

Rate Derivative Acceptance Test for Digital Controllers

R. Stroph and T. Clarke

Department of Electronics, The University of York
YO1 5DD York, UK, email: rs15@ohm.york.ac.uk

1 Introduction

The success of dynamic recovery from fault conditions rests to a great extent on the fault detection mechanism. This paper develops a simple but highly effective, novel approach to designing an acceptance test, as used in the recovery block schemes, for control applications. It is based on a reasonableness check of controller output signals, comparing it with a rate derivative threshold, as it operates, which is being developed here. This scheme, which has minimal overheads, was applied to a helicopter control system and results are given.

2 Controller Output Bounds

The following notation is used. A single-input, single-output control system, which is represented by the pair (S, \mathbb{F}) , is considered with input function $u : \mathbb{R} \rightarrow \mathbb{R}$ and output function $y(u) : \mathbb{R} \rightarrow \mathbb{R}$. \mathbb{F} is the set of all input functions and S is the input-output relationship, $S : u \mapsto y(u)$, which maps the input space, \mathbb{F} , into an output space. The input function at time t is denoted $u(t)$, so that $u : t \mapsto u(t)$, and the output at time t is denoted $y(t, u)$, so that $y(u) : t \mapsto y(t, u)$.

3 S-Domain Bounds

Assuming that the control system is linear, causal and time invariant, it may be described in the s -domain by its input $U(s)$, and output (for a given input u) $Y(s, u)$. The control system transfer function is thus written $Y(s, \delta)$, where δ indicates an impulse. The system is therefore characterised in the s -domain and the time domain, using the inverse Laplace transform, respectively as follows:

$$Y(s, u) = Y(s, \delta)U(s) \quad (1)$$

$$y(t, u) = \int_0^t y(t - \tau, \delta)u(\tau) d\tau \quad (2)$$

Typically, the control laws under consideration consist of parts or are linear, causal and time-invariant and therefore satisfy those assumptions.

3.1 Rate Derivative Bounds: $\mathbb{F}(A)$

The input space \mathbb{F} indicates the set of all functions $u : \mathbb{R} \rightarrow \mathbb{R}$ such that $u(t) = 0$, $t \leq 0$, the second derivative $\ddot{u}(t)$ is piecewise-continuous and there is a finite positive number A , such that

$$A = \sup\{|\ddot{u}(t)| : t \in \mathbb{R}^+, u \in \mathbb{F}(A)\} \quad (3)$$

The least upper bound performance indicator, ϕ on the rate derivative of a controller output, which forms the basis for the acceptance test, is defined as

$$\phi_{\dot{y}} = \sup\{|\dot{y}(t, u)| : t \in \mathbb{R}^+, u \in \mathbb{F}(A)\} \quad (4)$$

Theorem 1 Given the system described, which may be represented by Equation 2 for all $u \in \mathbb{F}(A)$, then

$$\phi_{\dot{y}} = \int_0^\infty |y(\delta)| dt A \quad (5)$$

where $y(\delta)$ is the impulse response, which is piecewise-continuous.

Proof The proof is straightforward, when applying the derivative property to Zakian's critical control framework [1].

Therefore, the control law outputs are constrained by the integral over the absolute impulse response multiplied by a rate derivative bound on the input signals. In the case of a violation of the least upper bound, $\phi_{\dot{y}}$, by the controller output signal, the controller is deemed to have failed.

4 The Experimental Implementation

Evaluating the acceptance tests requires a realistic control system encompassing a plant, and a digitally-implemented control law. A simulated plant was used, based upon a non-linear model of the Lynx helicopter [2] of which a 10th order linearisation was employed. The controller selected was an \mathcal{H}_∞ design and led to a fourteen state control law.

Testing the acceptance test relies on a comparison between two identical, synchronised control laws. During

fault detection tests, one control law is subject to fault injections and any resultant failures are manifest as differences between the two controller output signals detected by a comparator. A widely favoured fault injection approach uses the bit-flip, or bit-inversion, hardware error model, which has been shown to provide an accurate representation of micro-processor errors [3]. In addition, it is possible to argue that, theoretically, bit-flip errors correspond to random software faults.

Separate tests were run for data faults, injected into the workspace area, and instruction faults, initiated by corrupting the next instruction in sequence. A further fault type, called internal faults, caused errors in the internal registers of the processor. Full details of the fault injection scheme and the fault types are given in [4].

The maximum input rate derivatives, A , are derived from experience with helicopters [5]. Results were collected from 2500 experimental flight simulations for either injected data, instruction or internal faults. A very demanding controller frame rate of $400\mu\text{s}$ was used.

The compiled size of the source code for the control law and the absolute and relative size of the rate derivative acceptance test, plus its time overhead is shown in Table 1.

Table 1: Comparison of the overheads for the rate derivative acceptance test to the control law.

Code Description	Code Size	Data Size	Exec. Time
\mathcal{H}_∞ Control Law	2548bytes	1608bytes	6090 μs
Rate derivative acceptance test	216bytes 8.5%	72bytes 4.5%	36.94 μs 0.61%

5 Results

The testing described above, lead to either detected, undetected or falsely identified errors. Additionally, a watchdog timer identified processor stops. The following results in Table 2 are split into two parts, the first, considering both the watchdog timer, detecting the “processor stops”, and the acceptance test, combined in the number of “total detected” errors. In the second part the focus was on the acceptance test and not the watchdog timer and shows the results of all non processor stops, and so that only the rate derivative acceptance test performance was measured.

Table 2: Results of the novel rate derivative bounded acceptance tests.

Rate derivative	Data Faults	Instruction	Internal
Total detected	92.4%	99.5%	97.9%
Processor stops	64.1%	59.9%	74.9%
Detected errors	78.7%	98.8%	91.8%
Undetected errors	17.5%	1.1%	7.8%
False alarms	3.8%	0.1%	0.3%

The results show, that this simple and easily implemented acceptance test is highly effective for this complex multi-input multi-output \mathcal{H}_∞ control law implementation.

Another promising result obtained during these experiments concerns the speed with which errors resulting from injected faults are detected, which is of critical importance to real-time control systems. In the case of this simple, generic, rate derivative bounded acceptance test, 90.6% of detected data faults were detected immediately and 94.2% with a delay of one control iteration. For instruction faults, 99.0% and 99.1%, and for injected internal faults 96.2% and 99.9% were detected within the same and after one control iteration, respectively.

6 Conclusion

The theoretical foundations of an acceptance test using on-line output bounds has been presented. The appeal of this approach is due to a combination of the simplicity of the test implementation and the rigorous proof which supports it. Existing theory has been re-interpreted to provide bounds on the rate derivative of the outputs of MIMO control systems.

Including this test in, for example a distributed recovery block in which a watchdog timer is also used, the fault detection coverage would be the given by the average of the total detected errors, being 96.6%.

This test may be applied to any linear, causal, time-invariant part of or whole control system. To the best of the authors’ knowledge, this represents the first rigorous, completely implementation-independent, linear system rate derivative acceptance test to be developed.

References

- [1] V Zakian. Input spaces and output performance. *Int. Journal of Control*, 47(1):185–191, 1987.
- [2] T Clarke and X D Sun. Minimal state-space model realisation of a non-linear helicopter. In *Proc. Control Theory Applications*, volume 145. IEE, July 1998.
- [3] M Rimén, J Ohlsson, and J Torin. On microprocessor error behaviour modelling. In *Proc. 24th Int. Symp. on Fault Tolerant Computing*, pages 76–85. IEEE, 1994.
- [4] R Stroph and T Clarke. Hardware and software fault simulation. In *Proceedings on the International Conference on Simulation '98*, pages 413 – 419, York, United Kingdom, October 1998. IEE.
- [5] P Taylor. Personal Communications, 1998. GKN-Westland Helicopters Ltd., Yeovil, Somerset, UK.