

Virtual Manufacturing for Training and Education

Hazim El-Mounayri and Daniel Aw

Mechanical Engineering Department
Purdue School of Eng. & Tech., IUPUI, Indianapolis, IN

Tamer Wasfy and Ayman Wasfy
Advanced Science and Automation Corp., Smithfield, VA

Abstract

A virtual manufacturing laboratory for educating students and practitioners on advanced manufacturing is proposed. The particular environment for simulating CNC milling is developed. The environment provides the following training functions for students: (a) access to a fully-functional virtual CNC milling machine (FADAL[®] VMC 3016L), (b) training on the key operating procedures of the CNC machine, (c) a lecture describing the components of the milling machine, and (d) a lecture describing the concepts of CNC milling. The environment is driven by three software modules: (1) a CNC Milling machine simulator, (2) a virtual-environment display engine, and (3) an intelligent-agent engine. The three modules run on a single computer in a seamless web-based framework. This will allow the student to access and run the virtual CNC machining center on the web. The CNC Milling machine simulator simulates the machine physical components including the spindle, stages, manual controller interface, and controller software. It also simulates the cutting process including cutting tool motion, workpiece geometry during cutting, cutting forces, machining sounds, and chip-formation. These capabilities are described below. The same paradigm can be used to provide training on other machine tool controllers as well as CNC manufacturing processes.

Keywords

Virtual training environment; training and education; virtual tutor; ABET criteria; engineering program objectives; virtual reality; virtual manufacturing.

1. Introduction

Virtual environments (VEs) provide a safe, cost-effective, and more flexible environment for training. They represent a very promising tool for responding to the increasing need for a workforce that is highly trained on state-of-the-art/sophisticated technology, as well as engineering graduates who are able to use techniques, skills and modern tools of engineering (including manufacturing) effectively and correctly in engineering practice. In some cases, they represent the only alternative. For example, disabled students on wheelchairs cannot reach out to the machine table or controller for training or operational purposes.

In the particular case of manufacturing, which is the backbone of our economy, such a tool becomes even more relevant. On the one hand, the cost of establishing and/or keeping up with the fast changing economy is prohibitive for many of the US colleges. In addition safety is a significant concern. Both of these factors affect the quality of manufacturing education. On the other hand, industry, faced with a fast growing competition, needs to upgrade both its technology and the workforce skills. Virtual environments for training can help the manufacturing industry achieve those objectives more safely, effectively, and economically.

Recent applications of VE based training include: training for operation of: engineering facilities [1], CNC manufacturing machines [2], vehicle driving, piloting, traffic and flight control [3], maintenance simulators [4], medical procedures training [5-7], and military operations training. In [1], an intelligent VE, consisting of an intelligent-agent, an object-oriented virtual environment, and system simulator, was developed for training users in the operation of complex engineering systems. The LEA (Learning Environments Agent) intelligent-agent engine [8, 9] was used in [1] for tutoring, guiding and/or supervising the training. The LEA engine includes a hierarchical process knowledge base engine, an unstructured knowledge base engine for lecture delivery, a rule-based expert system for natural-language understanding, and an interface for driving human-like virtual characters. Three types of objects were used for representing the process knowledge, namely, processes, steps, and constraints. The hierarchical process knowledge base is similar to the Petri-trees used in [2]. The IVRESS (Integrated Virtual Reality Environment for Synthesis and Simulation) engine [10, 11] was used to drive the

virtual environment, display the engineering facility and manage a multimodal input from a variety of sources, including the natural language intelligent agent, hand-held computers, head-tracking, tracked wand, mouse, and keyboard. A system control simulator was used to simulate the system control logic and the physical system response, including the effects of the user's actions.

In the present paper, the LEA intelligent-agent engine and the IVRESS virtual environment engine are integrated with a machine simulator to produce a virtual training environment (VTE) for CNC milling. The machine simulator simulates the motion of the components of the CNC milling machine, the machine controller including the manual controls and control logic, the controller software, including software screens and logic, the interpretation and execution of the G-code program, and the machining process.

A general geometric modeling approach is used to model the milling process [12, 13]. It is based on modeling precisely the geometries involved in the machining operation, including work-piece geometry and tool geometry. Then, using the tool motion and solid modeling based computational algorithms, the new work-piece geometry, the material volume removed, and the in-cut information (radial and axial depth of cut) are calculated.

Mechanics modeling of the machining process involves modeling many physical phenomena that occur during machine. For the purpose of the CNC milling VTE, we only model the major physical phenomena, including cutting forces and machining sounds. Dry machining is assumed and thus coolant flow is not modeled.

The present VTE allows trainees to view and interact with a physically accurate simulation of a modern virtual CNC milling machine. The VTE can run in a web-browser as well as in immersive multi-screen stereoscopic virtual reality facilities such as the CAVETM. An intelligent-agent can train students to operate the CNC milling machine by tutoring, guiding or supervising them through a number of standard machine operating procedures. The VTE provides a safe environment where trainees can learn at their own pace and experiment with various techniques of operating the CNC machine. Also various manufacturing process plans can be tested on the virtual machine before cutting on the physical machine. The agent can also deliver introductory lectures on CNC machining, NC programming, the machine components, and the theory of operation of CNC milling machines.

2. Educational Rationale

2.1. Aim of Current Work

The current work proposes an Advanced Virtual Manufacturing Lab (AVML) for accurate simulation of a physical state-of-the-art manufacturing lab. Such a sophisticated technology is aimed at enhancing the quality, accessibility, and productivity of manufacturing education and training, increase student creativity and problem-solving capability and promote participation and equal access of underrepresented groups to manufacturing technology training. The virtual lab will enable remote access to a sophisticated, state-of-the-art manufacturing laboratory that may not be locally available and will promote long-distance collaboration among students, teachers, and experts. The eventual virtual system is anticipated to be an extremely valuable tool that can enhance student learning in advanced manufacturing.

2.2. Pedagogical Value

The developed technology will allow engineering schools to train students to operate the manufacturing machines in a safe environment. In addition, the proposed system will increase productivity and effectiveness of education and training on advanced manufacturing machinery. For example: 1) Manufacturing processes that require hours or days to complete so that the student can view the results can be simulated in a matter of minutes; 2) The pace and sophistication of the training can be easily adjusted to suite individual expertise level; 3) An intelligent virtual tutor can provide assistance and guidance during the training sessions; such virtual tutor would replace or complement the real lab tutor. Also, the virtual tutor is available any time. This would also translate into significant cost savings for the departments, which would be able to dramatically expand their manufacturing training; 4) Any number of students can operate and get trained on any machine at anytime. This would represent a dramatic improvement compared to the current situation, where students have to be introduced to the CNC machines in small groups and could only operate the machine and get hands-on-experience for a very short time due to limited lab hours; 5) Practically, it is very difficult (if not impossible) to bring a student, in a semester or two, to a level where he/she would feel comfortable operating CNC machine tools without supervision. Typically, departments will never take the risk of unsupervised operation of machine tools by students. The proposed system

provides the safety necessary to allow such an unsupervised usage of (virtual) machine tool laboratories; 6) The education will become more effective and complete as more sophisticated machine tools (e.g. 5-axis milling machines) and production size units become part of the training. This is not possible currently due to the very high initial investment needed, relatively high overhead cost, as well as space limitations. In addition, the proposed system can promote collaboration and teamwork in advanced manufacturing training. Real world manufacturing processes often require several people to collaborate. This can be difficult to replicate in a university lab but can be simulated in the proposed virtual environment. Furthermore, at the graduate level, the virtual reality system will enable conducting manufacturing experiments that are too difficult to perform in the real world. Next, it would allow interactive exploration of experimental results that are impossible to perform in the real world, as well as the optimization of the manufacturing process plan by testing various plans on the virtual machine before machining on the physical unit.

2.3. ABET Criteria, Program Objectives, and Learning Principles

Engineering & Technology schools would benefit from the developed technology which can be implemented in a number of courses in several departments (including ME, MET, and Computer Graphics) to enhance the pedagogy and better meet the ABET new requirements, programs' outcomes and educational objectives, and Principle of undergraduate learning. As far as our program at IUPUI is concerned, these outcomes and objectives include the following: i) Demonstrating and applying knowledge of mathematics, science, and engineering (Program outcome that is related to ABET criterion *a* and program objectives 1 and 5)¹; ii) Conducting experiments methodically, analyzing data, and interpreting results (Program outcome that is related to ABET criterion *b* and program objectives 1 and 5)². *The proposed technology would allow the undertaking of experiments in the virtual environment*; iii) Designing a system, component, or process to meet desired needs (Program outcome that is related to ABET criterion *a* and program objective 4)³; iv) Use techniques, skills, and modern tools of engineering effectively and correctly in engineering practice, including engineering design and manufacturing tools (Program outcome that is related to ABET criterion *k* and program objectives 4, 5, and 8)⁴.

As a test bed in our curriculum, we are planning to incorporate the new technology in a dual level course CAD/CAM – Theory and Advanced Applications, which is currently taught in the Mechanical Engineering Department. In this course, virtual product manufacturing is conducted on Personal computers. Introducing virtual production on the CNC machine (that the students will use to complete their projects) early in the semester would benefit the students in the initial phase of their product development projects by giving them a sense of how the product they are designing will eventually be produced. This knowledge could be critical to the success and completion of the final project.

2.4. Assessment

The outcomes and benefits would be assessed by the students who would be exposed to this new technology. In fact, a key role in this evaluation process will be played by the students who will have to try both the “virtual reality CNC machine” and the actual CNC machine (i.e. the FADAL CNC machining center, currently available in the machine tool lab). Each student will be asked to complete an evaluation form which would be centered around the extent to which he/she was able, through demos and practice on the “virtual machine”, get a sense of what a real production activity (on the FADAL machine) involves, including the geometric and physical constraints. The benchmark in this case is the actual FADAL machine to which the student will be introduced before completing the survey. Another way of evaluating the effectiveness of the new technology is to divide the students (who are taking the same manufacturing related class) into three groups. The first would be introduced to

¹ Program objectives: 1. Objective#1: Demonstrate excellent technical capabilities in mechanical engineering and related fields; 2. Objective#4: Apply sound design methodology in multidisciplinary fields of mechanical engineering; Objective#5: Competently use mathematical methods, engineering analysis and computations, and measurement and instrumentation techniques; Objective#8: Work collaboratively and effectively in engineering or manufacturing industries.

² See footnote#1

³ See footnote#1

⁴ See footnote#1

the Virtual machine early in the semester (i.e. before starting to work on the product development project). The second would be introduced to the actual machine early in the semester, while the third would have exposure to neither. Next, the project work of all students will be evaluated by the instructor, who would compare the average performance at the manufacturing level of the three groups. Such an evaluation should reveal the effectiveness as well as the benefits of the new technology. In addition, the students from the three groups would be asked to complete a survey that would be designed to assess the impact of the technology. Finally, a group of students from another college (Ivy Tech in Bloomington) will use the virtual training lab to learn how to operate the FADAL machine. At the end of the semester, they will be invited to come to IUPUI to be tested on the actual machine.

3. Virtual Training Environment Architecture

Figure 1 shows the basic architecture of the CNC milling VTE. It consists of three main software modules which communicate with each other using a TCP/IP network socket interface. The three modules are: (1) a CNC milling machine simulator, (2) a virtual-environment display engine, and (3) an intelligent-agent engine. The three modules run on a single computer in a seamless web-based framework. The CNC milling machine simulator consists of the machine logic engine and the machining process simulator. The machine logic engine maintains the current state of the CNC machine and properly propagates state changes. It includes an emulator of the machine controller software and a CNC G-code interpreter. The machining process simulator performs the following functions: discretizing the tool-motion, predicting the cutting forces, and generating the machining sounds.

The virtual-environment is displayed using the IVRESS object-oriented scene-graph virtual-reality toolkit which includes software objects for: (a) displaying a textured photo-realistic geometric model of the machine; (b) boolean-based solid-modeling for geometric simulation of the cutting process; (c) a sounds engine; (d) a humanoid display engine for displaying photo-realistic animated humanoid models of virtual instructor(s).

The intelligent-agent engine has facilities for structured and unstructured knowledge storage and retrieval and a hierarchical rule-based expert system for interpreting and executing the user's natural-language commands. It also has facilities for speech recognition and synthesis. Typical user's commands include: asking the agent to press a machine button, asking the agent to guide the user step-by-step through operating the CNC machine, or asking the agent to describe the function of a machine component. The agent can deliver a lecture on CNC milling as well as a lecture describing the various machine components and theory of operation of the CNC machine. More details on each sub-component of the VTE are presented subsequently.

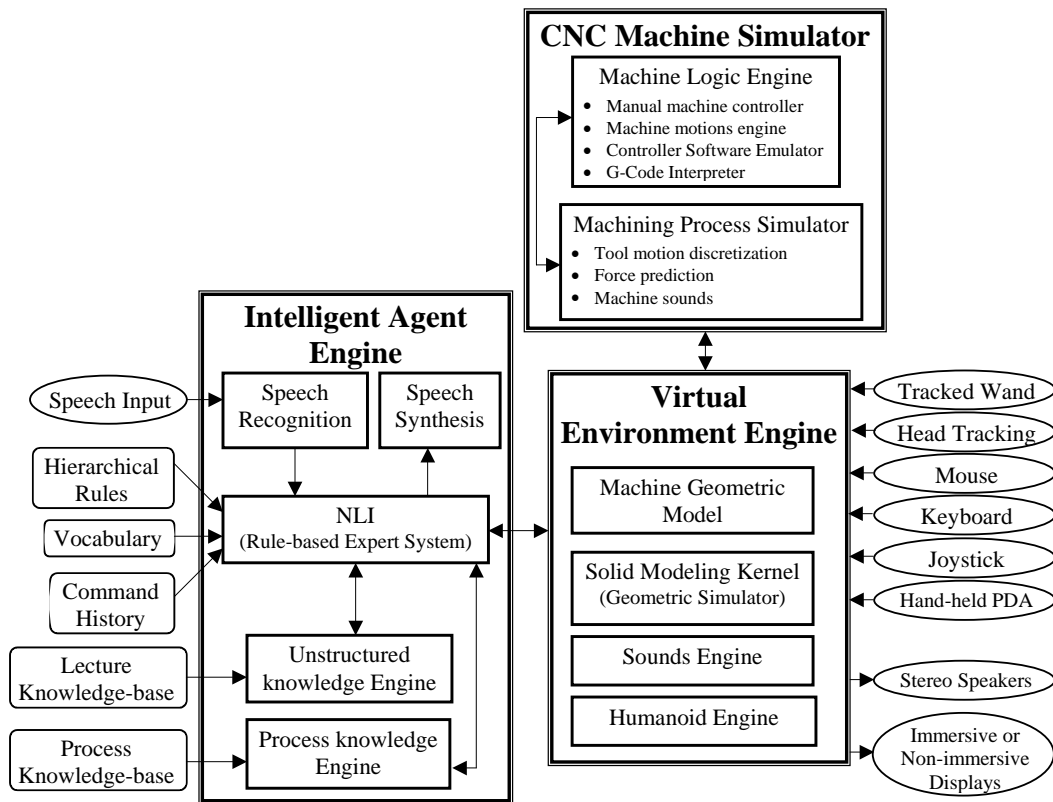


Figure 1: Architecture of the VTE for CNC milling.

4. CNC Machine Simulator

The machine simulator is a virtual representation of the actual CNC milling machine. It simulates the various physical and logical characteristics of the actual machine and the machining process. This module is written as a standalone module using Visual Basic 6.0. It interfaces with the virtual environment engine using a network socket interface. It includes two sub-components: 1) A machine logic engine and 2) A machining process simulator.

4.1 Machine Logic Engine

The machine logic engine simulates the CNC machine, including: 1) A manual machine controller interface; 2) Motions of the various parts of the machine; 3) An emulator of the machine controller software; and 4) A G-code interpreter. The machine logic engine maintains the current state of the CNC machine. Typical state variables are listed in Table 1. The operation logic of the machine is simulated using event-handling subroutines in the controller simulator and the VE.

Table 1: Typical machine state variables

State variable	Description
machinePowerSwitch	Whether the machine power lever is set to <i>On</i> or <i>off</i> .
machinePowerState	Whether the machine is turned <i>On</i> or <i>off</i> .
spindleState	Whether the spindle is on or off.
spindleSpeedStateProgrammed	The programmed spindle speed in RPM
spindelSpeedActual	The actual spindle speed in RPM.
spindelSpeedMultiplier	A multiplier between 0 to 200% for the spindle speed.
xAxisPosition	Absolute current position of the x stage
yAxisPosition	Absolute current position of the y stage
zAxisPosition	Absolute current position of the z stage
aAxisPosition	Absolute current position of the a rotation stage

bAxisPosition	Absolute current position of the b rotation stage
cAxisPosition	Absolute current position of the c rotation stage
Coolant	Whether the coolant is turned on or off
coolantFlow	The coolant flow rate percentage between 0 to 100%
feedRateProgrammed	The programmed feed rate
feedRateActual	The actual feed rate.
feedRateMultiplier	A multiplier for the feed rate between 0 to 150%
toolNumber	Current tool number.
toolOffsetNumber	Tool offset number in the tool offset table.
toolOffset	Current tool offset
motionType	Current motion type: G0: linear rapid, G1: linear interpolation, G2: circular clockwise, G3: circular counter-clockwise.
Unit	G70: inches; G71: mm
coordinateSystem	G90: absolute; G91: incremental
machineDoor	Whether the machine main doors are closed or open.

The manual machine controller interface maintains the state of the manual machine controller. Figure 2 shows machine controller (2-a) and the machine power box (2-b) as displayed in the VE. These two objects are designed using buttons, switches and dials.

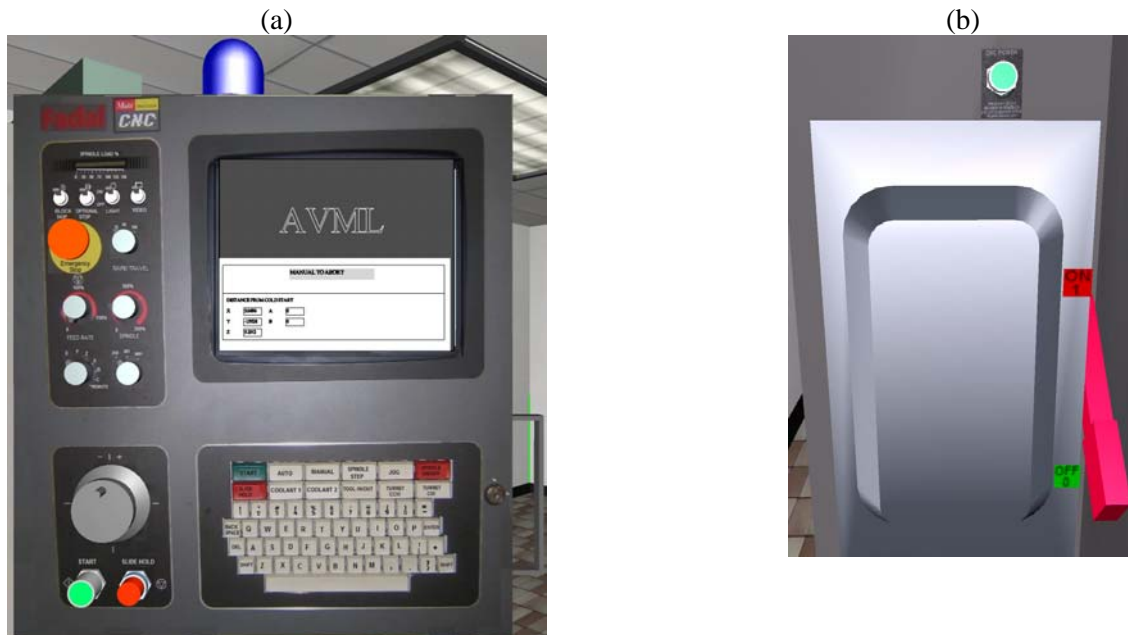


Figure 2: Machine controller and powerbox.

The simulation of the machine logic is done through event-handling and subroutines. The motion in the machine is controlled by the machine logic engine. The axes' motions are not the only motions that are controlled by the machine logic engine. Other motions, such as the turret movement, tool changing, spindle rotation, etc., are also controlled by the machine logic engine, and function basically the same way the axes movement function.

The software emulator as mentioned earlier is designed using Visual Basic 6.0. It consists mainly of forms, buttons, and text label. It models the software which is displayed on the screen of the machine controller. Each click of a button is tied to an event which can also be called by a click of a key in the virtual environment. The software emulator communicates with the Virtual Environment Engine in order to control what is displayed on the screen of the machine controller.

The G-code interpreter functions as follows. First, the whole G-Code program is loaded. The program is then broken down into blocks by the interpreter. These blocks are then broken down into individual G-code words. Each word, when executed will call an event, which will perform the task defined by the word.

4.2 Machining Process Simulator

ACIS[®] open architecture solid modeling kernel is used for the geometric modeling (see Section 5.2). Simulation of milling in the VE requires a real-time generation of the updated part shape and the computation of the tool immersion geometric information (axial and radial depths of cut). The immersion geometric information (needed to compute the cutting forces and other process parameters) is obtained by analyzing the infinitesimal removed material.

For static cutting force prediction, an Artificial Neural Network (ANN) based force model is used [14-17]. The three output parameters of the neural network are the maximum, minimum, and mean resultant cutting forces. The input layer consists of spindle speed, feed rate, axial depth of cut, radial depth of cut, and tool diameter. An IVRESS neural network object is used to predict cutting forces. Experiments that cover the ranges of recommended cutting conditions for different work-piece/tool combinations are used to train the neural network.

5. Virtual Environment Engine

IVRESS object-oriented scene-graph based toolkit is used in the present paper for constructing the VE. Four classes of objects are used to construct the VE:

- Interface objects include user interface widgets (e.g. label, text box, button, check box, slider-bar/dial/knob, table, and graph) as well as container objects (including Group, Transform, Billboard, etc). The container allows grouping objects including other containers. This allows a hierarchical tree-type representation of the VE called the “scene graph”.
- *Geometric entities* represent the geometry of the various physical components. Typical geometric entities include unstructured surface, boundary-representation solid, box, cone and sphere. Geometric entities can be textured using images and colored using the light sources and the material ambient, diffuse, and specular RGBA colors.
- *Finite elements* represent solid and fluid computational domains.
- *Support objects* contain data that can be referenced by other objects. Typical support objects include material color, position coordinates, and interpolators. For example, a sphere geometric entity can reference a material color support object.

All objects have the same basic structure. Each object has properties that determine its state and behavior, and methods, which are functions that it can perform. In addition, interface objects have events that are triggered when certain conditions, initiated by the user or the passage of time, are met. An event is triggered by calling a script-subroutine associated with that event. The subroutine name consists of the object name concatenated with an underscore and the event name (e.g., *object-name_event-name*). IVRESS includes JAVA script and VB script interpreters that allow setting the properties of the various objects, and writing custom event handling routines. In addition, custom objects can be added to IVRESS by writing C/C++ code for the object and linking that code to IVRESS either dynamically (using a dynamic link library), or statically (by linking with an IVRESS static library file). IVRESS can interface with output devices, including immersive stereoscopic screen(s) and stereo speakers; and a variety of input devices, including body tracking (head and hands), haptic gloves, wand, joystick, mouse, microphone, and keyboard. IVRESS can read and write file formats for geometry data such as VRML 2.0 [18], pictures such as Bitmaps, PNG, JPEG, and GIF; and movies such as MPEG, AVI, and MNG.

5.1 Machine Virtual Model

The machine virtual model (Figure 3) is a hierarchical scene-graph that includes the machine geometry and the machine control widgets. The machine geometry consists of textured surfaces of the various machine parts. The machine parts were modeled in Pro-Engineer Wildfire[®] as solid models and then exported in VRML format as tessellated surfaces. The IVRESS “button” widget is used to model the machine buttons, switches, and discrete knobs. The IVRESS “dial” widget is used to model the machine continuous knobs and dial gauges. The speed control widget is used to set the spindle rotational angular velocity at the desired value.



Figure 3: Virtual Model of the CNC machine

Figure 4a shows the IVRESS definition of the axis discrete selection knob. The knob has seven discrete positions namely X, Y, Z, A, B, C or remote axis (see Figure 2a). The knob is defined as an IVRESS ButtonWidget object. This object controls the “rotationAngle” of the “Transform” object that includes the knob geometry. When the user clicks on the knob, the ButtonWidget animates the Transform rotationAngle to the next position in 0.1 sec. Figure 3b shows a typical definition of a continuous knob (Spindle multiplier knob). The user can rotate the knob between angles 0.3 to 5.7 radians from a value of 0 to 200%. When the user clicks on the knob it rotates a value of “smallChange” or 2%. The user can also use the middle scroll mouse wheel to turn the knob up and down.

(a)	(b)
<pre> MODIFY vmc3016_cont_AxisKnob Transform { addChildren [DEF Combo_v3016_axis ButtonWidget { buttonObject USE vmc3016_cont_AxisKnob buttonProperty rotationAngle numPositions 7 position0 2.5 #X position1 3.0 #Y position2 3.5 #Z position3 4.0 #A position4 4.5 #B position5 5.0 #C position6 5.5 #Remote animationTime 0.1 toggle 1 value 0 }] } </pre>	<pre> MODIFY vmc3016_cont_SpindleKnob Transform { addChildren [DEF Slider_v3016_spindleSpeedMultiplier DialWidget { dialObject USE vmc3016_cont_SpindleKnob dialProperty rotationAngle positions [0.3 5.7] values [0 200] value 0 smallChange 2 largeChange 10 speed 60 }] } </pre>

Figure 4: Typical IVRESS object definitions of axis-control selector knob and the spindle speed multiplier knob.

5.2 Solid Modeling Kernel

ACIS[®], a commercial modeler from Spatial Corp., is used here to perform the geometric simulation of the material removal process. ACIS[®] was integrated with the Virtual Environment Engine through the ACIS C++ API. Simulation of milling in the VE requires a real-time generation of the updated part shape and the computation of the different process parameters. This is achieved here through the discretization of the tool motion into steps, calculating the tool swept volume and subtracting the tool swept volume from the work-piece solid model at each incremental position along the tool path. Linear, circular, parabolic and spline interpolations are all approximated using small linear segments. Figure 5 depicts a sample of part updating using ACIS-based geometric simulator.

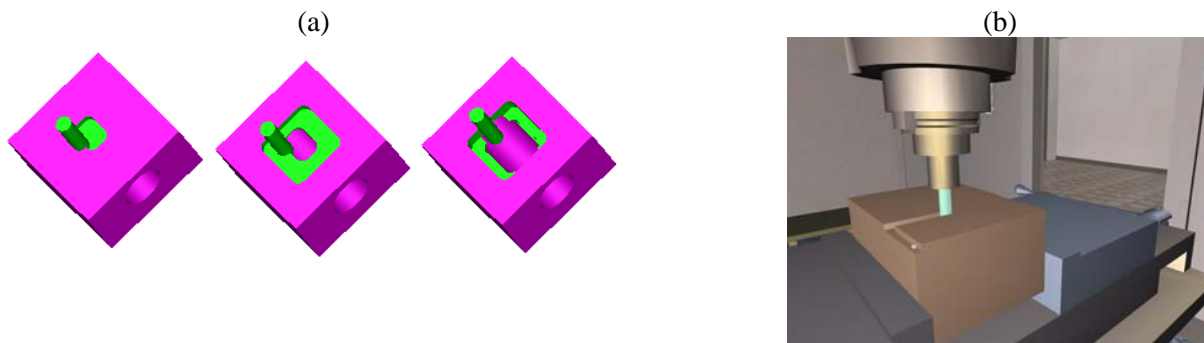


Figure 5: Integration of ACIS into VE.

5.3 Sounds Engine

The sounds are synchronized with the motions of the various parts of the machine by executing a sound play function in the event handling routines of the motions. Multiple sounds can be mixed and played at the same time. The sound generated depends on the location and orientation of the user in the virtual environment.

5.4 Humanoid Model Engine

An IVRESS object that wraps the Haptik humanoid model engine from Haptik Inc. was integrated into the VTE. The object allows loading and displaying full and half body textured highly detailed male and female characters in the VE. A character has a large set of pre-defined gestures. Typical gestures include: looking up, down, right and left; torso bend, twist, and bow; right/left hand; smile; blink; walk; etc. In addition, the gestures also include the visemes (e.g. aa, ih, g, s, eg, uh, etc.) or lip and face positions for lip-synching. Each gesture can take a modifier, which specifies the magnitude/amount of that gesture. Using the IVRESS character wrapper object, the gesture command is sent to a specific avatar in the VE using the script command “Agent_Object_Name.setSw = “talkGestL1 a”” where “setSw” is a property of the character wrapper object and “talkGestL1” instructs the Haptik engine to carry out a talking gesture with an amount of “a”. Also, the Haptik engine allows setting the character’s joints’ rotations and positions to any desired value. The wrapper object allows animation of the character hand motions by linear interpolation of the joint positions or angles.

5.5 Multimodal Interfaces

The IVRESS toolkit enables the user to interface with VE input and output devices through output of sensory information and input of commands (see Figure 1). Output devices include: Immersive stereoscopic displays; and two speakers used for speech and machine sounds output. Input devices include: position and orientation tracking devices for tracking the position and orientation of the user’s body; tracked wand; joystick.; mouse; keyboard; microphone for voice input; handheld computer (the user can control the VE using a hierarchical graphical menu on a handheld PDA).

6. Intelligent Agent Module

The LEA intelligent agent system includes speech recognition and synthesis components, a rule-based expert system natural-language interface (NLI), a hierarchical process knowledge base engine, and an unstructured knowledge base engine.

6.1 Speech Recognition

The user’s speech is acquired using a microphone. Any Microsoft SAPI 5.1 compliant speech recognition engine can be used for speech recognition. LEA uses primarily the single word/short phrase recognition SAPI speech recognition mode. A SAPI file which includes all the words that can be used in the CNC manufacturing training application as well as rules for common short phrases is used. LEA can also use the continuous dictation SAPI speech recognition mode.

6.2 Speech Synthesis

Any Microsoft SAPI 5.1 compliant speech synthesis (text-to-speech) engine can be used for generating the agent's speech. The LEA engine sends SAPI the text string that is to be spoken. SAPI generates the speech along with the following events:

- *Start of sentence event.* This event returns the starting and ending character positions of the sentence that is currently being spoken. This event is used by the NLI to highlight the sentence that is currently being spoken as well as to run any scripts that is contained within the sentence.
- *End of word event.* This event returns the starting and ending character positions of the word that is currently being spoken. This event is used by the NLI to highlight the word that is currently being spoken.
- *Viseme events.* These events are generated in order to do the lip synchronization. They are passed by the NLI to the VE engine, which in turn passes them to the agent avatar display module to place the lips of the agent avatar in the proper position.

6.3 Natural Language Interface

The function of the NLI is to provide two-way communication with the user in natural-language speech (and/or written text). The NLI accomplishes natural-language understanding by converting the user's natural-language commands (as well as pointing gestures) to script that can be sent to the VE. This script can query or change the properties of objects in the VE. The avatar is an object in the VE and hence it can be controlled in the same way as other VE objects. Thus, the script can be used to animate the avatar, including arm motions, walking, facial expressions, and lip-synching. The NLI communicates with the user by sending output speech to the speech synthesis engine, as well as by changing the visual state of objects in the VE. The NLI includes the following facilities:

- Hierarchical rule-based expert system engine [9] for speech understanding. The expert system receives the natural-language input of the user and converts it to VE engine script. The hierarchical rules approach takes advantage of the object-oriented hierarchical data structure of the VE by organizing the rules into three main types, namely, object, property, and action rules.
- VE interface for receiving and setting VE object property values through a TCP/IP network socket connection.

6.4 Hierarchical Process Knowledge Engine

The hierarchical process knowledge-base enables the agent to have knowledge about a sequence of steps that accomplish a specific objective and the consequences of mistakes [1]. Each process consists of a set of steps as well as other sub-processes. Each process and step can have pre- and post- constraints. Pre-constraints have to be satisfied before the step/process can be started. Post-constraints have to be satisfied before the step/process is completed. The agent can disseminate process knowledge using one of the following training modes. The desired mode is triggered using the rule-based expert system. The training modes are:

- *Process tutor.* In this mode, the agent performs the process steps while the user is watching. The user can pause/resume, repeat (go back) a step, or skip a step.
- *Process info.* This mode is similar to the tutor mode except that the agent will only recite the process steps to the user without demonstrating how they are done.
- *Process guide.* The agent guides the user step by step through the process. The agent will not go to the next step until the user says a command such as "go on", "continue", and "proceed". The user has to perform each step of the process. The agent checks the process constraints to determine if the user performed the step correctly. If a constraint is violated, then the agent instructs the user to repeat the step. If the user does not perform the step correctly three times in a row, then the agent performs the step.
- *Process supervisor/certification.* In this mode the agent instructs the user to perform the process. At the end of the process the user lets the agent know that s/he is done. At that point the agent checks the process constraints. If no mistakes are detected then the agent certifies the user in this process.

6.5 Unstructured Knowledge Engine

The agent can deliver a multimedia lecture. The lecture is stored as unstructured HTML knowledge items that are linked to a hierarchical outline. Two lectures are included in the VTE. A lecture on the basics of CNC milling and a lecture describing the various components of the CNC milling machine.

7. Education and Training

Figure 6 shows the web interface to the VTE. The interface consists of the following windows: 1) A hierarchical outline of the introductory lecture(s); 2) A hierarchical list of natural-language voice commands (this helps the user to become familiar with the available voice commands. The user can click on a command from the list instead of speaking or typing it); 3) An agent speech display window; 4) An agent options dialog box; 5) A multi-media display screen used to display multi-media content associated with the lectures; and 6) The VE display screen. The VE screen can be sized to cover the multi-media screen. All the windows can be sized and moved by the user to suit his/her preferences. Two agents provide assistance to the user in the VTE:

1. An "always on top of the screen" agent or screen agent is always visible and provides guidance and answers to the user's questions.
2. An assistant in the environment directly assists the user by showing the correct procedure to perform the required process steps.

Both agents are controlled using the LEA engine. The two agents collaborate to help the user. For example, when the user needs help, he asks the screen agent a question. The screen agent will say the answer and display any visuals, which support the answer. The screen agent will also command the assistant in the VE to actually do a demonstration if necessary. Visual cues such as 3D flashing arrows are used in addition to the agent in the environment pointing to the correct objects in the environment.

The CNC milling machine VTE provides the following training functions for students:

- *Introductory lecture to CNC milling.* Before the hands-on training starts, the screen agent can give the trainee a brief introductory overview on CNC milling. The lecture outline is in the window labeled "Introduction". This allows the user to skip to specific points in the lecture or return to previous points or even skip the lecture entirely and proceed with the hands-on training right away. In Figure 6, the screen agent is describing the types of CNC milling machines.
- *Introductory lecture to the FADAL CNC machine.* The screen agent automatically proceeds to introduce the FADAL CNC machine. The basic components of the machine are introduced. It functions the same as the introductory lecture to CNC machining process. In Figure 6b, the tutor is describing the base of the FADAL CNC machine. The other parts of the machine are made translucent so that the user can see the base clearly
- *Process training.* is described in Section 7.1.
- *Practicing to use the CNC machine.* The student can use the virtual CNC machine the same way as s/he would use the actual machine.

(a)

Advanced Virtual Manufacturing Lab - Microsoft Internet Explorer

Advanced Virtual Manufacturing LAB (AVML)

INTRODUCTION

- Play Stop Pause
- Lab Overview
- Introduction to Machining Centers
 - Characteristics of Modern CNC Machine Tools
 - Types of Machining Centers
 - Pallet Changers
 - Cutting Tools
 - Fixtures
- Programming of Machining Centers
 - Planning
 - Coordinate System
 - Speed and Feed Calculation for Milling
 - Types of Programming
 - Incremental Programming
 - Absolute Programming
 - Programming Area
 - Programming Hole Cycles
 - Tool Nose Radius Compensation
- Overview of the FADAL CNC Machining Center
- Intelligent Agent
- Skip Intro

VOICE COMMANDS

- Views
 - goto room entrance view
 - goto machine front view
 - goto machine left side view
 - goto machine right side view
 - goto machine controller view
 - goto machine work piece view
- brightness
- world scale
- background color
- Agents
 - screen agent
 - visibility
 - hide screen agent
 - show screen agent
 - position
 - transparency
 - machine agent
 - move machine agent to room entrance
 - move machine agent to front of machine
 - move machine agent to left side of machine
 - move machine agent to right side of machine

Media Screen

hide screen agent

Advanced Virtual Manufacturing Lab

Head Spindle Motor

Column Cutting Tool

Table Saddle

Z

X

Y

3-axis Machining Centers

Types of Machining Centers

SPEECH

of machining centers differentiated by the number of programmable axes. A three axis machining center has programmable x and y axes in the plane of the table and a z axis in the spindle's direction. This is the most basic type of machining center, and they start at about 30,000 dollars. Three axis machines can machine one surface of a cube with the end of the cutter and four additional surfaces with the side of the cutter. Typically, three axis machines are in the vertical configuration shown here. Please note that the tool changer, control, and enclosure are not shown in the schematic drawing. This illustration depicts a three axis horizontal machining center. Note the different orientation of the x, y, and z axes. This type of machine starts at about 50,000 dollars. Again, the tool changer, control, and enclosure are not shown. This configuration can still only machine one surface of a cube with the end of the cutter and four additional surfaces with the side of the cutter. Four axis machining centers are generally horizontal, and the table rotates to create the fourth axis as shown here. True four axis machines start around 100,000 dollars. This configuration can be achieved by adding a CNC-controlled rotary table to a three axis vertical machine, and this is commonly done for small parts. It can machine four surfaces of a cube with the end of the cutter and two additional surfaces with the side of the cutter. It is essential to spend about 150,000 dollars for a true four axis machine, making that it is very similar to

VIRTUAL ENVIRONMENT

AGENT OPTIONS

- Listen More
- Use dictation Det.
- Speak
- Speed
- Volume
- Recognized Words & Phrases

(b)

Advanced Virtual Manufacturing Lab - Microsoft Internet Explorer

Advanced Virtual Manufacturing LAB (AVML)

INTRODUCTION

- Play Stop Pause
- Lab Overview
- Introduction to Machining Centers
- Overview of the FADAL CNC Machining Center
 - Overview of the FADAL CNC machining center
 - Operation of the FADAL CNC machining center
- Programming of the FADAL CNC Machining Center
- Machine components
 - Base
 - Column
 - Z-Stage
 - X-Y-Stages
 - Tool changer
 - Machine controller
 - Coolant system
 - Body sheet metal
 - Machine doors
 - Power junction cabinet
 - Control box
- Intelligent Agent
- Skip Intro

VOICE COMMANDS

- Views
 - goto room entrance view
 - goto machine front view
 - goto machine left side view
 - goto machine right side view
 - goto machine controller view
 - goto machine work piece view
- Collision detection
- View mode
- Viewpoint
- Stereo viewing
- Brightness
- World scale
- Background color
- Agents
 - Model

VIRTUAL ENVIRONMENT

SPEECH

This is the machine base, also called "bed". The machine base must be stiff and massive in order to provide a rigid and stable platform for the entire CNC machine. The more rigid and massive the machine base, the higher the passive vibration damping and thus the higher the accuracy of the machine. The base is bolted to the floor to provide additional rigidity.

AGENT OPTIONS

- Listen More
- Use dictation Det.
- Speak
- Speed
- Volume
- Recognized Words & Phrases

Figure 6: Web interface of the VTE

7.1 Process Training

Training on typical procedures pertaining to the operation of the FADAL[®] machine and its use in milling is available in different modes. Examples of procedures currently supported include: 1) Starting up the machine; 2) Getting the machine ready for the execution of an NC code; 3) Running the machining operation from a G-code that is download to the controller from the user's computer; 4) Safely turning off the machine. Some of the procedures are available in one mode others in multiple modes. Presently, the AVML supports three different modes of operations. First, the user can be trained by asking the tutor to "Show him/her" the procedure, called, *Tutor Mode*. Second, the user can be guided by the intelligent tutor when learning a certain procedure, called *Guide Mode*. Third, the tutor can supervise the user and only interfere when the former does a mistake, called *Supervisor Mode*. Fourth, as an extension of the latter mode, the user can also be certified by the tutor, called *Certification Mode*. Finally, the user can interact with the machine and its controller, called *Navigation mode*.

8. Concluding Remarks

In this paper, the architecture of a virtual training environment (VTE) was presented and used to develop the corresponding system for the case of CNC milling. The proposed environment is unique as it combines all the components needed for a realistic, comprehensive and accurate modeling and simulation of both the machine tool and the machining process. The methodology is generic and could be extended to other manufacturing units as well as processes, ultimately leading to the Advanced Virtual Manufacturing Laboratory (AVML) for teaching, training and research. The current implementation has focused on the particular case of a vertical CNC machining center where milling is conducted. The resulting virtual system and environment provided the following training functions for users: (a) access to a fully-functional virtual CNC milling machine, (b) training on key operating procedures of the CNC machine, (c) lectures on CNC machine tools as well as CNC milling. The environment is driven by three software modules that communicate with each other using a TCP/IP network socket interface, namely, a CNC Milling machine simulator, a virtual-environment display engine, and an intelligent-agent engine. The modules run on a single computer in a seamless web-based framework. The current capabilities and system's functionality will be extended to achieve fully functional CNC machine tools where teaching and training of advanced manufacturing can be undertaken efficiently, safely, and economically.

Acknowledgments

The authors would like to thank NSF for partial support of this work under NSF grant number DMI-0339024. This work was also partially funded by Indiana 21st Century fund (grant number 0043). The authors would like to thank Spatial Corp. for providing the ACIS solid modeling kernel. The humanoid avatars display engine was provided by Haptik Inc. IVRESS and LEA were provided by Advanced Science and Automation Corp.

9. References

- [1] Wasfy, A., Wasfy, T., and Noor, A., "Intelligent Virtual Environment for Process Training," *Advances in Engineering Software*, Vol. 35(6), 2004, pp. 337-355.
- [2] Lin, F., Ye, L., Duffy, V.G., Su, C-J., "Developing virtual environments for industrial training," *Information Science*, Vol. 140, 2002, pp. 153-170.
- [3] Loftin, R.B., Wang, L., Baffes, G., Hua, G., "An intelligent agent training system for space shuttle flight controllers," *Telematics and Informatics*, Vol. 5(3), 1988, pp. 151-161.
- [4] Vora, J., Nair, S., Gramopadhye, A.K., Duchowski, A.T., Melloy, B.J., and Kanki, B., "Using virtual reality technology for aircraft visual inspection training: presence and comparison studies," *Applied Ergonomics*, Vol. 33, 2002, pp. 559-570.

- [5] Mayros, J., Kesavadas, T., Chugh, K., Joshi, D, and Ellis, D.G., "Utilization of virtual-reality for endotracheal intubation training," *Resuscitation*, Vol. 59, 2003, pp. 133-138.
- [6] Nakao, M., Oyama, H., Komori, M., Matsuda, T., Sakaguchi, G., Komeda, M., and Takahashi, T., "Haptic reproduction and interactive visualization of a beating heart for cardiovascular surgery simulation," *International Journal for Medical Informatics*, Vol. 68, 2002, pp. 155-163.
- [7] Tsai, M-D., Hsieh, M-S., Jou, S-B., "Virtual reality orthopedic surgery simulator," *Computers in Biology and Medicine*, Vol. 31, 2001, pp. 333-351.
- [8] <http://www.ascience.com/Science/LEA.htm>
- [9] Wasfy T., Noor A., "Rule-Based Natural-Language Interface for Virtual Environments," *Advances in Engineering Software*, Vol. 33, 2002, pp. 155-168.
- [10] <http://www.ascience.com/Science/IVRESS.htm>
- [11] Wasfy, T. and Noor, A., "Object-oriented virtual reality environment for visualization of flexible multibody systems," *Advances in Engineering Software*, Vol. 32(4), 2001, pp. 295-315.
- [12] H. El-Mounayri, M. Elbestawi, A. Spence, and S. Bedi, "General Geometric Modelling Approach for Machining Process Simulation," *International Journal of Advanced Manufacturing Technology*, v. 13, pp. 237-247, 1997.
- [13] H. El-Mounayri; A. Spence and M. Elbestawi, "Enhanced CAD/CAM for Simulation and Optimization of 3-5 axis Milling of Dies and Molds," *Proceedings of the 13th Symposium on Engineering Application of Mechanics*, 1996 CSME Forum, McMaster University, ON, Canada, pp. 394-401, 1996.
- [14] V. Tandon, H. El-Mounayri, and H. Kishawy, "NC End Milling Optimization Using Evolutionary Computation", *International Journal of Machine Tools and Manufacture*, v. 42, pp. 595-605, 2002.
- [15] H. El-Mounayri, H. Kishawy, and V. Tandon, "Optimized CNC End Milling: A Practical Approach", *International Journal of Computer Integrated Manufacturing*, v. 15, pp.453-470, 2002.
- [16] J. Briceno, H. El-Mounayri, and S. Mukhopadhyay, "Selecting an Artificial Neural Network for Efficient Modeling and Accurate Simulation of the Milling Process", *International Journal of Machine Tools and Manufacture*, v. 42, pp. 663-674, 2002.
- [17] V. Tandon and H. El-Mounayri, "A Novel Artificial Neural Networks Force Model for End Milling", *International Journal of Advanced Manufacturing Technology*, v. 18, pp. 693-700, 2001.
- [18] ISO/IEC 14772-1: 1997 Virtual Reality Modeling Language (VRML97), The VRML Consortium Incorporated, 1997.