

REAL-TIME CONTROL OF A MANUFACTURING CELL USING KNOWLEDGE-BASED SIMULATION

S. Manivannan
Jerry Banks

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332

ABSTRACT

Issues related to real-time control of a manufacturing cell using knowledge bases and simulation are discussed. The need for integrated simulation environments for modeling and analysis is advanced. A mathematical framework based on higher level abstractions representing entities, attributes, constraints, and other components of manufacturing cells is presented. A manufacturing cell has been modeled to illustrate the framework and real-time control issues using knowledge-based simulation. A temporal knowledge base has been designed to synchronize the events and their times of occurrence in both the manufacturing cell and the simulation model. Also, a dynamic knowledge base has been implemented using frame structures for storing the results of simulation. This feature provides a faster response to a control problem by reducing the number of resimulations conducted for evaluating various alternative policies in real time.

1 INTRODUCTION

Real-time control of a manufacturing cell is an increasingly important area of application for knowledge-based systems. Within the context of manufacturing, real-time control systems provide decisions for a specific problem associated with part processing, material movement, inspection, storage and distribution, etc. A real-time controller should be capable of (i) reacting to a problem instantaneously, (ii) evaluating several alternative policies, (iii) providing optimum or near optimum solutions, (iv) learning from the previous problems and (v) providing faster and more accurate solutions. Three major issues in accomplishing real-time control are data acquisition, quick response, and instantaneous feedback.

Applications of knowledge-based control systems are growing in alarm monitoring, fault diagnosis and maintenance scheduling to mention a

few areas (Krijgsman, *et al.* 1988; Manivannan and Banks, 1990). The response from a knowledge-based system must be extremely fast for real-time control and hence a strong emphasis is put on minimal computation times. Further, when the knowledge-based system is used as a controller, it has to make a decision at every sampling point. This requirement implies that the knowledge-based system must be capable of dynamic modification. Also, it must be embedded in a structure which allows cooperation with real-time software for data acquisition and performance evaluation. The main emphasis of most of the existing knowledge-based systems is not on time-critical behavior, but on explanation systems and handling a large number of rules.

In this paper, we discuss a framework for accomplishing real-time control. The framework integrates data collection devices with a dynamic knowledge base and a simulation model (Manivannan and Banks, 1990). The framework is known as real-time knowledge-based simulation (RTKBS). Using abstract terminology, we define modeling, analysis, and real-time control within the RTKBS framework. The issue of synchronizing various events and their times of occurrence in both the simulation and actual manufacturing cell is presented. Response time reduction for real-time control has been accomplished by storing the results from the previous simulations in a dynamic knowledge base. A detailed framework of RTKBS describes its operating principles. The RTKBS system is applied to a manufacturing cell connected by a variety of material handling systems.

2 RESEARCH IN REAL-TIME CONTROL

Several authors have reported the issues of building an intelligent controller for managing operations within a manufacturing cell (Voss, 1988; Krijgsman *et al.*, 1988). The progression from first to second generation of knowledge-based

controllers highlighting their advantages and difficulties is discussed in (Simmonds, 1988). Integration of simulation and knowledge bases is performed to solve a variety of control problems in manufacturing cells. A generalized framework for controlling a manufacturing shop floor is reported in (Manivannan and Banks, 1992). Several applications for combining shop floor data with intelligent control systems are reported in (McLean, 1986; Furlani *et al.*, 1988). An expert system software to diagnose failures and recommend procedures to correct those failures is reported in (Cantone and Caserta, 1988). A knowledge-based simulation system to diagnose failure propagation within a CNC turning center using real-time sensory information is discussed in (Manivannan and Banks, 1990).

Several integration tools for real-time control of flexible manufacturing systems (FMS) are reported in the recent literature. A control structure for managing an FMS using an AI-based, opportunistic, hierarchical planning architecture is described in (Young and Rossi, 1988). The framework uses knowledge-based simulation and a control mechanism for part routing in an FMS. A prototype real-time controller for scheduling an FMS is discussed in (Kachitvichyanukul, *et al.*, 1986). Another expert system for controlling and scheduling an FMS is described in (Ben-Arieh, Moodie and Nof, 1985). A knowledge-based on-line simulation to control a flexible manufacturing cell is reported in (Manivannan and Banks, 1992).

framework is to provide an integrated environment for the controller to evaluate various control policies using simulation. Data for simulation is collected from the manufacturing cell. One of the major difficulties of accomplishing real-time control with simulation is the synchronization of various events and their times of occurrence in the real system with those of the model. Another difficulty is responding to a control problem in real time, as simulation is time consuming. Evaluating each and every alternative for a control problem means executing several simulation runs. This increases the time required for the RTKBS system to respond to a control problem. If the response time is excessive, real time control is not possible.

In order to accomplish real-time control, a framework has been designed to integrate the data from the manufacturing cell with the simulation events. The times of occurrence of various events in the cell and simulation are synchronized using a temporal knowledge base. This knowledge base contains procedural rules for processing the time information collected on actual events occurring in the manufacturing cell. Data collection occurs only for those events that are being considered in the simulation model. For instance, if the simulation model includes a transporter, then the data acquired from the cell includes the arrival, departure, breakdown and repair times of the transporter. The temporal knowledge base has been partitioned into rules for handling past, current, and future events.

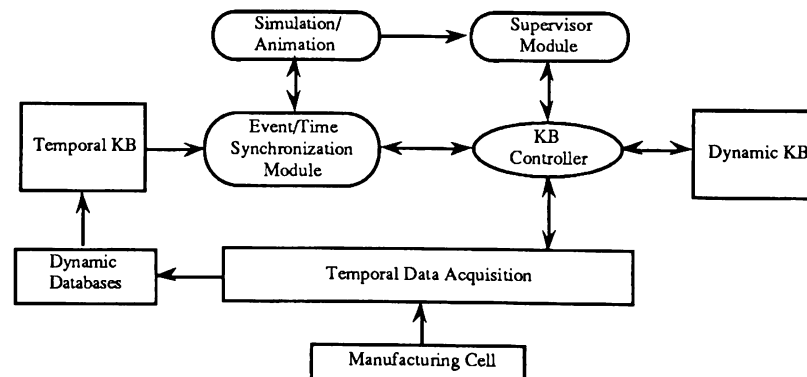


Figure 1. Overall Framework of RTKBS System

3 CELL CONTROL USING RTKBS

3.1 An Overall Framework of RTKBS

A general purpose discrete-event simulation language and a knowledge base are utilized in designing a framework for real time control of a manufacturing cell. An overall framework of the RTKBS (real-time knowledge-based simulation) is indicated in Figure 1. The purpose of this

A simulation model of a manufacturing cell has been created and linked to an event/time synchronization (ETS) module. The ETS module modifies the events and their times stored in a time flow mechanism (or the event calendar) based on the temporal data obtained for the corresponding events and event times occurring in the manufacturing cell. This synchronization offers a capability to emulate the manufacturing cell in real

time using simulation and knowledge bases. A supervisory interface is created to interact with the simulation model and perform real-time simulation for controlling the manufacturing cell by evaluating alternative scenarios. The ETS module ensures that the events and event times used in simulating these alternative scenarios are in agreement with the corresponding values in the manufacturing cell.

The RTKBS includes a dynamic knowledge base to store the results obtained after evaluating various scenarios simulated for each control problem. This feature reduces the response times in conducting simulations. Thus, the RTKBS framework provides a real-time knowledge-based simulation environment to control manufacturing operations. It also has the capability of analyzing several design alternatives by modifying the input parameters. The forward simulation has the advantage of using data that completely depicts the operation of the actual cell.

3.2 An Abstract Representation of RTKBS

In this section, we represent the RTKBS framework using mathematical abstractions for the purpose of formalizing the modeling, analysis and control within the RTKBS framework. Consider a manufacturing cell M_s that has the potential to self-learn, which operates in an environment E_s , requires various forms of inputs as represented by $[I_{ms}]$ and generates one or more output(s) $[O_{ms}]$. For instance, $[I_{ms}]$ can be the number of parts transported by an automated guided vehicle from a processing station TP_i to the next station TP_j , the number and type of vehicles currently available at a process station TP_k and so on. We define the following terminology to represent various items in order to perform modeling, analysis, and control of the manufacturing cell:

- (a) $[C_i(r,g)]$ is a set of policies used to perform various planning and control activities.
- (b) $[KB]$ is the knowledge base consisting of the factual and procedural knowledge of M_s . The knowledge base is structured and represented by frames that interact with various simulation models created to perform analysis and control.
- (c) $[g]$ specifies a set of user-defined goals that determine the performance measures to be selected during the model building phases.
- (d) For each $[C_i(r,g)]$ there is a plan $[r]$ consisting of a set of executable frames.
- (e) $[r] = [O_1, \dots, O_k]$ is a frame set selected from the knowledge base with a selection depending on M_m, M_s and $[g]$.

- (f) $[PM_i(r,g)]$ is a set of performance measures that need to be controlled (and determined through modeling and simulation phases) that depend upon $[r]$ and $[g]$. $[PM_i(r,g)]$ may consist of one or more items in the set. $[PM_i(r,g)]$ can have either 0 or 1 values, a fixed quantity, or minimum and maximum levels for a performance measure i . For instance, $PM_i(r,g)$ can be the average number of items waiting to be moved during a certain time period represented as minimum/maximum levels, a manufacturing vehicle is in busy/idle mode (0 or 1 value) or number of items transferred to pre-specified destination stations after processing.

Using these abstractions and notations we can define modeling, control and analysis within the RTKBS framework. The interaction between these abstract terms, the manufacturing cell, and the simulated cell is shown in Figure 2.

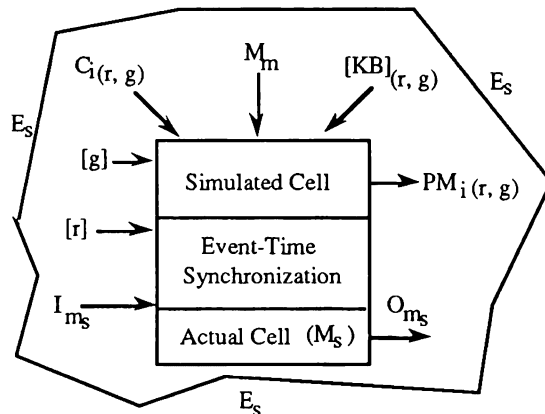


Figure 2. Interaction Between Various Abstract Terms

Activity 1: MODELING:

For a given manufacturing cell M_s in E_s that uses $[I_{ms}]$ and generates $[O_{ms}]$ for each of the selected $PM_i(r,g)$ using control policies $C_i(r,g)$, the objective is to determine the plan $[r]$ selected from the knowledge base $[KB]$ for a modeling method M_m such that the selection leads to accomplishing the goals $[g]$ of M_s . This activity is performed by the designer, planner and scheduler of the manufacturing cells.

Activity 2: CONTROL

For a given manufacturing cell M_s in E_s that has generated $[O_{ms}]$ for each of the selected $PM_i(r,g)$ using control policies $C_i(r,g)$, the objective is to determine $[I_{ms}]$ given the plan $[r]$ selected from the knowledge base $[KB]$, the modeling method M_m

and the goals [g] of M_s . This activity is performed by the controller in managing the operations of the cell.

Activity 3: ANALYSIS

For a given manufacturing cell M_s in E_s that uses $[I_{ms}]$ and control policies $C_i(r,g)$, the objective is to determine $[O_{ms}]$ for each of the selected $PM_i(r,g)$, using the plan [r] selected from the knowledge base [KB], the modeling method M_m and the goals [g] of M_s . This activity is performed by the planners and schedulers of the manufacturing cells.

The RTKBS framework has been designed to perform all three activities. The modeling is performed off-line, however, the control and analysis are conducted in real time. The structure of the [KB] is flexible and modular to accommodate changes in the manufacturing cell M_s , plans [r], and goals [g]. The modeling method is restricted to simulation and the control policies $C_i(r,g)$ are provided by the supervisor (human in the loop) or the controller of the manufacturing cell.

3.3 Event Synchronization in RTKBS

Discrete event simulation languages often utilize event lists (also known as an event calendar) in managing current and future events during the execution of a model. The future event list is an ordered set of events according to either time or priority. In order to perform real-time control using RTKBS concepts, the events and their times of occurrence in the future event list should be synchronized with corresponding events taking place in the manufacturing cell. This is known as event/time synchronization. This is necessary for the following reasons: (i) unforeseen delays and early arrivals of entities in the real system could cause a discrepancy with their event times in simulation, (ii) time sharing within a computer during simulation, and (iii) errors in data used in modeling and simulation.

Event/time synchronization can be carried out in three ways. The first method involves obtaining data continuously and connecting the data collection devices to an input data processor with the simulation software. In this case, the simulation model will be data driven. The approach is restricted to process industries in which the system status and control variables change continuously. However, in the case of discrete parts manufacturing, an event occurs at a specific point in time, and continuous data gathering is expensive and impractical.

The second approach involves developing a simulation model for each of the entities, resources, and queues. Data collection is restricted to those entities, resources, queues, and events that alter temporal information. In such cases, the simulation model needs to be built in a modular fashion separating entities, queues, and resources. This approach is advantageous in terms of data collection. However, it poses difficulty in managing several simulation models constructed for various entities and resources.

The third method involves use of past, future and current event lists. This approach essentially works with the event lists and does not affect the modeler or the analyst. In RTKBS, these past, future and current event lists in conjunction with the data collection devices are designed to handle the synchronization of events in both simulation and in the manufacturing cell. The synchronization is carried out by collecting data from the cell continuously and modifying the event times in the future and past event lists.

In RTKBS, the event/time synchronization is performed by the ETS module. In designing the ETS module, we need to represent the temporal information generated during simulation and in the manufacturing cell. The representation should handle both time points and time intervals associated with the past, current and future occurrences of events. The structure for a dynamic database for performing event-time synchronization is indicated in Figure 3.

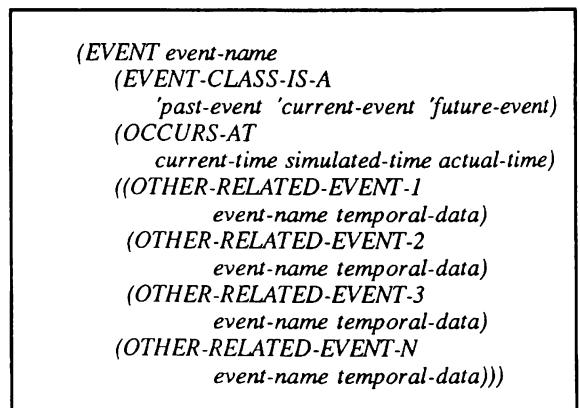


Figure 3. Structure of ETS Module

For instance, if the next event in the future event list during simulation is arrival of an agv to a machine, then the actual time of occurrence of that event in the manufacturing cell is compared with that of the simulated time. All those events that are related to this specific event are affected and need rescheduling. The example shown in Figure 4 illustrates the situation.

```

(EVENT (ARRIVAL-OF agv#1
        DESTINATION machine#6)
 (EVENT-CLASS-IS-A 'past-event)
 (OCCURS-AT
  200.00 200.00 195.00)
 (OTHER-RELATED-EVENT-1
  (unload machine#6 197.00)
  (load AGV#1 198.2)
  (load machine#6 199.00)))

```

Figure 4. Need for Event Rescheduling

Temporal data from the cell is continuously obtained via manual or automated data collection systems and stored in a dynamic database. During simulation, the next event is selected from a future event list. The event and the event time are compared with the real time data received from the cell. Let AT_x be the time of occurrence for an event in the cell and ST_x be the corresponding value from the future event list during simulation. The ETS module utilizes three temporal rules in evaluating these events and their times as shown in Figure 5.

```

If  $AT_x$  is equal to  $ST_x$ 
Then (EVENT-CLASS-IS-A 'current-event)
     (ETS 'not-required))

If  $AT_x$  is less than  $ST_x$ 
Then (EVENT-CLASS-IS-A 'past-event)
     (ETS 'required)
     (OCCURS-AT  $ST_x$   $ST_x$   $AT_x$  )
     (FIND-RELATED-EVENTS  $ST_x$   $AT_x$  ))

If  $AT_x$  is greater than  $ST_x$ 
Then (EVENT-CLASS-IS-A 'future-event)
     (ETS 'required)
     (OCCURS-AT  $ST_x$   $AT_x$   $ST_x$  )
     (FIND-RELATED-EVENTS  $ST_x$   $AT_x$  ))

```

Figure 5. Temporal Rules in ETS Module

Using these three rules, the ETS module performs continuous modifications in event times obtained from the event lists. These changes are stored in a new list called ETS-LIST which replaces the future event list. The ETS-LIST is accessed by the simulation program during execution thus providing an output depicting the actual occurrence of various events in the cell.

3.4 Dynamic Knowledge Bases in RTKBS

One of the main concerns during real-time control using knowledge-based simulation is responding to

an operational problem in a short time period. One of the design features included in the RTKBS system to ensure response time reduction is a dynamic knowledge base. This is included to store the results from each simulation experiment. During each simulation run, the outputs created are stored in the dynamic knowledge base. Therefore, if the RTKBS system has previously observed the control problem currently being investigated, then the results can be obtained from the dynamic knowledge base instead of through simulation. If not, the new control problem is considered for resimulation. One of the major difficulties of designing this dynamic knowledge base is its representation *a priori*. However, this feature provides the ability to construct a knowledge base based on real data and actual control problems as and when they occur.

3.5 Operating Principles of RTKBS

The operating principles of the RTKBS system are indicated in Figure 6. The data from the manufacturing cell is collected continuously and stored in a dynamic database. Data collection for RTKBS is primarily restricted to temporal data including part arrival times, part departure times, part processing time, part movement times, machine and material handling breakdown and repair times, and part storage and retrieval times. A knowledge-based controller has been designed to interact with the data collection, simulation, dynamic knowledge base, ETS, and supervisor modules in the RTKBS system.

Under normal circumstances, the temporal data collected from the manufacturing cell is utilized by the ETS module to synchronize the simulated events and their times of occurrence. Thus, the outputs generated by the simulation model will more accurately correspond to the activities of the manufacturing cell. These outputs are monitored by the supervisor. The ETS module is interfaced with the simulation and the dynamic database. Each of the next events during simulation is compared with the data collected from the manufacturing cell and modified by the temporal rules. If the event times in the real and simulated systems are the same, the ETS module does nothing. If the next event obtained from the future event list has already occurred in the real system, then it is considered as a past event. In such cases, the ETS module resets the time of occurrence of the next event and all those events that are related to it accordingly using the past event list.

Similarly, if the next event has not yet taken place in the real system, then it is reinserted into the future event list. The ETS module resets the time of occurrence of the next event and all

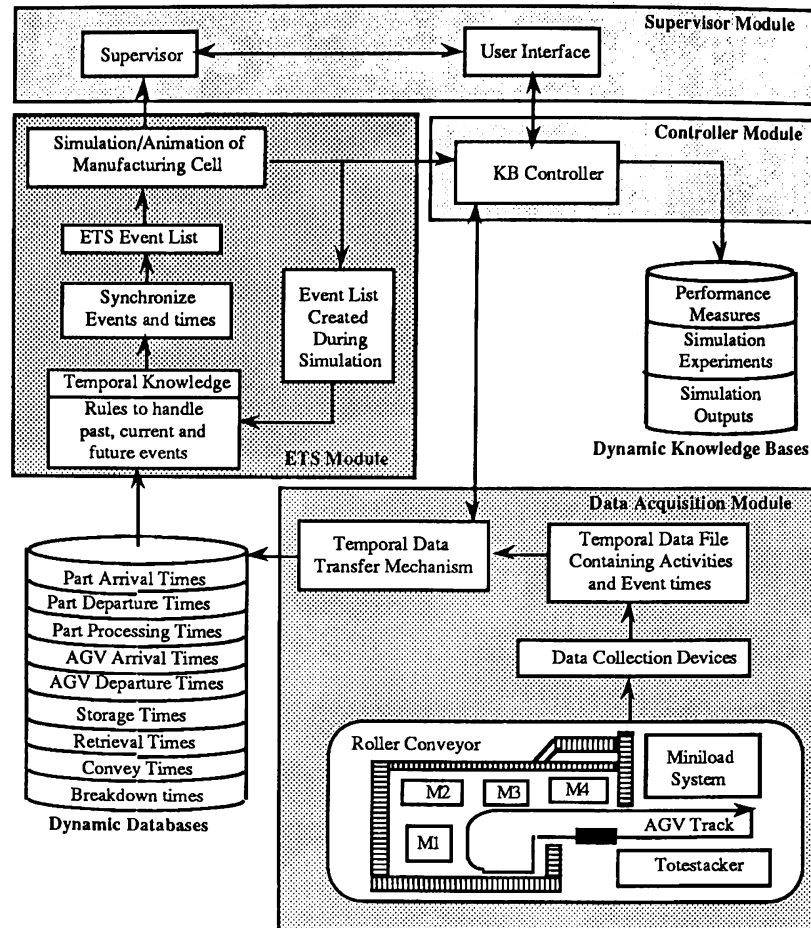


Figure 6. Operating Principles of the RTKBS System

those events that are related to it accordingly using the future event list. The discrepancy in the event times between simulated and real time is due to (i) modelling errors, (ii) errors in input data, and (iii) unforeseen situations in the real system that are not considered in the simulation model.

Suppose that the supervisor wants to examine an operational problem. Some of these problems include (i) injection of new parts into the system, (ii) changes in the arrival and processing patterns of parts, (iii) changes in the priority levels of parts currently in process, and (iv) breakdown of a machine or a material handling system. In any of these four cases, the supervisor changes the operating parameters of the simulation model that has already been synchronized with the operations of the manufacturing cell. The changes in the operating parameters are obtained from the supervisor via user interface. Once the data is obtained, the dynamic knowledge base is searched for a match in terms of the control problem and the conditions of the manufacturing cell. If there exists a match in the knowledge base, then results are displayed for the supervisor. This feature enhances the capabilities of RTKBS, reduces the

response time, and eliminates multiple resimulations in real time. If there does not exist a match, then resimulation will be conducted for a specified number of alternatives, and the best one is chosen. The results will be displayed to the supervisor and stored in the dynamic knowledge base for future use.

4 APPLICATION OF RTKBS

The RTKBS framework has been implemented to perform real time control in a manufacturing cell integrated with a variety of material handling systems. The cell is currently in operation at the Georgia Institute of Technology. The simulation model is restricted to material flow problems in the cell. Therefore, only the material handling and storage subsystems are simulated. The cell includes a miniloader subsystem, a totestacker subsystem, an AGV network, and a conveyor subsystem.

The simulation model has been developed as a first step to designing a real-time knowledge-based system for controlling a facility. The results generated during simulation will be constantly

updated using on-line data received from various processors, material handling systems and other auxiliary subsystems in the manufacturing cell. This design feature provides an ability to control the cell from a remote location and to obtain the most recent data for conducting simulation and analysis. The supervisor can monitor the facility from this remote location and evaluate several control policies related to material flow as well as the overall performance of the manufacturing cell. Each time, the supervisor conducts a series of simulation experiments, the best choice (either selected by the supervisor on-line or using a rulebase) is stored in the dynamic knowledge base. This eliminates the need for resimulation thereby reducing the response time.

4.1 Description of a Manufacturing Cell

The current layout of the manufacturing cell is shown in Figure 7. The miniload subsystem is a one-aisle system that can store and retrieve up to three totes at a time. Its storage capacity is 231 totes. The miniload system usually operates on a dual-command-strategy, i.e., one storage and one retrieval in each operation cycle. The totestacker subsystem also has only one aisle and stores one tote per storage location. Its maximum storage capacity is 248 totes.

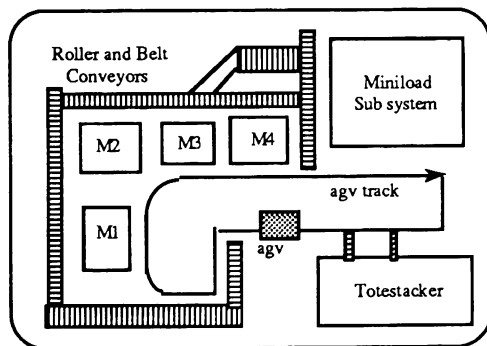


Figure 7. Manufacturing Cell Layout

A Litton AGV is used to transport totes between the work stations and the conveyor segments. The AGV travels along a fixed path and checks each workstation to determine whether there are parts to be loaded or unloaded. The information exchange between AGV and work stations is performed via infrared light. The conveyor subsystem consists of several segments of roller and belt conveyors. Transfer between segments is performed by pop-up chains or rollers. Infrared sensors detect arriving totes and initiate routing of parts to different locations. A pair of pop-up chains initially moves totes on the conveyor system and releases totes one-by-one to the conveyor

system to avoid cramming. After a distance of about 35 ft the conveyor divides into two segments. Incoming totes are routed cyclically either to the left or to the right segment, i.e., the first tote goes to the left, the second to the right, the third to the left again etc. The conveyor system terminates at the front end of the miniload system. There are also short conveyor segments in front of the totestacker which deliver totes from and to the AGV.

Pallets containing one, two or three totes are retrieved by the miniload S/R machine and placed at the front-end of the miniload system. The totes are manually loaded on a conveyor and moved to a pickup point for the AGV. The AGV can handle a maximum of three totes from the conveyor and transports them to a totestacker. Some or all totes are transferred to a short belt input conveyor at the totestacker where they wait for an S/R machine to pick up and store temporarily. The S/R machine also retrieves totes from totestacker area and places them on a short belt output conveyor. These totes wait for the AGV and are then transported to a short segment roller conveyor as shown in Figure 7. The totes are moved by a pop-up chain conveyor to the next conveyor segment which transports them to the miniload system. There is a branch segment on the conveyor which is used to alternate one tote along the straight segment and the other along the branch segment. Finally, the totes are manually placed on a pallet at the I/O point of the miniload where they are picked up by the S/R machine and stored.

4.2 Implementation Details of RTKBS

Currently, the implementation of the RTKBS framework is restricted to perform real-time control of the manufacturing cell discussed in Section 4.1. SIMAN/Cinema software on a Sun workstation has been utilized to build the simulation model of the manufacturing cell. For each of the subsystems in the cell, a model file and an experiment file were developed. The dynamic knowledge base was implemented in Common Lisp and interfaced with the SIMAN output files. Data from the manufacturing cell was collected on arrival times and departure times of the AGV at different work stations and stored in a temporal data file. The knowledge-based controller implemented using Common Lisp was used to check for discrepancy in events and their times of occurrence during simulation from the temporal file. In case of a discrepancy, the controller would perform the event/time synchronization and create an ETS-LIST. This list has been utilized by the simulation program in scheduling the next event. The dynamic knowledge base was represented using frames in

Common Lisp and was accessed by the controller. A menu-driven and user-friendly supervisory interface was implemented using Common Lisp.

4.3 Outputs From RTKBS

In RTKBS, the entities and resources are represented as icons and frames, for the purpose of examining alternative control policies in real time. The icons are represented in the SIMAN/Cinema environment and frames in the Common Lisp environment. The representation methods simplify the problem of integrating entities and resources in the simulation and knowledge base environments. Figure 8 indicates the frames implemented in the RTKBS system. By changing the values of parameters (A1, A2, A3,...) associated with the entities or resources, the performance of the manufacturing cell can be evaluated via simulation. For instance, the tote arrival rates at miniload can be increased or decreased within a certain limit. The maximum and minimum values for these parameters are controlled by the knowledge base. Based on the simulation outputs, the supervisor can take the necessary control measure. The simulation outputs are then stored in a dynamic knowledge base for future use.

```
(single-tote A1 A2 A3 A4 ....)
(double-tote A1 A2 A3 A4 ....)
(triple-tote A1 A2 A3 A4 ....)
(agv B1 B2 B3 B4 ..... )
(miniloader M1 M2 M3 M4 ..... )
(totestacker L1 L2 L3 L4 ..... )
(conveyor C1 C2 C3 C4 ..... )
(miniload O1 O2 O3 O4....)
(toteload T1 T2 T3 T4 .....)
```

Figure 8. Frame Representation in RTKBS

The outputs generated by the simulation module of the RTKBS system has been currently restricted to (i) travel time of a tote between the miniload and the totestacker (ttmt), (ii) travel time of a tote between the totestacker and the miniload (tttm), (iii) time between requesting and arrival of the AGV at miniload (tbrams), (iv) time between requesting and arrival of the AGV at totestacker (tbrat), (v) time between requesting and arrival of the miniload S/R machine (tbramsr), (vi) time between requesting and arrival of the totestacker S/R machine (tbratsr), (vii) miniload S/R utilization (miniutil), and (viii) totestacker S/R utilization (toteutil). These results are stored in a

dynamic knowledge base represented using frames as shown in Figure 9.

```
(DYNAMIC-KB
(number-of-parameter-changes-possible.....)
(current change.....)
(Start-sim.....)
(End-sim.....)
(Input-data
(parameter-1 parameter-2 ..... )
(Alt-list ((.....) (.....) (.....))
(Best-Option
(Input-data-modification
(parameter-1 parameter-2 ..... )
(Output-data-For-Selected-Alternative
(tttmt ttttm tbrams tbrat tbramsr
tbratsr miniutil toteutil)
```

Figure 9. Dynamic KB Structure

The *Start-sim* and *End-sim* provide the starting and finishing times of simulation, *Input-data* stores the original parameter values, *Alt-list* gives the list of alternatives generated using the knowledge base along with the changes in the parameter values and the outputs generated during simulation, and *Best-Option* provides the *input-data-modification*, and *Output-data-For-Selected-Alternative* for the control problem.

The supervisor can modify the following variables for real-time control purposes: (i) unit load size and its arrival rate, (ii) speed and capacity for the AGV, conveyor, totestacker and miniload, (iii) processing times at the totestacker and miniload, (iv) travel times between workstations, (v) storage capacity of totestacker and miniload subsystems, and (vi) buffer space in front of the workstations. These are specified as parameters in the SIMAN environment. For instance, parameter-1 is used to represent the batch size of the arriving totes. The parameter values are modified during simulation. Sample results from the dynamic knowledge base in RTKBS are shown in Figure 10.

Results from the RTKBS system indicate that there are eleven parameters which can be modified by the supervisor in real time. The event times are synchronized using the ETS module using the data collected from the manufacturing cell. For illustrating the operating principles of the RTKBS system, a control problem has been studied at 2074 minutes. Seven alternatives are generated using the knowledge base and simulation has been conducted from 2074 to 3200 minutes for each alternative. *Input-data* stores the original eleven parameter values. For each alternative, the changes in the parameter values and the outputs generated during simulation are stored in *Alt-list*.


```

(DYNAMIC-KB
(number-of-parameter-changes-possible '11)
(current change DDDCDDCDDC)
(Start-sim 2074 minutes)
(End-sim 3200 minutes)
(Input-data
(3 10 20 22 39 1 2 500 100 120 60))
(Alt-list
(DDDDDDDDDDDD)
(3 10 20 22 39 1 2 500 100 120 60)
(55 171 34 10 3.16 0.82 0 7.5 0.34 0.95))
((CDDCDDDCDDD)
(1 10 20 20 30 1 2 300 100 120 60)
(55 171 36 12 3.05 0.82 0 7.43 0.37 0.95))
((DCDCDDCDDCD)
(3 12 20 18 39 1 1 500 100 100 60)
(55 171 35 10 3.24 0.77 0 7.47 0.33 0.95))
((DDCDDDDDDDD)
(3 10 18 22 39 1 2 500 100 120 60)
(55 171 33 10 3.24 0.81 0 7.51 0.35 0.95))
((CCDDCCDDDDCC)
(2 12 20 22 30 2 2 500 100 100 50)
(55 172 34 11 3.16 0.71 0 7.45 0.35 0.95))
((DCDCDDCCCCD)
(3 12 20 20 39 1 1 200 110 100 60)
(55 171 33 10 3.16 0.82 0 7.51 0.35 0.95))
((DDDCDDCDDC)
(2 12 20 18 30 2 1 500 110 100 60)
(55 171 35 10 3.24 0.71 0 7.41 0.31 0.92)))
(Best-Option
(Input-data-modification
(2 12 20 18 30 2 1 500 110 100 60)
(Output-data-For-Selected-Alternative
(55 171 35 10 3.24 0.71 0 7.41 0.31 0.92))

```

Figure 10. Sample Results From RTKBS

From Figure 10, it can be inferred that only seven alternatives have been identified by the knowledge base for simulation. Without the knowledge base, we may have to examine a large number of alternatives leading to longer disruption times. Now, let us consider the second alternative in *Alt-list* in order to understand its contents.

```

((CDDCDDDCDDD)
(1 10 20 20 30 1 2 300 100 120 60)
(55 171 36 12 3.05 0.82 0 7.43 0.37 0.95))

```

(CDDCDDDCDDD) means that the first, fourth and eighth parameter values have been modified from the original value by the supervisor (or the knowledge base). (1 10 20 20) shows the new parameter values and (55 171 36 12) provides the outputs on various performance measures according to the structure shown in Figure 9. Finally, the *Best-Option* is selected by the supervisor and is stored in the dynamic knowledge base for future use. This list provides

the necessary changes in the operating parameters to be effected in the cell.

5 SUMMARY

A new framework (RTKBS) for real-time control of a manufacturing cell using knowledge based simulation has been developed. Using abstract terminology, the meaning of modeling, control and analysis within the RTKBS framework is provided. The need for event synchronization for real-time control and an approach to handle this problem are described. A dynamic knowledge base to store the results of previous on-line simulations is implemented to reduce the response time. Finally, a sample problem is described to illustrate the operating principles of the RTKBS system along with the implementation details.

Currently, the RTKBS system and the principles of event-time synchronization are being extended to diagnose quality degradation within several CNC grinding machines. Sensory devices will be connected to the critical components of the grinding machine to collect active and passive signatures on vibration, positional error and thermal effects. The dynamic knowledge base will be extended to analyze the sensory data and to determine the nature of quality degradations (along with the parametric values) for each of the critical components. Using the on-line information stored in the knowledge base, the factual and procedural rules, and a simulation model depicting the quality degradations within a grinding machine, the supervisor can diagnose and predict quality degradations, and evaluate alternative preventive maintenance policies. This research involves a consortium including the Material Handling Research Center at the Georgia Institute of Technology, Digital Equipment Corporation, The Torrington Company, and the National Science Foundation.

REFERENCES

- Ben-Arieh, D., C.L. Moodie, and S. Y. Nof. 1985. A Knowledge based system for automated production and assembly. *Toward the Factory of the Future*, eds. R. Bullinger and G. Warnecke, 285-293. Springer-Verlag, Berlin, W. Germany.
- Catron, B. A., and B. H. Thomas. 1988. Generic manufacturing controllers. In *Proceedings of the IEEE Conference on Intelligent Control*, 742-744. IEEE, Piscataway, NJ.
- Furlani, C. M., D. Libes, E.J. Barkmeyer, and M. J. Mitchell. 1988. The Distributed Data System of the AMRF of the National Institute of Standards and Technology. In *Proceedings of*

- the 33rd IEEE Computer Society International Conference*, 33-42. IEEE, Piscataway, NJ.
- Kachitvichyanukul, V., S.J. Cheng, and D.R. Denzler. 1986. KBSIM: A Knowledge Based Simulator for FMS Scheduling and Dispatching. In *Proceedings of the Summer Computer Simulation Conference*, 696-698. Society for Computer Simulation, San Diego, CA.
- Krijgsman, A.J., H.B. Verbruggen, and P.M. Bruijn. 1988. Knowledge-Based Real-Time Control. In *IFAC Proceedings of the Artificial Intelligence in Real-Time Control*, eds. M. G. Rodd and G. J. Suski, 13-19. Pergamon Press, New York, NY.
- Manivannan, S., and J. Banks. 1990. Towards a Real-time Knowledge-Based Simulation System For Diagnosing Machine Tool Failure. In *Proceedings of the Winter Simulation Conference*, eds. O. Balci, R. P. Sadowski and R. E. Nance, 603-608. IEEE, Piscataway, NJ.
- Manivannan, S., and J. Banks. 1992. Design of a knowledge based on-line simulation system to control a manufacturing shop floor. *IIE Transactions*: to appear.
- McLean, C. R. 1986. Information architecture of automated manufacturing research facility. In *Proceedings of the Information Technologies Conference*, 76-89. Rensselaer Polytechnic Institute, Troy, NY.
- Simmonds, W.H. 1988. Representation of real-knowledge for real-time use. In *IFAC Proceedings of the Artificial Intelligence in Real-Time Control*, eds. M. G. Rodd and G. J. Suski, 7-11. Pergamon Press, New York, NY.
- Voss, H. 1988. Architectural issues for expert systems in real-time control. In *IFAC Proceedings of the Artificial Intelligence in Real-Time Control*, eds. M. G. Rodd and G. J. Suski, 1-6. Pergamon Press, New York, NY.
- Young, R.E. and M.A. Rossi. 1988. Towards a knowledge based control of a flexible manufacturing system. *IIE Transactions*, 20: 36-42.

algorithmic approach for analyzing generalized conveyor networks.

JERRY BANKS is an associate professor in the School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research interests are in the simulation of manufacturing systems and in the verification and validation of large scale simulation models. He is a founding partner of the simulation consulting firm Carson/Banks & Associates. He is the Institute of Industrial Engineers' representative to the Board of the Winter Simulation Conference and is completing his term as Board Chair of that body.

AUTHOR BIOGRAPHIES

S. MANIVANNAN is an assistant professor in the School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research interests are in the areas of knowledge-based simulation, design and analysis of manufacturing cells, robotics, and automation. He is involved in developing (i) a knowledge-based simulation system for diagnosing quality degradation in CNC machine tools, (ii) a collision avoidance system for a robot in a flexible assembly cell using a dynamic knowledge base, and (iii) an