Annual Reviews in Control 000 (2018) 1–7

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review article

Supervisory control of discrete-event systems: A brief history

W.M. Wonham[a], Kai Cai[b,*], Karen Rudie[c]

[a] Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada
[b] Department of Electrical and Information Engineering, Osaka City University, Osaka 558-8585, Japan
[c] Department of Electrical and Computer Engineering, Queen's University, Kingston, ON K7L 3N6, Canada

ARTICLE INFO

Article history:
Received 23 December 2017
Revised 11 March 2018
Accepted 21 March 2018
Available online xxx

Keywords:
Supervisory control theory
Discrete-event systems
History

ABSTRACT

This brief history summarizes the 'supervisory control of discrete-event systems' as it has evolved in the period 1980–2017. Overall, the trend has been from centralized or 'monolithic' control to more structured architectures, and from 'naive' to symbolic computation. Like any 'history' this one represents the perspective of the authors; in consequence some important contributions may have been overlooked or short-changed.

© 2018 Elsevier Ltd. All rights reserved.

Contents

1. Background and early motivation c.1980 . . . 1
2. Language controllability and monolithic supervisory control – 1981–1987 . . . 2
3. Confronting the computational challenge . . . 2
4. Language observability and monolithic control with partial observations . . . 3
5. Decentralized and distributed control with partial observations . . . 4
6. Extensions to SCDES with broader functionality . . . 5
7. Industrial applications . . . 6
Acknowledgment . . . 6
References . . . 6

1. Background and early motivation c.1980

The year 1980 is a convenient point of departure for supervisory control of discrete-event systems (SCDES) in the current sense. By that time the broad field known as 'modern' systems control that had evolved over the previous 20 years was well-established, on the basis of the five fundamental concepts of feedback, stability, controllability, observability, and quantitative optimality (Athans & Falb, 1966).[1] Of particular relevance later to SCDES were geometric concepts for regulator synthesis by feedback in linear multivariable systems (Basile & Marro, 1991; Wonham, 1985), namely the lattice of subspaces of a linear vector space, including controlled invariant subspaces, controllability subspaces, supremal (unique maximal) elements in the sense of partial order by subspace inclusion, and the concomitant notion of qualitative optimality. The dynamical systems to which systems control applied were generally those described by ordinary or partial differential equations and their discrete-time sampled-data counterparts, the main application drivers being the industrial process industries and various national space programs.

By contrast, discrete-event systems (DES) was an area apart, concerned with systems usually discrete in time and state space, driven by instantaneous events other than (or in addition to) the tick of a clock, and 'nondeterministic' in the sense of making state-transitional choices by internal chance or other mechanisms not necessarily modeled by the system analyst. Such systems were generally not amenable to the differential techniques of systems control. The application drivers included manufacturing, traffic, database management, and logistic systems. Owing to model complexity and analytical intractability, sys-

* Corresponding author.
E-mail addresses: wonham@ece.utoronto.ca (W.M. Wonham), kai.cai@eng.osaka-cu.ac.jp (K. Cai), karen.rudie@queensu.ca (K. Rudie).

[1] For the reader's convenience, in place of original sources textbooks or monographs may be cited where references to the former can be found.

https://doi.org/10.1016/j.arcontrol.2018.03.002
1367-5788/© 2018 Elsevier Ltd. All rights reserved.

tem simulation played a major role in analysis and optimization; indeed the term 'DES' seems to have originated with the simulation community and computer languages like SIMULA and SIMSCRIPT (Fishman, 1978). Theoretical analysis rested on queues (Ephremides, Varaiya, & Walrand, 1980), Markov chains (Howard, 1960), Petri nets (Peterson, 1981), and boolean transition structures (Aveyard, 1974); while design approaches exploited semaphores (Dijkstra, 1965), path expressions (Shields, 1979), and computer program representation by pseudo-code along with 'by-hand' cut-and-try alternating with informal verification (Ben-Ari, 1982). Formal attacks evolved in response, especially in software science, either expressed in process algebras such as *communicating sequential processes* (CSP) (Hoare, 1985), the *calculus of communicating systems* (CCS) (Milner, 1989), and work of the emerging Dutch school (Baeten, 2004); or proposed from a language perspective in terms of process behaviors (Arnold & Nivat, 1980; Beauquier & Nivat, 1980).[2]

In this literature control problems were certainly implicit, but formal synthesis (in the style of systems control) was broadly lacking. No standard paradigm existed analogous to optimal control, and there was often no clear separation of controller and uncontrolled 'plant'. The need (or at least the interest) was therefore apparent of a DES control theory which would (1) be discrete in time and space, asynchronous, event-driven as well as (possibly) clock-driven, and nondeterministic (supporting autonomous transitional choices); (2) rest on a simple control 'technology' and exploit standard control concepts; (3) be amenable to computation and applicable to the DES drivers (such as manufacturing); and (4) be accessible to practitioners and students of control engineering.

2. Language controllability and monolithic supervisory control – 1981–1987

The DES control theory which appeared in response (Ramadge & Wonham, 1982) was not startling. The plant was modeled internally as a finite state machine (FSM) for ease of physical interpretation and explicit computation;[3] plant external behavior was thus a regular language, which could optionally be considered the conceptual starting point independently of representation. Control specification was also modeled as an FSM, the corresponding regular language providing an upper bound on acceptable controlled behavior. The key advantages of this setting were its flexibility, broad expressiveness, and the technical feature of regular language closure under the boolean operations. The proposed control technology was simply a partition of the language alphabet (of *events*) into *controllable* and *uncontrollable*, the former amenable to disablement (prevention from occurrence) by a hypothetical external agent dubbed *supervisor*, the latter events not capable of direct disablement but presumed liable to occur (by chance, or internal system 'volition') whenever defined at the system's current state.[4]

The *supervisory control synthesis problem* was then formalized as that of designing a finite-state supervisor which, on observing the string of events generated by the plant, would at each state disable a suitable subset of controllable events to ensure that the generated controlled behavior (regular language) continued to satisfy the control specification, namely remained within the specification or 'legal' language. As in standard systems control, the approach was thus to separate the issues of problem definition and explicit computation.

Inasmuch as disabling all controllable events (in a sense, allowing as little as possible) could often yield a (trivial) formal synthesis, a concept of qualitative optimality was introduced requiring that the controlled behavior be as rich as possible (*maximally permissive* or *minimally restrictive*) subject to the specification constraint. Therefore the need arose to identify the subclass of (regular) languages which, for given plant and specification, could be synthesized as just described, and within which an optimal behavior could be shown to exist. Thus the final and key ingredient of the theory (Ramadge & Wonham, 1982; 1987b) was the concept of *controllable language*, and the crucial fact that the controllable sublanguages of a given (specification) language admitted a *unique* maximal (or *supremal*) element. More technically, the legal controllable sublanguages form an upper semilattice under the partial ordering of language (i.e. string subset) inclusion. The solution of the formal problem of optimal control is thus precisely the *top* element of this semilattice, or *supremal controllable sublanguage* (of the legal language). The latter was shown to be effectively computable, indeed polynomial in the state size of the legal language, by an algorithm we shall later call 'Supcon'. The conceptual debt to previous geometric regulation theory was evident: while in geometric regulation theory one synthesizes subspaces, in SCDES one synthesizes languages; the synthesis technique is to use feedback.

Not uniquely in the annals of interdisciplinary research (Gold, 1989), the community response to this proposal ranged from indifference to hostility. Computer specialists dismissed the engineering application of FSM and regular languages as trivial and/or nothing new, while control specialists regarded FSM as impractical and/or irrelevant: "Finite automata," declared one anonymous reviewer for a leading journal, "have no place in control engineering." Eventually the archival paper was accepted by a third journal as a putative contribution to 'optimal control' (Ramadge & Wonham, 1987b).

3. Confronting the computational challenge

Attempts to apply the new theory to industrial problems encountered the barrier notorious as *exponential state space explosion*.[5] Thus a workcell with N machines each having k states would be modeled as a plant with *a priori* state size $\sim k^N$, so 10 machines each with 5 states would result in a global model with state count $5^{10} \sim 10$ million. Naive *extensional* representation of such systems (whereby the transitions are all listed and stored explicitly) rapidly becomes infeasible. As a first response, researchers therefore turned to 'smart architectures' involving horizontal and vertical modularity, or in systems terms decentralized (Ramadge & Wonham, 1987a) and hierarchical (Zhong & Wonham, 1990) decompositions, later including distributed control by supervisor localization (Cai & Wonham, 2010a) (described below).

The decentralized approach (Ramadge & Wonham, 1987a) adopts the view that the plant behavior is restricted by multiple local control specifications. The approach thus synthesizes a decentralized supervisor to enforce each of these specifications separately. In benign cases the synthesized decentralized supervisors work 'cooperatively', and their parallel operations achieve the same controlled behavior as the centralized, or 'monolithic', supervisor did (to enforce the overall specification). In general, however, decentralized supervisors are myopic' and unaware of their global interaction; the result may be conflict that leads to livelock (block-

[2] While in several ways (Arnold & Nivat, 1980) foreshadowed the publications (Ramadge & Wonham, 1982; 1987b) cited below, it was not until 1992 that the respective authors became aware of each others' work. We thank Prof. Arnold for helping to clarify this connection, and retrospectively Dr. Angelo Bean for mediating between our two communities which, at that time, were mutually rather isolated.

[3] That the two areas of control theory and automata theory shared ideas in common had been recognized for some time (see e.g. Arbib, 1965).

[4] The term *supervisor* for such a 'disabling agent' was adopted to distinguish it from *controller* (conventionally a 'forcing agent'). Nevertheless, forcing action can effectively be modeled within the theory when needed.

[5] Formal complexity studies were reported in Gohari and Wonham (2000) and Rohloff and Lafortune (2005).

Please cite this article as: W.M. Wonham et al., Supervisory control of discrete-event systems: A brief history, Annual Reviews in Control (2018), https://doi.org/10.1016/j.arcontrol.2018.03.002

Download English Version:

https://daneshyari.com/en/article/7107747

Download Persian Version:

https://daneshyari.com/article/7107747

Daneshyari.com