MODEL-BASED DIAGNOSTICS FOR THE SUPERVISORY CONTROL OF MANUFACTURING SYSTEMS

R.A. Williams†, B. Benhabib† and K. C. Smith*

Computer Integrated Manufacturing Laboratory
Department of Mechanical Engineering
University of Toronto
5 King's College Road
Toronto, Ontario
Canada, M5S 1A4
e-mail: beno@me.utoronto.ca

†Department of Mechanical Engineering
*Department of Electrical Engineering

Abstract

Because of their nondeterministic nature of behaviour, the supervisory control of manufacturing systems must be carried out in closed loop. This trait greatly increases the size and complexity of the Discrete-Event System (DES)-based supervisory-controllers of manufacturing systems.

Attempting to cope with this complexity, a Hybrid Supervisory Controller (HSC), that distributes operations between a DES-based supervisory controller, an alternate mechanism, and a diagnostic system, was developed. The implemented diagnostic system had to be able to detect multiple faults, be robust, as well as integrate well with the other elements of the HSC. A model-based method derived from the General Diagnostic Engine [de Kleer and Williams, 1989] utilizing a set of models was developed and found to be suitable. Models are chosen and expectations of their predictions are adjusted according to the state of the equipment, the operations they are performing, and the location of parts.

Introduction

Background and Motivation

The design of a supervisory controller entails the formulation of control laws, and the synthesis of supervisors. The laws specify how the supervisor is to react to the behaviour of the manufacturing system, the goal being to have some production specifications satisfied within the standing control enforcement constraints.

Petri nets [Long *et al.*, 1992b], [David, 1991], knowledge engineering [Benhabib *et al.*, 1989, Camarinha-Matos and Steiger-Garcao, 1986a], timed transition models, real-time temporal logic [Ostroff, 1990, Ostroff and Wonham, 1990], and controlled automata [Brandin *et al.*, 1991] have been used as supervisory controllers for manufacturing systems. Supervisory controllers designed around Discrete-Event Systems (DES) theory have the desirable feature that their behaviour may be proved and verified correct before implementation using the tools developed within the theory.

However, the control of even moderately complex systems can easily require an immensely large DES strategy [Ho, 1987]. This necessitated the development of a hybrid approach that uses some alternate mechanism in addition to a DES supervisory controller, which would relieve the latter of the need for so many states by (i) taking on the responsibility for some of the control objectives, and (ii) asserting control whenever events diverge from the significantly reduced number of states of the DES supervisory controller.

The tasks of the supervisory controller include the detection of failures and recording of repairs to equipment, which are part of the job of (i) monitoring of the workcell, (ii) processing and analyzing large quantities of sensory input, and (iii) confirming the supervisor's hypothesis about the state of the workcell. These are the responsibilities of an on-line diagnostic system which forms part of the supervisory controller.

This paper will focus on the application and development of a diagnostic system, based on the General Diagnostic Engine (GDE) of de Kleer and Williams [de Kleer and Williams, 1987], that meets the needs of a DES-theory based supervisory controller. The environment in which the diagnostic system will operate is described briefly as well[1].

[1]For a more complete description of the rest of the supervisory controller, see [Williams *et al.*, 1994].

A New Approach

The diagnostic system described in this paper is one of three main elements belonging to our Hybrid Supervisory Controller (HSC), Figure 1:

1. DES Supervisor, which contains the nominal supervisory-control strategy,
2. Diagnostic system, and
3. Alternate-Strategy Driver (ASD), which generates alternate part routes when needed.

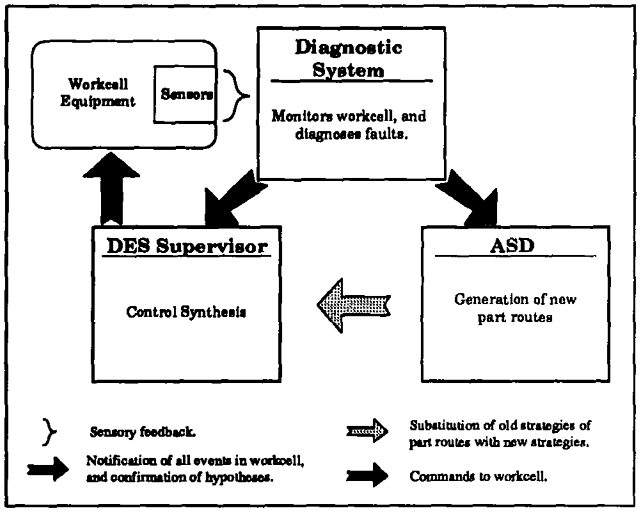

Figure 1: Overview of the HSC.

The diagnostic system interprets sensory data. It feeds its interpretation to the DES Supervisor and the ASD. If the ASD determines that new (alternate) part routes are needed, it derives them and submits them to the DES Supervisor. The DES Supervisor reacts to the information received from the diagnostic system, accepts the alternate routes from the ASD, and proceeds to synthesize control commands.

DES Supervisor

Representation of a Controlled DES

A DES is modeled as a generator of a formal language [Ramadge and Wonham, 1989]. The generator consists of states with transitions between them. If the generator is a model of a system being supervised, it is called a plant. The DES-based supervisor exerts control on a plant by disabling certain events that the plant can generate or accept. The supervisor enables events in accordance to logic specifications depending upon the current state. This process is called control synthesis.

Generating the DES Supervisor

The DES Supervisor for our workcell is synthesized, using the tools provided by the DES theory, from (i) a plant model for the workcell, (ii) specifications for the routes of the parts, referred to as *linear-part-routing* specifications, and (iii) specifications that enforce other constraints, referred to as safety specifications[2]. The workcell plant is constructed from a set of individual equipment models[3] (i.e. plants). Two types of plants are defined in this paper: *Transport plants* (e.g., robots and conveyors) and *Stationary plants* (e.g., machining-centers and part buffers). Each plant is a generator for a set of events. Significant features of the equipment related to supervisory control are captured in the plant definitions in the form of state-transition diagrams.

The New DES-Supervisor Design

The DES Supervisor within the HSC consists of two modular supervisors, Figure 2. The first is synthesized from the safety specifications. The second is a conjunction of linear-part-routing specifications, which are treated as modular (sub)supervisors. Only the linear part-routing specifications are re-generated during run-time. The safety specifications are not changed during run-time, thus, neither is their modular supervisor.

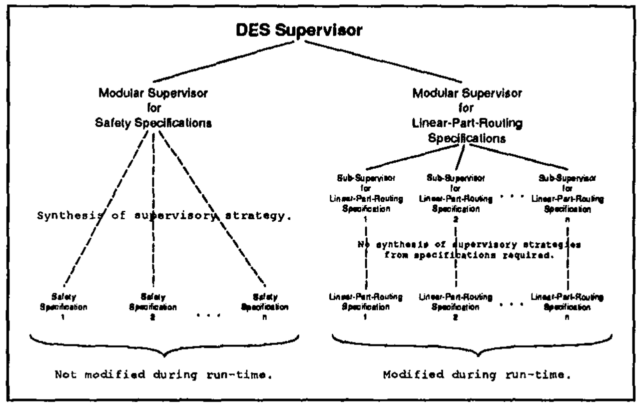

Figure 2: Design of DES Supervisor.

Implementation of the DES Supervisor

The strategy of the two modular supervisors of the DES Supervisor is encoded in look-up tables of event descriptions. The tables for the modular supervisor for the safety specifications were embedded into the computer code of the implementation. The tables for the modular subsupervisors for the linear-part-routing specifications remains in the form of tables in order that their contents may be revised during run-time.

Control synthesis is enforced in a two-step procedure. First, a set of events is enabled according to the strategy of the DES Supervisor. Second, events are processed after their occurrence by making the appropriate transi-

[2]All plants and specifications are discussed in detail in [Williams *et al.*, 1994].

[3]These models are distinct from those utilized by the diagnostic system.

tions in the tables, automatically updating the current hypothesis about the state of the system.

Diagnostic System

The GDE

Decision trees, fault dictionaries, and expert systems are commonly employed in diagnostic systems for Flexible-Manufacturing Systems (FMSs) [Abu-Hamdan and El-Gizawy, 1992]. However, according to Davis and Hamscher [Davis, 1984] a model-based approach may often be the most suitable. The General Diagnostic Engine (GDE) of de Kleer and Williams [de Kleer and Williams, 1987] is one of a handful of available model-based approaches [Williams *et al.*, 1994]. It is, unlike many others, capable of handling multiple faults, and does not require fault models. As well, its design allows its easy integration with the rest of the HSC. Thus, the basic GDE reported in [de Kleer and Williams, 1987] was chosen as the prototype for the means of meeting the needs of the HSC in regards to failure detection.

The decisions of the GDE are based on the knowledge that a device must be faulty, if its behaviour is inconsistent with its model. This interaction of observation and prediction forms the basic paradigm of model-based reasoning for diagnosis, Figure 3. Symptoms of faults are detected, and faults are isolated by utilizing this relationship.

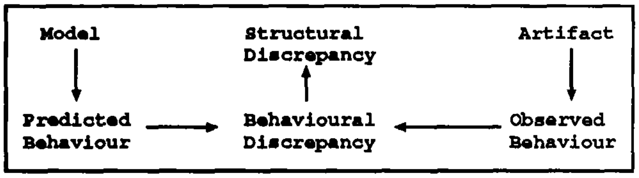

Figure 3: Diagnosis as the interaction of observation and prediction.

In modeling the workcell, it is generally assumed that the workcell is a single device. The plants within the workcell are the components of this device. The components of the plants are the sub-components of the device. This hierarchy continues downwards until the lowest level of abstraction, Figure 4. Connections are used for modeling interactions between the components of the device (i.e., the workcell), and sub-components of the components [4].

Notably, the workcell model automatically defines the minimal subset of sensors to monitor. This ensures the efficient use of sensor resources.

[4]Since in our case interactions are possible only between the transport plants and the stationary plants, no connections exist amongst the transport plants, nor amongst the stationary plants. This point will be discussed further in the next section

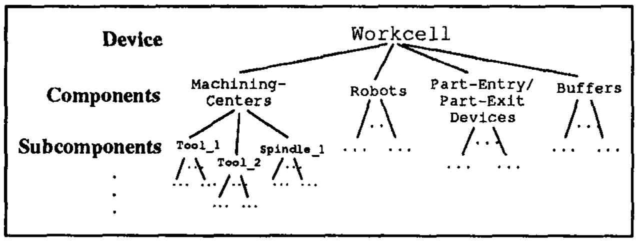

Figure 4: Hierarchical structure of workcell device model.

Implementation of the Diagnostic System

A simple (model-based) diagnostic system based on the generic General-Diagnostic-Engine (GDE) approach was developed and used in our work. The new system lacks some of the features of the GDE, and opts for several simple models rather than one universal model. It preserves some of the most beneficial aspects of the generic GDE approach and allows for the implementation of the "full-blown" GDE in the future. Modeling is still based on knowledge of the structure and behaviour of the devices, even though the exact formalism of de Kleer and Williams [de Kleer and Williams, 1989] was not used. Hence, the models are still easier to construct and to maintain than they would be otherwise. Hierarchies in the models are permitted. Both approaches are robust in the face of unknown faults, since neither makes predictions based on knowledge of the behaviour of faulty components; they assume a device is faulty when its behaviour is inconsistent with its model. Both handle multiple faults easily and elegantly.

Simplification of the Diagnostic Mechanism

The diagnostic mechanism[5], was reduced in complexity in two steps. In the first step, the need for a GDE and Assumption-Based-Truth-Maintenance-System (ATMS) was eliminated. This step did not relinquish the need for a model. However, it allowed the architecture of the model to be simplified, which was the second step.

A workcell controller, specifically the Alternate-Strategy Driver (ASD) in our case, would require knowing only which plants have failed. Therefore, during run-time when a failure occurs, it is only necessary to determine which plant contains the fault. For the case of a plant operating by itself, this is a relatively simple task. If the plant begins exhibiting behaviour that results in a discrepancy, it is clear that that plant has failed. No fault search is needed to infer this. Hence, no fault search mechanism is needed for isolating the fault. Therefore, in this instance, no ATMS or GDE is needed. All that is needed is:

1. A model of each individual plant, which presumes the plant is operating in isolation, and

[5]Not presented here. See [de Kleer and Williams, 1987].

2. A simple mechanism to propagate values, but not environments, through the model.

A model would be needed for each plant in a workcell.

There is a cost to this simplification. Plant interaction now must be handled explicitly using special models. A model would be required for each pair of transport plant and stationary plant, although not for every combination of plants. It is assumed that stationary plants cannot interact, nor will two transport plants be permitted to work together or with the same stationary plant simultaneously, so no models are required for such combinations. In the likelihood that one of these events occurs, it is hoped that the employed models would interpret the interaction of the involved plants as anomalous behaviour.

The second step in simplifying the diagnostic mechanism is to simplify each model. The models need to be able to determine a set of outputs from a set of inputs in order that the initial discrepancy may be detected. This requires a causal representation of the device and its components. The source of the fault is implied by the model that detected that fault. Thus, an (inference-logic based) mechanism is no longer required for isolating the source of a discrepancy; no inference logic-representations are needed in the models.

The result of the simplification of the diagnostic mechanism is a library of simple models for different scenarios.

Our (simpler) diagnostic approach has no inference engine, so it is impossible for it to isolate the source of a fault through some inference process. Unless it is obvious that a fault has occurred in one of the plants, it must be assumed that all plants have failed. Thus, with the simpler approach, we have lost some of the power to resolve.

Fault Detection

The process of fault detection is a comparison of the current hypothetical situation with the true situation. As was discussed earlier, each state of a DES Supervisor defines which plants are operative, where each part is, which part each plant is operating on, etc. Thus the current state of the DES Supervisor provides the hypothesis. The task at hand is to verify that the hypothesis agrees with reality. The implementation problem now becomes one of knowing which models to apply and what information to provide them. Again the current state of the DES Supervisor is used to resolve those issues.

Fault detection requires performing two tasks simultaneously: failure detection and part-location verification [Williams *et al.*, 1994].

(a) *Failure Detection*: The proposed failure-detection strategy proceeds through three stages. In the first stage, all parts that are in the workcell are considered sequentially. All plants deemed to be operating on or in possession of said part are checked. The number of plants in possession or operating on the part, and the type of operation, among other factors, affect the choice of models on which the evaluations are based, as well as the expectations of their predictions. All remaining *idle* stationary and transport plants are checked in the second and third stages, respectively.

(b) *Verification of Part Location*: The objective is the verification of the location of the parts, plus the identification of the plant on which the part resides for the instance where the observibility of the DES state specifications are such that it is permissible for a part to be on one of several plants.

Operation within the HSC

The diagnostic system would initiate proceedings with the rest of the HSC whenever plants fail (including misplaced parts), or are repaired.

The sequence of events upon detection of a plant failure are as follows:

1. Detection of failures.
2. Generating a list of the failed plants.
3. Sending corresponding event signals to the DES Supervisor.
4. Sending a list of the failed plants to the ASD along with a list of the locations of the parts in the workcell. In response, the ASD —
   (a) Disables all events of the DES Supervisor that start an operation. This ensures that no new production operations are begun.
   (b) Derives new linear-part-routing specifications.
   (c) Asks diagnostic system for confirmation of the current state of the workcell.
   (d) Transfers the new linear-part-routing specifications, and the new current states of those specifications to the DES Supervisor.
   (e) Returns control to the DES Supervisor.

The process upon the repair of a plant proceeds as follows:

1. Generating a list of the repaired plants.
2. Sending corresponding event signals to the DES Supervisor.
3. Sending the list of repaired plants to the ASD.
4. Repeating the Steps 4(a) to 4(e) of the procedure for dealing with plant failures discussed above.

Alternate-Strategy Driver

A scheduled production plan might have to be revised during run-time in response to the occurrence of unplanned events. However, the re-scheduling problem cannot be solved to (global) optimality due to time restrictions. Instead, only a portion of the scheduled production plan can be modified. The routing level is the most amenable to, and capable of, delivering quick responses to the altered conditions. So, herein the efforts of re-scheduling the production plan are focused on the routing level. In this context, the ASD re-routes parts by replacing their linear-part-routing specifications in the DES Supervisor with ones that constitute the new routes.

Choosing a Re-routing Strategy

A "mixed" re-routing strategy was developed in our work. It combines many of the positive attributes of the off-line and on-line approaches reported in the literature [Rodammer and White, 1988, Tang and Denardo, 1988, Shaw, 1988]. Specifically, however, the works of Camarinha and Garcao [Camarinha-Matos and Steiger-Garcao, 1986b], Long, et al. [Long *et al.*, 1992a], and Hadavi, et al. [Hadavi *et al.*, 1990] had the greatest influence on our work.

In our mixed approach, the "best" routes[6], referred to as the nominal routes, represent the original routes of the scheduled production plan. They are assumed to have been provided to the HSC. Deviation from a nominal route is allowed whenever it cannot be maintained. This occurs under circumstances of unscheduled events, or deadlocks. "Complete routes" are generated to replace nominal routes that cannot be maintained.

The Algorithms for Re-Routing

The ASD produces a new linear-part-routing specification in three steps:

1. *Maintenance of look-up tables of performable transport operations.* An entry exists in each table for each ordered-pair of plants — one table per transport plant.
2. *Compilation of part-production routes.* A manufacturing operation is selected and paired with a stationary plant that will perform it. An appropriate path, formed of performable transport operations, is retrieved from the tables. These are joined into a production-stage entity. A part-production route is created by linking production-stage entities.
3. *Conversion of the part-production route into a linear-part-routing specification.* Done only if the derived route contains non-nominal operations.

Simulation of the HSC

The HSC was implemented in the C language and tested via simulation on a Sun Sparc Workstation. The example workcell considered (Figure 5) consisted of two robots, two machining-centers, two buffers, a part-entry device, and a part-departure device. Each part to be produced had its own production plan listing production goals and alternatives.

The nominal route for the production of one of the parts was defined as follows:

[6] A route is a path of production (i.e., a sequence of manufacturing and transport operations) through the resources of the manufacturing system.

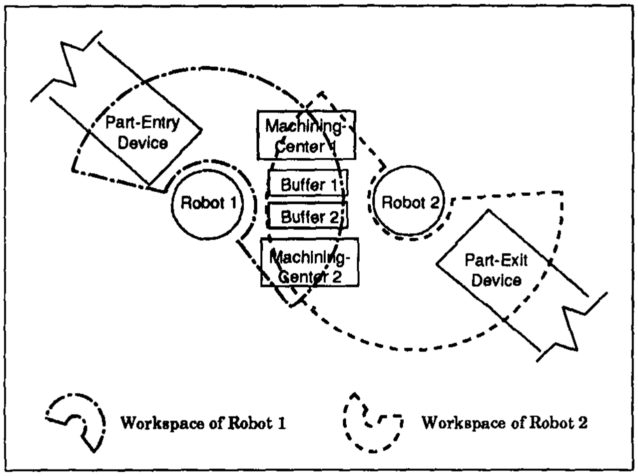

Figure 5: Arrangement of workcell.

1. Part arrives at Part-Entry Device.
2. Robot 1 moves the part to Machining-Center 1.
3. Machining-Center 1 performs a milling operation.
4. Robot 1 moves the part to Machining-Center 2.
5. Machining-Center 2 performs drill operation #1.
6. Machining-Center 2 performs drill operation #2.
7. Robot 2 moves the part to the Part-Exit Device.
8. Part departs the workcell.

Conclusions

The work presented in this paper is original in its application of a model-based technique to the run-time diagnosis of a workcell. Notable of our approach is the utilization of several different models rather that one universal model. Models are customized to their range of application. This permits them to be significantly simplified. As well, the model-based technique automatically defines a strategy for optimal sensor usage.

During simulations the production of parts was exposed to failures of equipment. Tests showed that: (i) the diagnostic system can successfully detect faults, (ii) it operates successfully with the rest of the HSC, and (iii) the ASD and DES Supervisor can successfully perform their duties.

A diagnostic system incorporating the inference capabilities of the GDE would have had the potential to be better at isolating failures. Despite the lack of such properties in our implementation, our diagnostic system performed satisfactorily. Nonetheless, the implementation does allow for easy incorporation of the full GDE.

Little work has been done developing models of robots, and other artifacts of the workcell for model-based diagnostic methods. This suggests an area for further research. This paper sets the context within which those models would exist.

References

[Abu-Hamdan and El-Gizawy, 1992] M.G. Abu-Hamdan and A.S. El-Gizawy. "An Error Diagnosis Expert System for Flexible Assembly Systems". In *Preprints, 7th IFAC Symposium on Information Control Problems in Manufacturing Technology*, pages 451–456, Toronto, Canada, 1992.

[Benhabib *et al.*, 1989] B. Benhabib, C.Y. Chen, and W.R. Johnson. "An Integrated Manufacturing Workcell Management System". *Manufacturing Review*, vol. 2, no. 4:266–276, 1989.

[Brandin *et al.*, 1991] B.A. Brandin, W.M. Wonham, and B. Benhabib. "Discrete Event System Supervisory Control Applied to the Management of Manufacturing Workcells". In V.C. Venkatesh and J.A. McGeough, editors, *Computer-Aided Production Engineering*, pages 527–536. Elsevier, 1991.

[Camarinha-Matos and Steiger-Garcao, 1986a] L.M. Camarinha-Matos and A. Steiger-Garcao. "Robotic Cell Programming: A Knowledge-Based Approach". In *Robotics and Artificial Intelligence 86*, pages 533–551, 1986.

[Camarinha-Matos and Steiger-Garcao, 1986b] L.M. Camarinha-Matos and A. Steiger-Garcao. "Robotic Cell Programming: A Knowledge-Based Approach". In *RAI/IPAR '86*, pages 533–551, Toulouse, France, 1986.

[David, 1991] R. David. "Modeling of Dynamic Systems by Petri Nets". In *ECC91 European Control Conference*, pages 136–147, Grenoble, France, 1991.

[Davis, 1984] Randall Davis. "Diagnostic Reasoning Based on Structure and Behaviour". *Artificial Intelligence*, vol. 24:347–410, 1984.

[de Kleer and Williams, 1987] Johan de Kleer and Brian C. Williams. "Diagnosing Multiple Faults". *Artificial Intelligence*, vol. 32:97–130, 1987.

[de Kleer and Williams, 1989] Johan de Kleer and Brian C. Williams. "Diagnosis with Behavioral Modes". In *Proc. 11th Int. Joint Conf. on Artificial Intelligence*, pages 1324–1330, Detroit, MI, 1989.

[Hadavi *et al.*, 1990] K. Hadavi, M.S. Shahraray, and K. Voigt. "ReDS - A Dynamic Planning, Scheduling, and Control System for Manufacturing". *Journal of Manufacturing Systems*, vol. 9, no. 4:332–344, 1990.

[Ho, 1987] Y-C. Ho. "Performance Evaluation and Perturbation Analysis of Discrete Event Dynamic Systems". *IEEE Transactions on Automatic Control*, vol. AC-32, no. 7:563–572, 1987.

[Long *et al.*, 1992a] J. Long, B. Descotes-Genon, and P. Ladet. "Distributed Intelligent Control and Scheduling of Flexible Manufacturing Systems". In *8th International Conference on CAD/CAM, Robotics and Factories of the Future*, volume 2, pages 1760–1767, Metz, France, 1992.

[Long *et al.*, 1992b] J. Long, B. Descotes-Genon, and P. Ladet. "Hierarchical and Intelligent Control of Flexible Manufacturing Systems". In *Preprints, 7th IFAC Symposium on Information Control Problems in Manufacturing Technology*, pages 243–248, Toronto, Canada, 1992.

[Ostroff and Wonham, 1990] J.S. Ostroff and W.M. Wonham. "A Framework for Real-Time Discrete-Event Control". *IEEE Transactions on Automatic Control*, vol. 35, no. 4:386–397, 1990.

[Ostroff, 1990] J.S. Ostroff. "Deciding Properties of Timed Transition Models". *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, no. 2:170–183, 1990.

[Ramadge and Wonham, 1989] P. Ramadge and W.M. Wonham. "The Control of Discrete Event Systems". In *Proc. of the IEEE*, volume 77, no. 1, pages 81–98, 1989.

[Rodammer and White, 1988] F.A. Rodammer and K.P. White. "A Recent Survey of Production Scheduling". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 6:841–851, 1988.

[Shaw, 1988] Michael J. Shaw. "Dynamic Scheduling in Cellular Manufacturing Systems: A Framework for Networked Decision Making". *Journal of Manufacturing Systems*, vol. 7, no. 2:83–94, 1988.

[Tang and Denardo, 1988] C.S. Tang and E. Denardo. "Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches". *Operations Research*, vol. 36, no. 5:767–777, 1988.

[Williams *et al.*, 1994] R.A. Williams, B. Benhabib, and K.C. Smith. "A Hybrid Supervisory Control System for Flexible Manufacturing Workcells". *1994 IEEE International Conference on Robots and Automation*, San Diego, California, May 1994. In print.