitation will also be present. If the actuator saturation limit is smaller than the artificial limit (10b), then the smallest limit should be used. The observer-based anti-windup of [3] is a general method where the control signal error is fed back to the controller. With observer-based anti-windup the controller (3) is modified as

$$\begin{aligned} z(k+1) &= \Phi_c z(k) + \Gamma_c^y y(k) + \Gamma_c^{u_c} u_c(k) \\ &\quad + K\left(u(k) - v(k)\right) + \eta_z(k) \\ &= \hat{\Phi}_c z(k) + \hat{\Gamma}_c^y y(k) + \hat{\Gamma}_c^{u_c} u_c(k) \\ &\quad + Ku(k) + \eta_z(k) \\ v(k) &= C_c z(k) + D_c^y y(k) + D_c^{u_c} u_c(k) + \eta_u(k) \\ u(k) &= \mathrm{sat}(v(k)) \end{aligned} \tag{11}$$

with $\hat{\Phi}_c = \Phi_c - KC_c$, $\hat{\Gamma}_c^y = \Gamma_c^y - KD_c^y$, and $\hat{\Gamma}_c^{u_c} = \Gamma_c^{u_c} - KD_c^{u_c}$. The gain K is chosen as to obtain the desired observer dynamics given by $\hat{\Phi}_c$. The anti-windup scheme will now reduce the effect of data errors that are causing the controller output to exceed the estimated limits. Note that the observer based anti-windup operates on all controller states, while certain other schemes only operates on the integrator state. Hence, errors in any of the controller states are eliminated due to the anti-windup. Also note that the observer-based method does not require any additional states to be introduced. In the context of data errors this is important, since data errors affecting explicit anti-windup states would not be handled gracefully by the system. The closed-loop system resulting from combining (11) with (2) is shown in Figure 2.

6. Example

As an illustration of the inherent tolerance to transient faults that may be achieved with the combination of signal bounds and anti-windup, we study the control of the simple servo process

$$G(s) = \frac{100}{s(s+10)} \tag{12}$$

A discrete-time two-degrees-of-freedom tracking controller is synthesized using the polynomial pole-placement design method of [3]. The controller is designed for a closed-loop

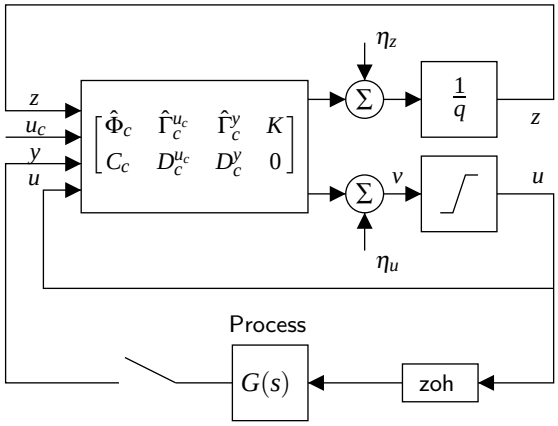

Figure 2 Blockdiagram of closed-loop system with artificial control-signal bound and anti-windup.

bandwidth of $\omega_{cl} = 15$ rad/s, observer dynamics of $2\omega_{cl}$, and integral action. The sampling time is set to 0.01 s. A minimal second-order modal-form state-space realization of the controller is used for implementation, with $z = (z_1, z_2)$. In this

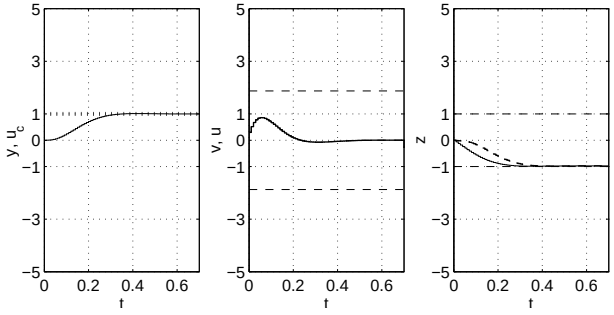

Figure 3 Closed-loop system response to a command signal step. Computed bounds on u and z are shown. (Left: u_c (dotted), y (solid); Middle: u (solid), bounds (dashed); Right: z_1 (solid), z_2 (dashed), bounds (dashed))

realization z_1 is an integrator state, and z_2 may be interpreted as an observer state. The command signal u_c is assumed to be bounded by $|u_c(k)| \leq 1$. The l_1 state-scaling of (10a) is applied such that $|x_i(k)| \leq 1$. The control signal bound according to (10b) is $|u(k)| \leq 1.88$. (The corresponding open-loop bound according to [16] is $|u(k)| \leq 341$.) The anti-windup scheme of (11) is implemented, with K chosen as to obtain dead-beat dynamics, to obtain the closed-loop system of Figure 2. In Figure 3 the closed-loop step-response to a command-signal unit step is shown. Note that the computed state-bounds are very tight, while the control signal bound seems to be a little more conservative. In Figure 4 the closed-loop response to a data error $\eta_{z_1}(k) = 5\delta(k)$, affecting the computation of the integrator state z_1, is shown in the case when the artificial signal-limit and anti-windup are not applied. It can be seen how the integrator state slowly recovers, while the control error grows large. In presence of actuator limitation large data-error amplitudes even result in instability. Figure 5 shows the corresponding response with applica-

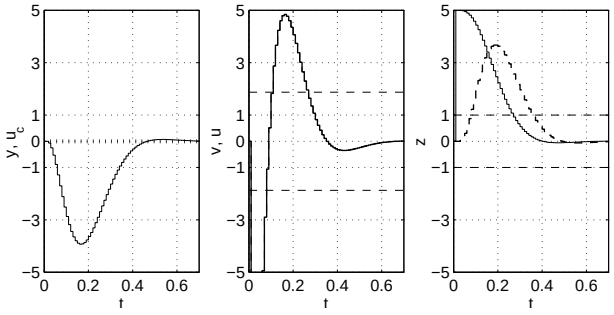

Figure 4 Closed-loop system response to a data error $\eta_{z_1}(k) = 5\delta(k)$, on controller state z_1, without signal limitation and anti-windup. Computed bounds on u and z are shown. (Left: u_c (dotted), y (solid); Middle: u (solid), bounds (dashed); Right: z_1 (solid), z_2 (dashed), bounds (dashed))

tion of the artificial signal limit $|u(k)| \leq 1.88$, in combination with dead-beat anti-windup. In this case the controller state recovers within a few samples, and the control error is moderate. Figure 6 shows the largest absolute control-error that

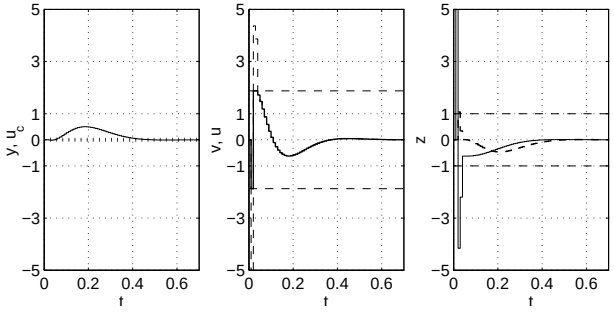

Figure 5 Closed-loop system response to a data error $\eta_{z_1}(k) = 5\delta(k)$, on controller state z_1, with signal limitation and dead-beat anti-windup. Computed bounds on u and z are shown. (Left: u_c (dotted), y (solid); Middle: u (solid), v (dashed), limits (dashed); Right: z_1 (solid), z_2 (dashed), bounds (dashed))

results from various magnitudes on the data errors η_u, η_{z_1}, and η_{z_2}. Four different system settings are shown for comparison: (i) No control-signal limits or anti-windup. Here the graph is a straight line with a slope equal to the l_1-norm of the impulse response. (ii) Dead-beat anti-windup. Note that the response to data errors affecting the control signal (η_u) leads to larger control errors in this case. This is because the data error is interpreted as a saturation by the anti-windup, which is fast enough to react immediately. Important to note is also that the control error magnitude does not increase with the data error magnitude for large data errors. (iii) Dead-beat anti-windup only on the integrator state (this is common in e.g. PI-controllers). This is normally sufficient for handling actuator limitations, but in the case of data-errors the performance is inferior compared with full-state anti-windup. (iv) Dynamic anti-windup with a bandwidth of $2\omega_{cl}$. Here the response to data errors entering the control signal is better, since the anti-windup is too slow to react immediately on the error impulse. The response to data errors on the state is, however, worse than for the dead-beat design, as is clear

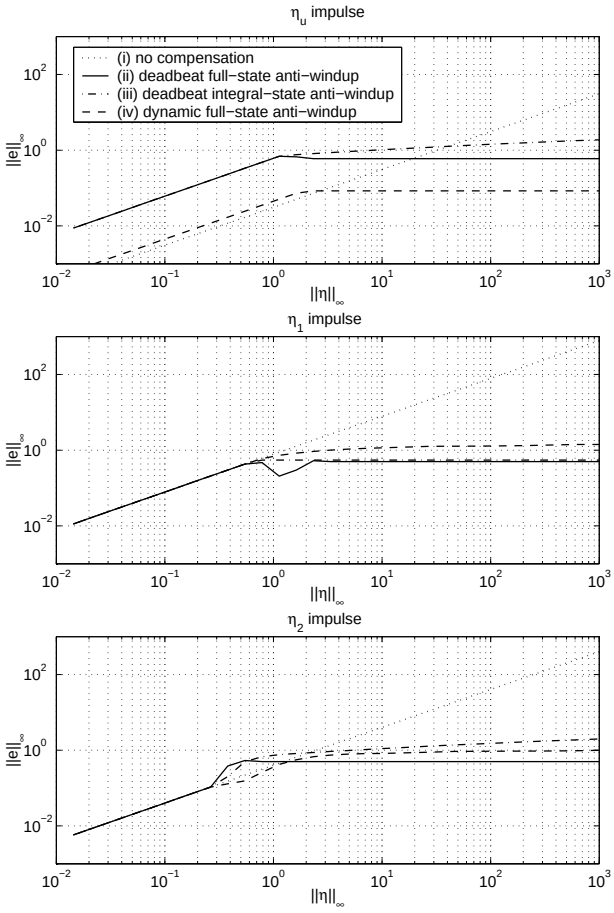

Figure 6 Closed-loop system response to impulse data errors. Control-error magnitude versus data-error magnitude. (Upper: η_u; Middle: η_{z_1}; Lower: η_{z_2})

from the figure. The recovery time is also significantly slower compared to the dead-beat design, which may be seen from time-domain response plots.

7. Discussion

In general it can be noted that the integrator state z_1 is most sensitive to data errors, which seems very intuitive since it depends on feedback of the control error to decay, while the controller state z_2 is stable with a short time constant, and decays by itself when the loop is broken by saturation. Another general observation is that the control-signal bound is computed from bounds on the command signal, and in consequence is related to the expected magnitude of control errors during normal operation. Hence, the artificial limits and the anti-windup scheme will capture data errors resulting in control errors larger than those expected during normal operation. The proposed method may also be combined with the dynamic bounds of [17], to increase the coverage for data errors.

Since the controller state z is bounded by (10a), it may seem natural to introduce explicit limits also for this variable.

However, simulations with saturations on the controller state indicate that the performance improvement is minor. The increased complexity resulting from additional saturations in the loop also makes the system difficult to analyze, even if it seems to perform well in presence of data errors.

In presence of stochastic signals such as measurement noise there will be a probability of false error detections. This may be handled by computing the resulting variance of the control signal. The deterministic control-signal bound (10b) is then adjusted with some measure depending on the variance. The size of the adjustment will determine the probability of false detections. Sporadic false detections will affect control performance as the anti-windup intervenes. By using slower anti-windup dynamics the noise sensitivity is decreased.

If a rate bound on the command signal is known $|\Delta u_c(k)| = |u_c(k) - u_c(k-1)| \leq M_\Delta$, then state and control-signal rate bounds may be computed in analogy with (10). An artificial rate bound on the control signal may then also be introduced in the controller, and used together with the anti-windup scheme. Note that the noise sensitivity will be worse than for the case with magnitude bounds, since the rate of the control signal will depend on the noise variance.

8. Summary

Data errors resulting from transient faults in the computer hardware may be modeled as impulse errors entering the control algorithm internally. The effect on closed-loop performance therefore depends on the controller realization. Previous methods to optimize controller realizations with respect to round-off errors in finite-precision numerics are applicable also in the context of transient data errors. Given a controller realization, robustness to data errors may be achieved by introducing artificial signal limitations based on l_∞-bounds, in combination with an anti-windup scheme. As one may expect, the integrator state of the controller is most sensitive to data errors.

9. References

[1] Ö. Askerdal. *Design and Evalutation Techniques for Detection and Coverage Estimation of Low-Level Errors*. Lic. thesis, Chalmers University of Technology, Sweden, 2000.

[2] Ö. Askerdal, M. Gäfvert, M. Hiller, and N. Suri. A control theory approach for analyzing the effects of data errors in safety-critical control systems. In *Proceedings of the Pacific Rim International Symposium on Dependable Computing*, 2002.

[3] K. J. Åström and B. Wittenmark. *Computer-Controlled Systems*. Prentice Hall, 3rd edition, 1997.

[4] C. Constantinescu. Impact of deep submicron technology on dependability of VLSI circuits. In *Proceedings of the IEEE International Conference on Dependable Systems end Networks*, 2002.

[5] J. C. Cunha, R. Maia, M. Z. Rela, and J. G. Silva. A study of failure models in feedback control systems. In *Proceedings of the IEEE International Conference on Dependable Systems and Networks*, pages 314–323, 2001.

[6] C. Edwards and I. Postlethwaite. Anti-windup and bumpless-transfer schemes. *Automatica*, 34(2):199–210, 1998.

[7] H. Hanselmann. Implementation of digital controllers — A survey. *Automatica*, 23(1):7–32, 1987.

[8] P. Hazucha. *Background Radiation and Soft Errors in CMOS Circuits*. PhD thesis, Linköping University, Sweden, 2000.

[9] H. Kim and K. G. Shin. On the maximum feedback delay in a linear/nonlinear control system with input disturbances caused by controller-computer failures. *IEEE Transactions on Control Systems Technology*, 2(2):110–122, 1994.

[10] H. Kim, A. L. White, and K. G. Shin. Effects of electromagnetical interference on controller-computer upsets and system stability. *IEEE Transactions on Control Systems Technology*, 8(2):351–357, 2000.

[11] P. Moroney, A. S. Willsky, and P. Houpt. Roundoff noise and scaling in the digital implementaion of control compensators. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31(6):1464–1477, 1983.

[12] D. Pradhan. *Fault-Tolerant Computer Systems Design*. Prentice Hall, 1996.

[13] M. A. Rotea and D. Williamson. Optimal realizations of finite wordlength digital filters and controllers. *IEEE Transactions on Circuits and Systems — I*, 42(2):61–72, 1995.

[14] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi. Modeling the effect of technology trends on the soft error rate of combinatorial logic. In *Proceedings of the IEEE International Conference on Dependable Systems end Networks*, 2002.

[15] T. J. Slegel, R. M. Averill III, M. A. Check, B. C. Giamei, B. W. Krumm, C. A. Krygowski, W. H. Li, J. S. Liptay, J. D. MacDougall, T. J. McPherson, J. A. Navarro, E. M. Schwarz, K. Shum, and C. F. Webb. IBM's S/390 G5 microprocessor design. *IEEE Micro*, 19(2):12–23, 1999.

[16] R. Stroph and T. Clarke. Static acceptance test for complex controllers. In *Proceedings of the UKACC International Conference on CONTROL*, pages 392–397. IEE, 1998.

[17] R. Stroph and T. Clarke. Dynamic fault detection approaches. In *Proceedings of the American Control Conference*, pages 627–631, 1999.

[18] J. Vinter. Reducing critical failures for control algorithms using executable assertions and best effort recovery. In *Proceedings of the IEEE International Conference on Dependable Systems and Networks*, pages 347–356, 2001.

[19] B. Wittenmark, J. Nilsson, and M. Törngren. Timing problems in real-time control systems: Problem formulation. In *Proceedings of the American Control Conference*, pages 2000–2004, 1995.