PROCEEDINGS OF THE

27th IEEE CONFERENCE ON DECISION AND CONTROL

DECEMBER 7-9, 1988
HYATT REGENCY AUSTIN ON TOWN LAKE
AUSTIN, TEXAS

IEEE
Control
Systems
Society

VOLUME 1 OF 3

88CH2531-2

THE 27th IEEE CONFERENCE ON DECISION AND CONTROL

DECEMBER 7-9, 1988

SPONSORING ORGANIZATIONS
IEEE Control Systems Society

COOPERATING ORGANIZATIONS
Society for Industrial and Applied Mathematics (SIAM)
Operations Research Society of America (ORSA)

OPERATING COMMITTEE

GENERAL CHAIRMAN	Michael P Polis	Dept of Elec & Comp Eng Wayne State University Detroit MI 48202
PROGRAM CHAIRMAN	William E Schmitendorf	University of California Irvine Irvine CA 92717
EXHIBITS	Ching-Fang Lin	American GNP Corp
FINANCE	Frank L Lewis	Georgia Institute of Technology
LOCAL ARRANGEMENTS	Klaus Dannenberg	Merit Technology Inc
PUBLICATIONS	Marija Ilic	Massachusetts Institute of Technology
PUBLICITY	Robert Lobbia	Orincon Corp
REGISTRATION	Kenneth M Sobel	CCNY
PROGRAM COMMITTEE	W E Schmitendorf	University of California Irvine
	J Baillieul	Boston University
	S Banda	Wright Patterson AFB
	B R Barmish	University of Wisconsin
	T Basar	University of Illinois
	D L Boley	University of Minnesota
	P Dorato	University of New Mexico
	J E Falk (ORSA)	George Washington University
	J S Freudenberg	University of Michigan
	J S Gibson	University of California LA
	C V Hollot	University of Massachusetts Amherst
	F Jabbari	University of California Irvine
	S M Joshi	NASA Langley Research Center
	E W Kamen	University of Pittsburgh
	P R Kumar	University of Illinois
	D P Looze	University of Massachusetts Amherst
	C Martin (SIAM)	
	K Poolla	University of Illinois
	W J Rugh	Johns Hopkins University
	M K Sain	University of Notre Dame
	H Stalford	Virginia Polytechnic and State Univ

Library of Congress Catalog Card Number: 79-640961
IEEE Catalog Number: 88CH2531-2

Additional copies of this volume may be ordered from the IEEE Service Center 445 Hoes Lane Piscataway NJ 08854 referring to IEEE Catalog Number: 88CH2531-2

Proceedings of the 27th Conference
on Decision and Control
Austin, Texas • December 1988

RULE-BASED REAL-TIME IMPLEMENTATION OF NONLINEAR SELF-SYNTHESIZING CONTROL SYSTEMS

James H. Taylor
Control Systems Lab
GE Corporate R & D
PO Box 8
Bldg. KW Room D209A
Schenectady, NY 12301

ABSTRACT

Implementation of **advanced nonlinear control systems** is studied by combining recent advances in nonlinear control system synthesis with a rule-based system approach to real-time control. The basic idea is that a nonlinear plant to be controlled may be quite sensitive to both operating point and input amplitude, so that the "best" control system performance is obtained with a nonlinear controller that is "retuned" or "re-synthesized" whenever the operating point changes significantly in the sense that the nonlinear control system input/output behavior changes substantially in an undesirable way.

The control system thus obtained is hierarchically organized, with a standard reprogrammable controller under the direction of a rule-based system that:

- monitors the behavior of the nonlinear control system to determine when retuning or re-synthesis is required; if the behavior is satisfactory then continue passive monitoring; else
- when retuning or re-synthesis is required, sets up and executes experiments to derive the information required to tune or synthesize a new nonlinear controller in terms of a given structure and parameterization; executes the retuning / re-synthesis procedure; re-programs the controller (down-loads the parameterization); and recommissions the updated nonlinear control system and returns to normal operation.

In essence, the rule-based system provides supervisory control ("meta-control") for the conventional reprogrammable nonlinear controller. This concept represents one way to combine artificial intelligence with control; it will be discussed and illustrated by example below.

1. INTRODUCTION

This presentation focusses on implementing **advanced control systems** via the use of **rule-based systems for real-time control**. The question is: how can one apply expert systems technology to control, and what are the advantages? We have explored this via several examples and describe the results below. First, some general comments:

Rule-based systems are software environments that can be used for real-time control system implementation, relieving the system engineer of much of the burden of taking a "textbook design" (e.g., nonlinear control algorithm) and making it work in a real-world application. The specific problem addressed is the well-known fact that obtaining a control algorithm is often a small percentage of the control engineer's overall effort; implementing it (making it work in terms of system interfaces, initialization procedures, logic, exception-handling, operator interface, etc.) is usually the more challenging part of the engineering task. This is especially true with advanced systems designs where complicated features such as adaptive algorithms, failure detection and isolation schemes, parameter identification, etc. must be embedded as an integral part of the control software.

There is often no standard or even systematic approach to implementing a controller; rather, the engineer adds logic and other code "patchwork" to the controller software until it works in the real system. If the system is complicated, the resulting logic and algorithmic modifications may become a large and unwieldy mass of "spaghetti code" that is difficult to implement, document, and maintain.

The rule-based systems approach can be used to provide a great deal of support for the above task, resulting in a more flexible system implementation with less iteration and greater likelihood of success. In essence, the rule-based system provides the environment in which to develop and test all of the required heuristic logic and control, using a programming paradigm (rule bases) and language (rule writing) that is ideally suited for the task. An appropriate expert system shell then provides the software framework in which the rule-based system operates.

The resulting rule-based real-time control system may actually be implemented as a rule-based system, or it may be "compiled" into a lower-level language if the flexibility of the rule-based environment is no longer needed in the target application. Compiling the meta-controller results in a system that is typically faster in execution and that can be installed on a less expensive host processor. Either way, the design and implementation effort has been greatly simplified, and the full control implementation can be maintained and documented in the rule-based form.

2. A BASIC ARCHITECTURE FOR REAL-TIME RULE-BASED CONTROL

There are various models for the combination of expert systems and control technology. Our concept [1] is a **real-time rule-based control system** having:

- **sensors** (monitors) to determine the state of the process in terms of "signals", e.g., sensor outputs,
- **pattern recognizers** or feature extractors to process signals and create "symbols" or linguistic representations of the information (e.g., TRANSIENT %OVERSHOOT . EXCESSIVE),
- **conventional controllers,** with interfaces permitting the introduction of commands for gain-setting, reconfiguration, etc.,
- **rule bases** that contain the knowledge of the control system designer and of the overall control strategy for all regimes, and
- **an expert system shell or "inference engine"** that exercises the higher-level control ("meta-control") of the system.

A real-time rule-based control system configuration that incorporates these elements was described in detail in [1] and is portrayed in Fig. 1. The end result is a hierarchical control scheme that, at the higher level, embeds the expertise of the experienced system designer (or, in some contexts, a capable human operator) while making the best possible use of conventional control technology in the lower level. This general concept is quite similar to other approaches; cf. [2, 3].

The architecture outlined above was developed first as a vehicle for studying a failure detection and isolation methodology [4]. In that study [1] we developed a software framework for studying rule-based control; however, the problem provided little real substance for the rule-based system. The framework (a real-time control simulation environment with an embedded rule-based system) is depicted in Fig. 2. It incorporates a standard simulation environment for continuous- and discrete-time systems (in which one can model the plant and conventional reconfigurable controller) coupled to an expert system shell (inference engine) where the rule-based system is implemented in terms of generic real-time control logic and application-specific meta-control rules. The coupling of these software packages is managed via a simple coordination protocol so that the simulator stops at each meta-control sample time, passes control to the rule-based system which performs meta-control tasks, and then continues the simulation when the expert system has completed its part of the cycle. From the controls simulation point of view, the expert system is just another discrete-time module. We emphasize that this is not a real-time implementation, just a simulation.

3. APPLICATIONS

More serious applications of this approach began by developing a rule-based system implementation of a recently-developed nonlinear PID autotuning algorithm [5] based on a linear autotuning scheme [6, 7] combined with a nonlinear controller synthesis method founded on sinusoidal-input describing functions (SIDFs) [8]. An overview of this approach in terms of the rule-based system implementation is shown in Fig. 3; it features a rule-based real-time control (RBRTC) supervisor partitioned as follows:

- a **performance monitoring rule base (unbuilt),** which samples the transient response of the control system and decides if retuning is needed;
- a **retuning rule base,** which implements the algorithm in [5] as follows:
  > *Experiment setup* - replace the PID controller with a hysteretic relay via the retune/control switch to produce relay-induced oscillations (RIOs) for system identification in the frequency domain (different relay output levels are used to obtain the plant frequency response at different amplitudes);

  > *first experiment execution* - carry out one RIO experiment for a "nominal" amplitude to obtain the corresponding magnitude and phase of the nonlinear plant response for several values of hysteresis (this determines several points on the nominal plant SIDF "Nyquist" locus);

  > *linear controller synthesis* - synthesize the corresponding controller using the above SIDF data (e.g., design a PID controller via Ziegler-Nichols tuning for the nominal amplitude);

  > *second experiment execution* - carry out a *set* of RIO experiments to obtain the corresponding magnitude and phase of the linear controller in series with the nonlinear plant as the open-loop system responds to different amplitude periodic inputs (different relay levels);

  - *nonlinearity synthesis* - determine the compensating piece-wise-linear nonlinearity that reduces the open-loop amplitude sensitivity determined above by determining the desired gain(amplitude) relation and using SIDF inversion; and

- a **recommissioning rule base,** which puts the new nonlinear PID controller into service once retuning is complete via the retune/control switch.

Thus, when required by a determination that the control system is not performing satisfactorily, the control system goes through a period of retuning under the supervision of the rule-based system; the new controller definition (parameters for PID gains and the parameterization of a piece-wise-linear desensitizing nonlinearity) is downloaded to the reconfigurable controller; and it is placed back in service.

After the above system was realized in simulation form (Fig. 2), we invented a new, more general nonlinear control self-synthesis approach that achieves the generality of the synthesis method described in [9]. The main difference between the synthesis in [8] and [9] is that the method in [8] synthesizes a *single* nonlinearity to precede an *arbitrary* linear controller to attempt to reduce the sensitivity of the open-loop system to input amplitude; in [9] a nonlinear PID controller is synthesized that has *three independent nonlinearities* in the controller paths (proportional, integral, derivative), thus allowing more degrees of freedom in desensitizing the open-loop system and thus less sensitive closed-loop system behavior as demonstrated in the example in [9]. A schematic of the nonlinear PID control system is shown in Fig. 4. The benefit of added degrees of freedom can be appreciated by noting that this configuration allows the synthesis of desensitizing nonlinearities for low frequency (the 'I' term), mid-frequency (the 'P' term), and high ('D').

The details of our latest synthesis approach differ from those in [9]; they are reported in [10]. The most noteworthy aspect of this approach is that the derivative feedback path need not be parallel to the proportional and integral paths as shown in Fig. 4; instead, the rate term may be in the feedback path, which is highly preferable as this avoids over-driving the plant when the reference input changes abruptly.

An outline of the corresponding rule-based system differs from that mentioned above and defined in [5] primarily in replacing the retuning rules with **self-synthesizing rules:**

- **the monitor** (still unimplemented) determines when re-synthesis is required; if the behavior is satisfactory then continue passive monitoring; else
- the **re-synthesis rule base** is invoked to:
  - set up and execute a series of experiments to drive a *variable-parameter linear PID* control system with sinusoids of various amplitudes $\{ a_i \}$ (these range from "small" to "large" according to the designer's knowledge of the process) and frequencies $\{ \omega_k \}$ near the desired crossover frequency ω_{co} to determine (tune) linear controller gains $\{ K_{P,i} \}$, $\{ K_{I,i} \}$, $\{ K_{D,i} \}$ for each amplitude that desensitize the open-loop system input/output response $C_i\, G(j\omega_k; a_i)$ to varying amplitude → gain(amplitude) data $\{ K_{P,i}(a_i) \}$ etc.
  - execute a numerical procedure that uses the gain(amplitude) information as the basis for synthesizing desensitizing controller nonlinearities by SIDF inversion [9], and
  - re-program the controller (down-load the parameters that define the re-synthesized controller nonlinearities); and then
- the **recommissioning rule base** places the re-synthesized nonlinear control system into service and returns to normal operation.

Primary emphasis was placed on the re-synthesis rule base in the above studies; in fact, the monitoring and recommissioning rule bases have not been built. Re-synthesis was fully implemented, however, and the nonlinear plant from [5, 9] was modeled in the simulation environment shown in Fig. 2 to study the effectiveness of the approach and rule-based system. In brief summary:

- The plant was a simple but notoriously difficult model of a position control system with torque motor saturation and stiction (Fig. 5).
- The behavior of the plant in combination with a *linear* controller is depicted in Fig. 6, where it can be noted that the resulting feedback system exhibits "sticking" for low-amplitude inputs and excessive overshoot for large reference input steps due to integral windup.
- The behavior of the plant in combination with a **nonlinear** controller synthesized by the methodology outlined above is depicted in Fig. 7, where it can be noted that the resulting feedback system is much less sensitive to the amplitude of the reference step input.

4. CONCLUSION

A full implementation of a nonlinear self-synthesizing control system would clearly be a difficult task without the powerful environment provided by the rule-based systems approach described in Section 2. That such a system can be effective in controlling nonlinear plants (assuming that the control objective is minimizing the sensitivity of the feedback system to input-amplitude dependence without unnecessarily sacrificing performance) has also been shown. This study thus serves to demonstrate both the efficacy of the nonlinear synthesis approach and the rule-based meta-control concept.

REFERENCES

[1] J. H. Taylor, R. W. Gerhardt, and E. C. Luce, "Rule-Based Real-Time Control Systems", *Proc. Conf. on Decision and Control,* Los Angeles, CA December 1987.

[2] K. J. Åström and J. J. Anton, "Expert Control", *Proc. IFAC World Congress,* Budapest, 1984.

[3] K. J. Åström, "Adaptation, Auto-Tuning, and Smart Controls", *Proc. Chemical Process Control III,* Asilomar, CA, 1986.

[4] N. Viswanadham, J. H. Taylor, and E. C. Luce, "A Frequency-Domain Approach to Failure Detection and Isolation with Application to the GE-21 Turbine Engine Control System", *Control - Theory and Advanced Technology (C-TAT), Vol. 3,* March 1987.

[5] J. H. Taylor and K. J. Åström, "A Nonlinear PID Autotuning Algorithm", *Proc. American Control Conf.,* Seattle, WA June 1986.

[6] K. J. Åström and T. Hägglund, "Automatic Tuning of Simple Regulators", *Proc. IFAC 9th World Congress,* Budapest, 1984.

[7] T. Hägglund and K. J. Åström, "A New Method for Design of PID Regulators", Report TFRT-7273, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1984.

[8] J. H. Taylor and K. L. Strobel, "Applications of a Nonlinear Controller Design Approach Based on Quasilinear System Models", *Proc. American Control Conf.,* San Diego, 1984.

[9] Taylor, J. H. and Strobel, K. L., "Nonlinear Compensator Synthesis Via Sinusoidal-Input Describing Functions", *Proc. American Control Conf.,* Boston, MA, June 1985

[10] J. H. Taylor and P. Stringer, "A New Rule-Based Nonlinear PID Autotuning Algorithm", GE Corp. R & D Report 88CRDxxx, October 1988. 1986.

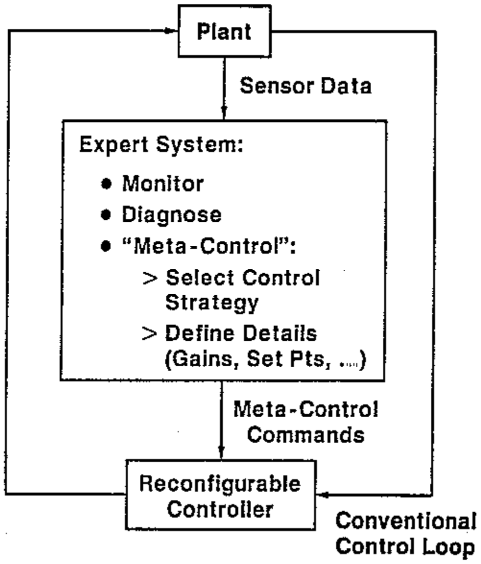

Fig. 1 Architecture for Real-time Control

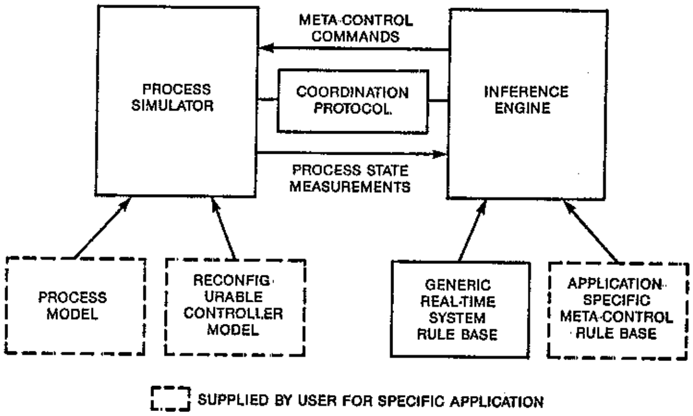

Fig. 2 Simulation Environment for Rule-based Real-time Control

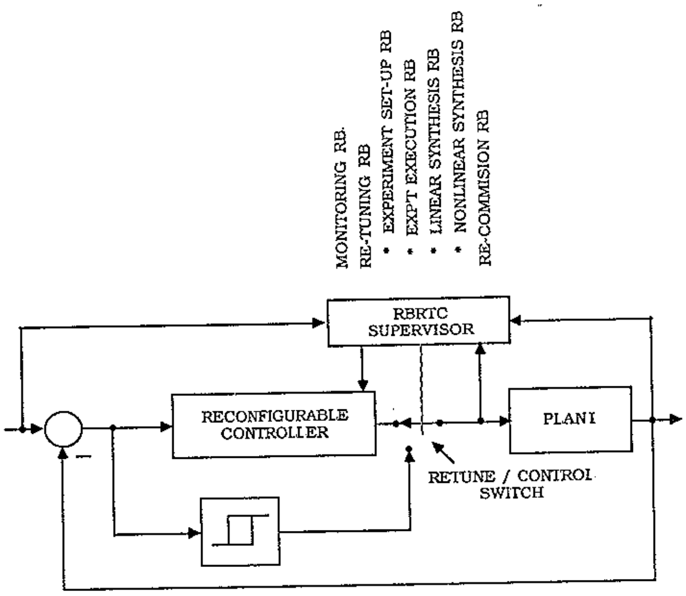

Fig. 3 RBRTC for Nonlinear Autotuning

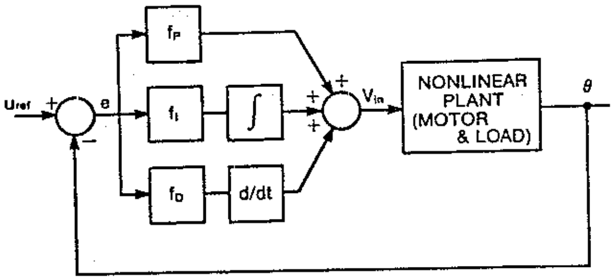

Fig. 4 Nonlinear PID Control System

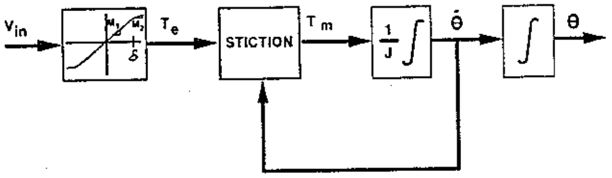

Fig. 5 Plant for Positioning Servo

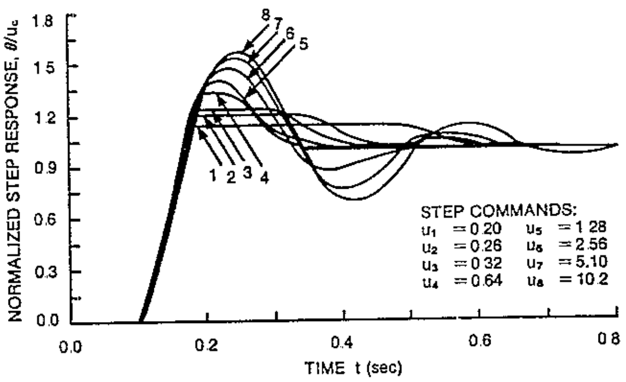

Fig. 6 Linear Position Control Performance

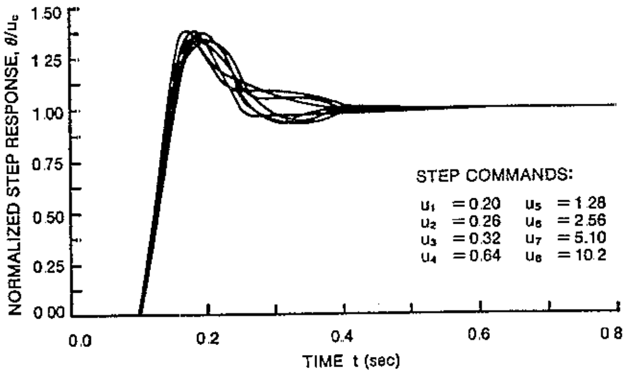

Fig. 7 Nonlinear Position Control Performance