Robert L. Grossman Anil Nerode
Anders P. Ravn Hans Rischel (Eds.)

Hybrid Systems

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editors

Gerhard Goos
Universität Karlsruhe
Postfach 69 80
Vincenz-Priessnitz-Straße 1
D-76131 Karlsruhe, Germany

Juris Hartmanis
Cornell University
Department of Computer Science
4130 Upson Hall
Ithaca, NY 14853, USA

Volume Editors

Robert Lee Grossman
Department of Mathematics, Statistics & Computer Science
University of Illinois at Chicago, Chicago, IL 60680, USA

Anil Nerode
Mathematical Science Institute, Cornell University
Ithaca, NY 14853, USA

Anders Peter Ravn
Hans Rischel
Department of Computer Science,Technical University of Denmark
DK-2800 Lyngby, Denmark

CR Subject Classification (1991): C.1.m, C.3, D.2.1, F.3.1, F.1-2

ISBN 3-540-57318-6 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-57318-6 Springer-Verlag New York Berlin Heidelberg

Typesetting: Camera-ready by author
Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
45/3140-543210 - Printed on acid-free paper

PREFACE

This volume of invited refereed papers is inspired by a workshop on the Theory of Hybrid Systems, held October 19-21, 1992 at the Technical University, Lyngby, Denmark, and by a prior Hybrid Systems Workshop, June 10-12, 1991 at the Mathematical Sciences Institute, Cornell University, USA, organized by R.L. Grossman and A. Nerode. Some papers are the final versions of papers presented at these workshops. Some are invited papers from other researchers who were not able to attend the workshops.

We are very grateful to Albert Benveniste, Anil Nerode, Amir Pnueli and Willem-Paul de Roever for their help in organizing this volume. We also wish to thank the following referees: H. R. Andersen, M. Basseville, T. Gautier, P. Le Guernic, C. Jard, Y. Lakhnech, H. H. Løvengreen, E. Rutten, P. Sestoft, J. Sifakis, J. U. Skakkebæk, and A. Yakhnis.

We gratefully acknowledge the financial support of the workshop in Lyngby granted by the Technical University of Denmark under the research programme "Mathematical Modelling of Computer Based Systems".

Department of Computer Science
Technical University of Denmark
Lyngby, June 1993

Anders P. Ravn
Hans Rischel

CONTENTS

Introduction

R.L. Grossman[1], A. Nerode[2], A. Ravn[3] and H. Rischel[3]

[1] Department of Mathematics, Statistics,
& Computer Science (M/C 249)
University of Illinois at Chicago
Chicago, IL 60680, USA
[2] Mathematical Sciences Institute
Cornell University, Ithaca, New York 14850
[3] Department of Computer Science
Technical University of Denmark
DK 2800 Lyngby, Denmark

Hybrid Systems are networks of interacting digital and analog devices. Inherently unstable aircraft and computer aided manufacturing are typical control theory areas of application for hybrid systems, but due to the rapid development of processor and circuit technology modern cars, for instance, and even consumer electronics use software to control physical processes. The identifying characteristic of hybrid systems is that they incorporate both continuous components, usually called plants, which are governed by differential equations, and also digital components, i.e. digital computers, sensors and actuators controlled by programs. These programs are designed to select, control, and supervise the behaviour of the continuous components. Modelling, design, and implementation of hybrid systems has recently become an active area of research in both computer science and control engineering. Hybrid systems are in almost all cases modelled as interacting networks of automata, possibly with an infinite number of states, and input and output letters.

How are hybrid systems to be analysed? How are they to be synthesized? How can we verify that they meet performance requirements? There are many possible approaches, and more questions than answers. Issues that have been addressed for many years in computer science in concurrency, distributed computing and program development and verification have to be rethought to be extended to hybrid systems. Likewise, issues that are classical in control engineering, such as observability, controllability, and stability, have to be rethought to be useful in hybrid systems.

For sequential programs, verifying that programs satisfy specifications comprises proving by induction that (some, all, no) execution sequences satisfy conditions. For concurrent and reactive systems the notion of an execution sequence has to be generalized, but proofs go along the same inductive lines. Hybrid systems are inherently concurrent and reactive. Furthermore, a suitable formalism has to incorporate techniques for specifying real-time constraints, and relating the model of the reactive system to the differential equations which describe the plants.

A suggestion is temporal logic based systems, as in the phase transition systems by *Manna* and *Pnueli* which introduce the notions of sampling compu-

tations and important events together with an inductive proof rule for verifying properties of hybrid systems. A more radical proposal is to use interval logic, which is the basis for the extended Duration Calculus by *Zhou, Ravn* and *Hansen*, and the model used by *Henzinger, Manna* and *Pnueli* in refining hybrid systems. It is also possible, as shown by *Lamport*, to extend a notation like TLA+ with explicit variables that denote continuous states and clocks and prove properties of hybrid systems. A similar approach has been taken by *Kurki-Suonio*.

An approach based on extensions of a Hoare style proof system to real-time programs is illustrated by *Hooman*, while *Nicollin, Olivero, Sifakis* and *Yovine* base their approach on extensions of CCS style process algebras. This leads to development of a model checking algorithm for timed state graphs, as developed for integration graphs by *Kesten, Pnueli, Sifakis* and *Yovine*. This work is closely related to the Hybrid Automata of *Alur, Courcoubetis, Henzinger* and *Ho*. This work is very important because it illuminates techniques that eventually may lead to automated support for analysis and even synthesis in the above-mentioned computer science related approaches.

The SIGNAL language introduced by *Benveniste, Le Borgne* and *Le Guernic* beautifully illustrates how synthesis can be supported by a simple, yet powerful design language. It would be an interesting exercise to relate abstract, logic based specifications to such a language which would allow automatic code generation for a controller.

Analysis and synthesis are at the heart of approaches that investigate how to extend differential equation methods for continuous dynamics to cover hybrid systems. But in fact little theory has been developed, apart from solutions of differential equations with discontinuous right hand sides and solutions of non-smooth variational problems. Nevertheless, from the point of view of simulation there is progress. Dynamical systems simulators intended to explore state space, if event driven, can be extended to yield similar phase portraits for hybrid systems as shown by *Guckenheimer, Back* and *Myer*.

"Mode switching" is the most commonly used design method for digital controllers of continuous plants. The plant state space is divided into regions (modes). Changing the control currently used is typically triggered by entering such a mode. For instance, an aircraft control system may have climbing, descending, and level flight modes, in which different control laws are used. Mode switching design is ad hoc for several reasons. The main reason is that it is a very complex mathematical task to identify the possible behaviours in the plant state space of even a small continuous dynamical system. Beyond that, identifying the effects of a proposed mode switching scheme is even more daunting. However, G. Sussman of MIT and his students have had success in using his dynamical system simulator not only to analyse plant state space but also to heuristically guess control schemes which will alter the plant state trajectories to meet performance requirements. Grossman and Myer have taken the approach of precomputing the mode changes resulting from a wide collection of pairs of controls and plant states. To go from one mode to a desired mode one does a table look-up for a suitable control. For real-time applications this leads to the

need for high speed database retrieval. Using a discrete event simulation point of view *Lemmon*, *Stiver* and *Antsaklis* introduce algorithms for identifying modes which are useful for control.

A fundamental problem is to find general procedures to extract digital control programs from system models and specifications. For compact convex optimization problems *Nerode* and *Kohn* use a relaxed calculus of variations to extract finite control automata, which guarantee an approximately optimal performance. In their paper on multiple agents this is done in a distributed control context. Kohn has implemented this system in his Declarative Control software. In conventional control theory, a fundamental form of stability is to insure that arbitrary small changes in control and input parameters do not lead to big changes in the resulting plant state trajectories. The usual definitions of stability are not applicable in hybrid systems because the control laws can be changed frequently. The "Models" paper by *Nerode* and *Kohn* proposes stability definitions based on continuity of system functions with respect to non-Hausdorff finite subtopologies of the usual topologies on the spaces of control theory.

Concepts familiar from control and systems theory can be carried over to hybrid systems. But the analogies of many familiar concepts from control systems have still not been worked out for hybrid systems. There are many representations to be studied, such as state space, input-output form, operators, linear representations on higher dimensional spaces, stochastic control, and Markoff process representation, etc. The paper by *Grossman* and *Larson* on hybrid flows introduces the observation space representation of hybrid systems, dual to the state space representation, and the relation to bialgebras.

The mode switching approach is illustrated by *Antsaklis*, *Stiver* and *Lemmon*. The examples of this paper and the one by *Blanke*, *Nielsen* and *Jørgensen* may hopefully be seen as a challenge by those who wish to test their theories. The same concern for application is the theme of the paper on Industrial-Scale Requirements Analysis by *Anderson*, *de Lemos*, *Fitzgerald* and *Saeed*, and the paper on Requirements Documentation by *Engel*, *Kubica*, *Madey*, *Parnas*, *Ravn* and *van Schouwen*.