Proceedings of the 1999 IEEE/ASME
International Conference on Advanced Intelligent Mechatronics
September 19-23, 1999 • Atlanta, USA

Integrated Machine and Control Design for Reconfigurable Machine Tools

D. M. Tilbury and S. Kota
Engineering Research Center for Reconfigurable Machining Systems
Department of Mechanical Engineering and Applied Mechanics
The University of Michigan
Ann Arbor, MI 48109-2125

{tilbury,kota}@umich.edu

Abstract— **In this paper, we describe a systematic design procedure for reconfigurable machine tools and their associated control systems. The starting point for the design is a set of operations that must be performed on a given part or part family. These operations are decomposed into a set of functions that the machine must perform, and the functions are mapped to machine modules, each of which has an associated machine control module. Once the machine is constructed from a set of modules, the machine control modules are connected. An operation sequence control module, user interface control module, and mode-switching logic complete the control design. The integration of the machine and control design and the reconfigurability of the resulting machine tool are described in detail.**

I. Introduction

In today's competitive markets, manufacturing systems must quickly respond to changing customer demands and diminishing product life cycles. Traditional transfer lines are designed for high volume production, operate in a fixed automation paradigm, and therefore cannot readily accommodate changes in the product design. On the other hand, conventional CNC-based "flexible" manufacturing system offer generalized flexibility but are generally slow and expensive since they are not optimized for any particular product or a family of products.

An effort at the University of Michigan aims to develop the theory and enabling technology for reconfigurable machining systems [3], [4]. Instead of building a machining system from scratch each time a new part is needed, an existing system can be reconfigured to produce the new part. In this paper, we describe how an integrated machine and control design strategy can result in machine tools which can be quickly and easily configured and reconfigured.

In order to provide exactly the functionality and capacity needed to process a family of parts, RMTs are designed around a given family of parts. Given a set of operations to be performed, RMTs can be configured by assembling appropriate machine modules. Each active module in the library has a control module associated with it. As the mechanical modules are assembled, the control modules will be connected and the machine will be ready to operate. Extensive and time-consuming specialized control system design will not be required. Section II describes how the machine is designed from a set of basic machine modules, connected in a well-defined fashion, and Section III describes how the control is similarly assembled from a library of control modules. This modular construction of the machine and control allows for many levels of reconfigurability as described in Section IV. The paper concludes with a description of future work in Section V.

II. Machine Design

Ongoing work on manufacturing system configuration at the University of Michigan addresses the problem of starting from a part (or part family) description and extracting the machining operations necessary to produce the part(s) [7]. The operations are grouped according to tolerance, order of execution, and desired cycle time of the system, with the intention that each operation "cluster" can be produced on a single machine tool. The operation cluster considered here is to drill a set of holes for the cam tower caps on V6 and V8 cylinder heads shown in Figure 1. The input to the reconfigurable machine tool design procedure is the cutter location data generated by a process planner for this operation cluster. Figure 2 shows sample data which includes positioning and drilling information.

The RMT design procedure consists of three main stages: task clarification, module selection, and evaluation [14]. After a brief literature review, these three stages will be outlined in this section.

A. Related research

Since reconfigurability is a relatively new concept in machining systems, there is little, if any, published literature on the design of reconfigurable machine tools. However, modular machine tools have been on the market for several years, and some of the published articles on modular robots, modular machines and assembly are relevant to the design of reconfigurable machine tools. For example, Shinno and Ito [17], [18], [19], [20] proposed a methodology for generating the structural configuration of machine tools. They decomposed the machine tool structures into simple geometric forms: e.g. boxes, cylinders, etc. Yan and Chen [21], [1] extended this work to the machining center structural design. Ouyang et al. [12] adapted Ito's method for modular machine tool synthesis and developed a method for enumerating machine tool modules. Paradis and Khosla [15] determined the modular assembly configuration which is optimally suited to perform a specific task. Chen [2] addressed the problem of finding an optimal assembly configuration for specified tasks; his procedure was

This research was supported in part by the NSF-ERC under grant number EEC95-92125.

0-7803-5038-3/99/$10.00 © 1999 IEEE

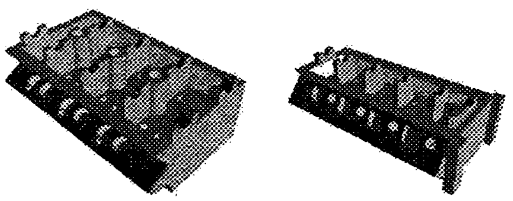

(a) V6 cylinder head (b) V8 cylinder head

Fig. 1. Two sample parts. The operation to be performed is to drill the positioning holes for the cam tower caps. On the V8 engine, there are two such positioning holes located in a line. On the V6 engine, there are eight holes in an array.

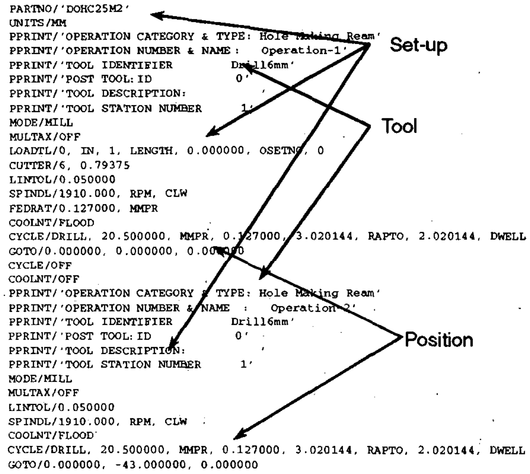

```
PARTNO/'DOHC25M2'
UNITS/MM
PPRINT/'OPERATION CATEGORY & TYPE: Hole Making Ream'
PPRINT/'OPERATION NUMBER & NAME :   Operation-1'
PPRINT/'TOOL IDENTIFIER         Drill6mm'
PPRINT/'POST TOOL: ID            0'
PPRINT/'TOOL DESCRIPTION:         '
PPRINT/'TOOL STATION NUMBER      1'
MODE/MILL
MULTAX/OFF
LOADTL/0, IN, 1, LENGTH, 0.000000, OSETNO, 0
CUTTER/6, 0.79375
LINTOL/0.050000
SPINDL/1910.000, RPM, CLW
FEDRAT/0.127000, MMPR
COOLNT/FLOOD
CYCLE/DRILL, 20.500000, MMPR, 0.127000, 3.020144, RAPTO, 2.020144, DWELL
GOTO/0.000000, 0.000000, 0.000000
CYCLE/OFF
COOLNT/OFF
PPRINT/'OPERATION CATEGORY & TYPE: Hole Making Ream'
PPRINT/'OPERATION NUMBER & NAME  :   Operation-2'
PPRINT/'TOOL IDENTIFIER         Drill6mm'
PPRINT/'POST TOOL: ID            0'
PPRINT/'TOOL DESCRIPTION:         '
PPRINT/'TOOL STATION NUMBER      1'
MODE/MILL
MULTAX/OFF
LINTOL/0.050000
SPINDL/1910.000, RPM, CLW
COOLNT/FLOOD
CYCLE/DRILL, 20.500000, MMPR, 0.127000, 3.020144, RAPTO, 2.020144, DWELL
GOTO/0.000000, -43.000000, 0.000000
```

Fig. 2. Sample sequence of operations (cutter location) data for drilling holes in the parts shown in Figure 1. The CL file is generated from a CAD package (such as IDEAS) and includes the locations of the holes to be drilled along with the spindle speed, feedrate, and coolant information.

based on the assembly incidence matrix and employed a genetic algorithm to solve the optimization problem. On the systems front, Rogers and Bottaci [16] discussed the significance of reconfigurable manufacturing systems, and Owen et al. [13] developed a modular reconfigurable manufacturing system synthesis program for educational purposes.

In our work, traditional methods of motion representation and topology (i.e. screw theory, graph theory, etc.) are employed to capture the characteristics of RMTs. These mathematical schemes are used for topological synthesis, function-decomposition, and mapping procedures; details can be found in [9].

B. Task clarification

The design of an RMT begins with task clarification, which entails analyzing the cutter location data to determine the set of functions which are necessary to accomplish the desired kinematic motions. There are three steps. First, graphs are generated which abstractly representation the motions. These graphs are then decomposed into functions, and finally the functions are mapped onto machine modules which exist in the library.

A graph representation of the machine tool structure allows for systematic enumeration of alternate configurations and also provides a method of identification of non-isomorphic graphs. The graph representation is also used for bookkeeping to assign machine modules to the graph elements. A graph consists of a set of vertices connected together by edges. In using a graph as an abstract representation of a machine tool structure, we define two different types of vertices: type 0 and type 1. A vertex represents a physical object with two ports; each port represents the location on the object where it can be attached to a neighboring object. A type 0 vertex has input and output ports that are in-line with respect to each other, whereas a type 1 vertex has input and output ports that are perpendicular to each other. Machining tasks are also classified as type 0 or type 1, depending on whether the tool is parallel or perpendicular to the workpiece.

	Position	Feed	Spindle	Coolant
t1	▨			
t2			▨	▨
t3		▨	▨	▨
t4			▨	▨
t5	▨			
t6			▨	▨
t7		▨	▨	▨
t8			▨	▨

Fig. 3. High-level operation sequence, showing causal dependencies and concurrencies. This abstract representation of the sequence of operations is derived from the CL data shown in Figure 2, and will be used to design the sequencing control.

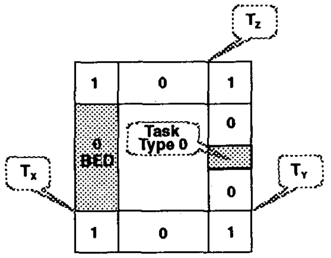


Fig. 4. An graph representing a machine tool structure. Translational motions (T_X, T_Y, T_Z) are assigned to edges of the graph; the vertices have structural functionality.

Figure 4 shows a graph for a type 0 task. Four type 1 vertices are combined with several type 0 vertices to create a machine structure in the form of a C. Because type 0 vertices don't change the orientation, they can be used as spacers in various combinations. The root vertex represents the base or bed of the machine tool. The choice of the root vertex is not unique; different choices will result in distinct machine tool designs. Structural functions are assigned to the vertices of the graph; kinematic functions (where needed) are assigned to the edges. For instance, Figure 4 shows one example of how translational motions in the X, Y and Z directions can be assigned to graph edges, representing relative motion between the physical objects represented by the two vertices of the edge.

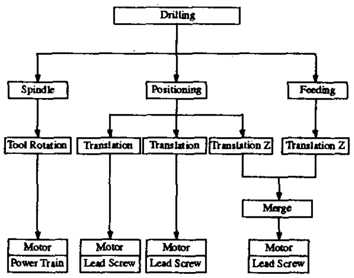


Fig. 5. Function decomposition template.

The basic functionality of a machine tool is described by the kinematic motion between the tool and the workpiece. These kinematic functions will be represented by a homogeneous transformation matrix [11]; the desired functionality of the machine tool will be encoded in the matrix T. The motions necessary to carry out a given machining task are derived from the sequence of operations. The process file shown in Figure 2 contains tool positions and motions in a Cartesian coordinate system. For example, the first motion can be extracted as:

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0.127 \\ 0 & 1 & 0 & 3.02 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad F_1 = \begin{bmatrix} 1 & 0 & 0 & 0.127 \\ 0 & 1 & 0 & 3.02 \\ 0 & 0 & 1 & 2.02 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where P_1 represents the position and orientation of the tool for the positioning task, and F_1 represents the feed motion. From the transformation between any two adjacent positions, the motion description can be extracted:

$$M_1 \quad = \quad P_1{}^{-1}F_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2.02 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The other motion descriptions are extracted similarly.

Corresponding to each type of machining operation, a template is retrieved as a starting point in identifying various kinematic functions necessary to carry out the machining task. For instance, the template for milling and drilling operations show that kinematic functions are necessary for spindle revolution, tool feeding and tool positioning. By using this template, together with the exact feeding and the positioning information given in the process plan, we can derive the exact set of kinematic functions that are necessary such as tool rotation, translations X, Y, and Z for feeding and translation Z for tool positioning as depicted in Figure 5.

Each of the kinematic functions identified in the function decomposition stage is mapped to an edge of the graph as described above. Assigning the functions to different edges can generate multiple solutions. Because purely translational motions are commutative, their order in the graph can be interchanged. In function mapping, the important information is the topology of screw motions (including pure rotational motions) and the topology of the bed.

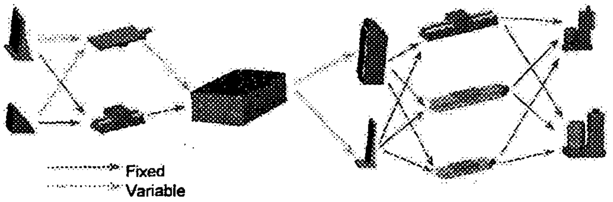
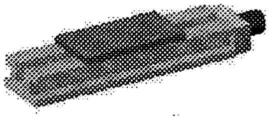


Fig. 6. The structural graph of Figure 4 can be realized by many different choices of modules.



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \theta_1 \\ 0 & 0 & 1 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 7. Representation of a machine module. The CAD model of a slide for a modular machine tool is shown on the left, and its transformation matrix is shown on the right.

C. Module selection

Commercially available modules are selected from the module library for each of the functions (structural as well as kinematic) that were mapped to the graph in the task clarification stage. The data stored for each module in the library includes the homogeneous transformation matrix representing its kinematic or structural function, the twist vector supplemented by range of motion information, a compliance matrix representing the module stiffness, module connectivity information, and power requirements (for active modules such as spindles and slides).

The first step in module selection is to compare the homogeneous transformation matrices of the modules with the task requirement matrix such that when appropriate modules are selected to meet the task requirements, the product of all module matrices should be equal to the desired task matrix: $T = T_1T_2\cdots T_n$. Again, there may be many possible choices of modules for a given structural configuration. Figure 6 shows how different slides, spindles, and structural elements can be assembled according to the graph of Figure 4.

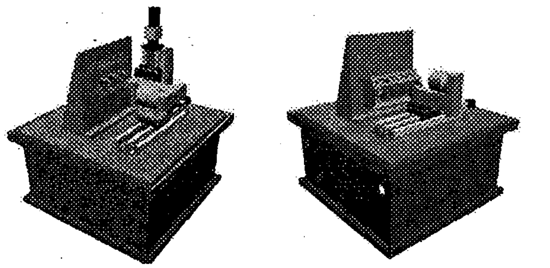
A slide module, with its CAD model and transformation matrix, is shown in Figure 7. It is capable of one direction of linear motion, indicated by the θ_1 variable in its transformation matrix. Its database entry, shown in Table I, stores not only its transformation matrix but also the manufacturer name, model number, initial position, power level, and motion data. The twist vector is augmented by information on the minimum, initial, and maximum displacement of the module.

D. Evaluation

Once a set of kinematically-feasible modules have been selected, the resulting machine design must be evaluated. The criteria for evaluation of the reconfigurable machine tools synthesized by the above systematic procedure include the work envelope, the number of degrees of freedom, the number of modules used, and the dynamic stiffness.

TABLE I
DATABASE INFORMATION AND DOCUMENTATION FOR THE MACHINE MODULE SHOWN IN FIGURE 7.

Manufacturer	SUHNER
Model Name	UA 35-AC
Initial Position	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&100\\0&0&0&1\end{bmatrix}$
Twist vector	$\begin{bmatrix}0&0&0&0&1&0\end{bmatrix}^T$
Range of motion	$\begin{bmatrix}-155&0&155\end{bmatrix}$
Max. force	5500N
Etc.	Compliance matrix, Power level, Connectivity information, ...



(a) V6 machine

(b) V8 machine

Fig. 8. Reconfigurable machine tool designs for the two different parts.

The number of kinematic degrees of freedom of the machine tool must be kept to a minimum required to meet the requirements, both to reduce the actuation power and minimize the chain of errors. Examples show that the designs generated by this methodology have exactly the number of degrees of freedom necessary to perform the required machining operations on the given part [10]. Machine tool designs which are generated using this methodology for the example parts of Figure 1 are shown in Figure 8.

The resulting designs must be evaluated with respect to the expected accuracy. The stiffness of the entire machine tool, one of the most important factors in performance, is estimated based on the module compliance matrices and the connection method.

III. Control Design

As the machine is built from modular elements, so is the control. In this work, we focus on the logic control for sequencing and coordination of the machine modules; a discrete-event system formalism is used [6]. There is one control module associated with each active machine module; we refer to these as *machine control modules.* In the machine design, there are passive elements which connect the active elements together. In the control design, there must also be "glue" modules which connect the machine control modules. The overall architecture of the control system for an RMT is shown in Figure 9. The structure is similar for either of the two machines shown in Figure 8; for the V6 machine, there is also Y-axis control module. As shown, the machine control modules are at the lowest level; these interact directly with the mechanical system. The user interface control module is at the highest level, interacting with the user through pushbuttons and a display. The operation sequence control module is defined based on the high-level operation sequence for the part as shown in Figure 3. Three modules handle the mode switching logic.

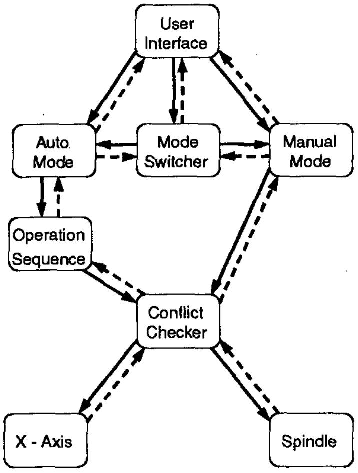


Fig. 9. The overall structure of the modular control system.

A. Machine control modules

Each machine control module has a well-defined interface specification: it accepts discrete-event commands from a given set, and returns discrete-event responses from a given set. Following the discrete-event system framework, events which it receives are called *uncontrollable* and events which it can generate are *controllable.* Within the control module will be all of the continuous-variable control, such as servo control for axes. This continuous control is designed using standard PID algorithms and the axis parameters such as inertia, power, lead screw pitch, which come from the machine module definition. In addition, each machine control module will contain controls for any machine services associated with the machine module, such as lubrication or coolant. Thus, each machine control module is a self-contained controller for the machine module it accompanies, and can be designed and tested independently of the rest of the machine.

The design of a machine control module must be done only once for each machine module in the library. Whenever the machine module is used in a machine design, the control module can be used in the associated control design. The control module may be used independently, with its own processing power, I/O and a network connection to the rest of the control system, or it may be used as a piece of the overall machine controller which is implemented in a centralized fashion.

An example of a machine control module for a slide is shown in Figure 10. Each circle represents a state, and

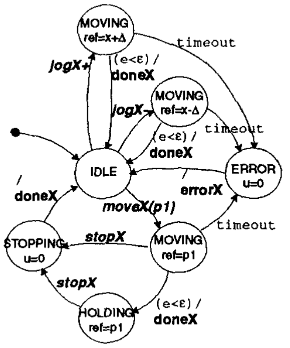


Fig. 10. Slide Controller. When the slide has reached the commanded position to within some tolerance ϵ, a "done" response is returned. If the motion times out, and "error" event is issued.

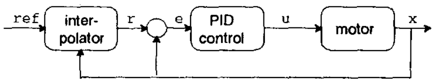


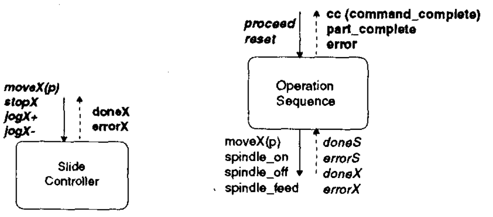
Fig. 11. Servo loop for a slide. The slide controller of Figure 10 sets either the reference **ref** or the motor voltage **u**.

the arrows represent transitions between states. Associated with each transition is an event/action pair. When an uncontrollable (italicized) event is received, the appropriate action is generated.

As shown by its block diagram in Figure 12a, there are four events that the module can receive: move to a position x, stop, jog in $+x$, and jog in $-x$ direction. When the desired operation is finished, the "done" response is returned. A watchdog timer is included; if a prespecified amount of time elapses and a done response has not been issued, an "error" response will be returned. A simple block diagram of the associated servo loop for the slide module is shown in Figure 11. The discrete-event slide control module interfaces with the servo loop by setting either the reference value **ref** or the motor input voltage **u**.

B. Operation sequence

The operation sequence module is defined from the high-level sequence extracted from the cutter location data shown in Figure 3. The main structure of this control module is a sequence of states representing the sequence of operations that must be performed on the part; wait states are included at the completion of each step. Figure 13 shows the operation sequence module for the machine of Figure 8(b) and the operation sequence of Figure 3. Simple error handling which merely passes the error up to the user interface is incorporated in the design but is not shown in the figure for simplicity. If a "reset" command is received, the spindle is turned off and the slide is reset to its home position. The operation sequence for the V6 machine is similar, but has more operations because there are two linear axes that need to be sequenced. As shown in the overall structure of Figure 9, there are two ports to the operation sequence control module: one connects to the Auto Mode control module, and another connects to the conflict checker. The interface to the operation sequence control module is shown in Figure 12b.



a. Slide control block b. Operation sequence block

Fig. 12. The block diagrams of the slide controller and operation sequence control modules, showing the ports and shared events. Events received by the module are uncontrollable and are shown in italics; events shared with the upper-level module are in bold.

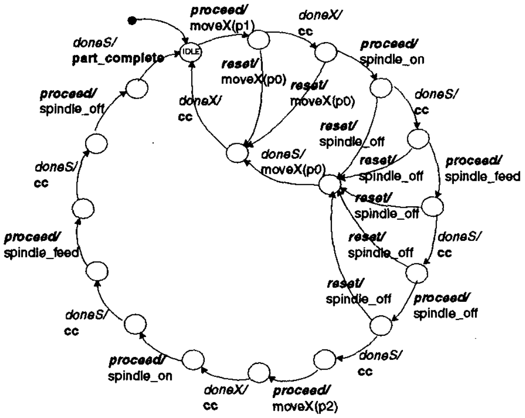


Fig. 13. Operation sequence module, showing the overall sequence of operations and events. The interface to the module is shown in Figure 12b. The reset command can be received at any time; only some of the event traces are shown for simplicity. The error event traces are also omitted from the figure.

C. Modular control structure

The user interface control module interacts with the user through a set of pushbuttons to turn the control system on and off, switch between control modes, and single-step through the operation sequence. It passes user commands through to the rest of the controller and displays the current state of the machine.

Machine tool controllers have several different modes. In the auto mode, the operation sequence executes continually; another mode may execute the operation sequence only once. In step mode, a pushbutton command must be used to initiate every step of the operation sequence, and in manual mode, finer control is available through jog commands that move the active elements a small amount at a time. Instead of repeating the operation sequence for every control mode, one representation of the sequence is used. The mode-switching logic determines the appropriate times to send the "proceed" event to the operation sequence.

The main function of the conflict checker control module is to pass the commands from the operation sequence and manual mode modules to the appropriate machine control module(s). It has access to the database of the machine module definitions, and can check for illegal commands which would result in mechanical interferences. Because of the well-defined interface to the low-level machine control modules, the design of the conflict checker can be done using high-level control commands. The details of the physical I/O are handled in the machine control modules.

As described above, each control module is represented by a finite state machine which accepts a certain language (sequences of events which are allowable). We have shown that with some well-defined conditions on these languages and the module connections, the overall control structure can be guaranteed to be deadlock-free [8]; enumerating all possible sequences of the combined logic controller, which would be impractical, is not required for verification.

IV. Reconfiguration Properties

The machine modules in the library can be used in many different machine designs. The control module associated with each machine module will be incorporated into the control design of the overall machine. This library of machine and control modules can significantly reduce both the lead time of a new machining system, by shortening the design cycle, and the ramp-up time, since each module can be tested independently before they are connected.

For some part changes (such as between the V6 and V8 cylinder heads shown in Figure 1), the machine tool will need to be reconfigured, perhaps by adding an axis or changing the spindle. When this type of reconfiguration occurs, changes need to be made to the operation sequence control module and the conflict checker (if new mechanical interferences are created).

Because they possess a well-defined interface, each individual control module can be changed independently of the others. As long as the redesigned control module has the same discrete-event interface, the resulting system is guaranteed to be deadlock free. For example, a friction compensation control algorithm may be added to one of the slide control modules. This would increase the performance of that module, but the only changes necessary would be within the lowest-level module.

V. Conclusions and Future Work

Historically, machine tool design has been experience-based. In this research, we described a mathematical basis for synthesis and evaluation of Reconfigurable Machine Tools and their associated controllers. This research work has addressed both the generation of machine tool configurations and modular control design. The systematic design process begins with the machining requirements.

The presented methodology for synthesis of machine tools allows a library of machine modules to be precompiled and stored in a database, self-contained with controllers and ready to be used in any machine design. The methodology also ensures that all kinematically viable and distinctly different configurations are systematically enumerated to reduce the chance of missing a good design.

We have already developed a Java-based program which automates the machine design process; we are currently incorporating the control design procedure within the existing framework. We are also expanding the currently-available machine and control module library and formalizing the abstraction from the continuous-variable control to the discrete-event domain.

Acknowledgments

The authors would like to acknowledge the support and invaluable feedback from the industrial members of the ERC who have participated in this project. MEAM graduate students Eric Endsley, Morrison Lucas, and Yong-Mo Moon contributed to the work described in this paper.

References

[1] F.-C. Chen and H.-S. Yan. Configuration synthesis of machining centres with tool change mechanisms. *International Journal of Machine Tools and Manufacture*, 39(2):273–295, February 1999.

[2] I.-M. Chen. *Theory and Application of Modular Reconfigurable Robot Systems*. PhD thesis, California Institute of Technology, 1994.

[3] Y. Koren. Reconfigurable machining systems: Vision with examples. ERC/RMS #19, University of Michigan, January 1999.

[4] Y. Koren and A. G. Ulsoy. Engineering research center for reconfigurable machining systems. http://erc.engin.umich.edu.

[5] S. Kota. A methodology for automated design of reconfigurable machine tools. In *ERC/RMS: Annual Report*, pages 35–40. University of Michigan, 1999.

[6] R. Kumar and V. K. Garg. *Modeling and Control of Logical Discrete Event Systems*. Kluwer Academic Publishers, 1995.

[7] C. Ling, S.-Y. Sung, T. M. L. Olsen, and D. Yip-Hoi. System level process planning for RMS. ERC/RMS #24, University of Michigan, 1999.

[8] M. R. Lucas, E. W. Endsley, and D. M. Tilbury. Coordinated logic control for reconfigurable machine tools. In *Proceedings of the American Control Conference*, pages 2107–2113, 1999.

[9] Y.-M. Moon and S. Kota. Generalized kinematic modeling method for reconfigurable machine tools. In *Proceedings of the ASME Design Engineering Technical Conferences*, Atlanta, September 1998.

[10] Y.-M. Moon and S. Kota. Design of reconfigurable machine tools. *ASME Journal of Manufacturing Systems*, 1999. Submitted.

[11] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[12] M.-A. Ouyang, C. Yi, C. Li, and J. Zhou. Intelligent layout for modular design of machine tools. *SPIE*, 2620:547–552, 1995.

[13] S. Owen, M. C. Bonney, and A Denford. A modular reconfigurable approach to the creation of flexible manufacturing cells for educational purposes. *Institute of Electrical Engineers, Colloquium Digest*, 1(174):1–13, October 1995.

[14] G. Pahl and W. Beitz. *Engineering Design*. Springer-Verlag, New York, 1984.

[15] C. J. J. Paredis and P. K. Khosla. Synthesis methodology for task based reconfiguration of modular manipulator systems. In *Proceedings of the International Symposium on Robotics Research*, pages 2–5, 1993.

[16] G. G. Rogers and L. Bottaci. Modular production systems: a new manufacturing paradigm. *Journal of Intelligent Manufacturing*, 8:147–156, 1997.

[17] H. Shinno and Y. Ito. Structural description of machine tools – 1. Description method and some application. *Bulletin of the JSME*, 24(187):251–258, January 1981.

[18] H. Shinno and Y. Ito. Structural description of machine tools – 2. Evaluation of structural similarity. *Bulletin of the JSME*, 24(187):259–265, January 1981.

[19] H. Shinno and Y. Ito. A proposed generating method for the structural configuration of machine tools. In *ASME Winter Annual Meeting*, 1984. ASME paper 84-WA/Prod-22.

[20] H. Shinno and Y. Ito. Computer aided concept design for structural configuration of machine tools: Variant design using direct graph. *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, 109:372–376, September 1987.

[21] H.-S. Yan and F.-C. Chen. Configuration synthesis of machining centers without tool change arms. *Mechanism and Machine Theory*, 33(1–2):197–212, 1998.