

EXPERT SYSTEMS

*The Technology of
Knowledge Management
and Decision Making
for the 21st Century*

VOLUME 2

Edited by

Cornelius T. Leondes

*Professor Emeritus
University of California
Los Angeles, California*



ACADEMIC PRESS

A Harcourt Science and Technology Company

San Diego San Francisco New York Boston London Sydney Tokyo

This book is printed on acid-free paper. ∞

Copyright © 2002 by ACADEMIC PRESS

All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher.

Requests for permission to make copies of any part of the work should be mailed to: Permissions Department, Harcourt, Inc., 6277 Sea Harbor Drive, Orlando, Florida 32887-6777.

The appearance of the code at the bottom of the first page of a chapter in this book indicates the Publisher's consent that copies of the chapter may be made for personal or internal use of specific clients. This consent is given on the condition, however, that the copier pay the stated per copy fee through the Copyright Clearance Center, Inc. (222 Rosewood Drive, Danvers, Massachusetts 01923), for copying beyond that permitted by Sections 107 or 108 of the U.S. Copyright Law. This consent does not extend to other kinds of copying, such as copying for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale. Copy fees for pre-2001 chapters are as shown on the title pages. If no fee code appears on the title page, the copy fee is the same as for current chapters. \$35.00

Academic Press

A Harcourt Science and Technology Company
525 B Street, Suite 1900, San Diego, California 92101-4495, USA
<http://www.academicpress.com>

Academic Press

Harcourt Place, 32 Jamestown Road, London NW1 7BY, UK
<http://www.academicpress.com>

Library of Congress Catalog Card Number: 00-2001089412

Set International Standard Book Number: 0-12-443880-6

Volume 2 International Standard Book Number: 0-12-443882-2

PRINTED IN THE UNITED STATES OF AMERICA

01 02 03 04 05 06 HP 9 8 7 6 5 4 3 2 1



INTELLIGENT SYSTEMS TECHNIQUES AND THEIR APPLICATION IN MANUFACTURING SYSTEMS

TIEN-FU LU

Department of Mechanical Engineering, Adelaide University, South Australia

GRIER C. I. LIN

*Centre for Advanced Manufacturing Research, University of South Australia,
Australia*

I. INTRODUCTION	381
II. MANUFACTURING PROBLEMS AND TRENDS	382
A. Manufacturing Control Problems	382
B. Global Trends	383
III. INTELLIGENT SYSTEM TECHNIQUES	384
A. Software Aspect—Knowledge Manipulation	386
B. Hardware Aspect—Information Retrieval	389
IV. APPLICATION OF INTELLIGENT SYSTEM TECHNIQUES IN MANUFACTURING—DEVELOPMENT OF AN INTELLIGENT WORKCELL ROBOT SYSTEM	392
A. Introduction	392
B. System Hardware Overview	393
C. System Software Overview	395
D. System Working Principle	400
E. System Programming	401
F. System Online Performance	406
G. Discussion and Conclusion	408
V. CONCLUSION	408
REFERENCES	409

I. INTRODUCTION

Artificial intelligence (AI) has been claimed to have yielded revolutionary advances in manufacturing since the 1980s. It can be defined as the development of techniques, that can be used to reproduce the human ability of making primitive

judgments and reacting accordingly by logical arguments in computers and other machines. At a minimum, intelligence will show the ability to sense the environment, make decisions, and control action. Higher levels of intelligence may include the ability to recognize objects and events, to represent knowledge in world models, and to reason about and plan for the future.

A system can be defined as a collection of interrelated elements brought together to achieve a specified objective [1]. System intelligence therefore concerns how to integrate the interrelated elements to reproduce human-like capabilities such as learning, observation, perception (sense the environment), interpretation, reasoning, planning, decision-making, and controlling action [2]. Systems that can simulate some degree of the human-like capabilities are called intelligent systems. They use the capabilities of solving problems that cannot be attempted using routine algorithms or other approaches of computing sciences [3].

Manufacturing is the process of transforming natural resources and basic raw materials into goods for the consuming public. It is the cornerstone of all economic activities and the pivotal link to a richer and stable future [4]. However, recent socioeconomic changes have generated many problems, which could threaten the foundation of a country's manufacturing industry. For example, a dynamic change in market requirements on a global scale has resulted in shorter lead times for production and more diversified needs. To deal with such trends efficiently, the international manufacturing community has launched a feasibility study to establish a framework for developing future intelligent manufacturing systems (IMS) since Japan submitted its international collaborative research program in October 1989 [5-8].

Intelligent manufacturing systems are recognized as being very important for resolving the problems associated with socioeconomic changes [9]. Many research and development topics have therefore been identified and are in the process of being investigated for the construction of IMS. These topics cover production system development technologies, production-related information and communication technologies, production control equipment and processing technologies, application technologies for new materials, and human factors in production [6].

To construct intelligent systems, AI techniques have to be further developed and appropriately integrated. In this chapter, common manufacturing system control problems and some global trends are first reviewed. Then some intelligent systems techniques for software and hardware are introduced. A discussion of implementing intelligent systems techniques in manufacturing follows. Finally, one example, the development of an intelligent workcell robot system, is presented.

II. MANUFACTURING PROBLEMS AND TRENDS

A. Manufacturing Control Problems

In any manufacturing system, control always plays a very important role. Modern large-scale manufacturing systems are often equipped with complex components such as industrial robots and automatically guided vehicles. Because there are so many different components integrated together, the control issues for these manufacturing systems become very complicated and difficult. In addition, the compo-

nents that could be used for different applications must be programmed explicitly for each application. This is extremely costly.

In practical manufacturing systems, the performance of traditional controllers deteriorates for the following reasons:

- Many manufacturing systems are inherently highly nonlinear, time varying, and complex with unknown model parameters.
- There exist unexpected disturbances, such as load changes, unscheduled changes of working conditions, system parameter changes, and so on.

It is thus very difficult to obtain a suitable mathematical model of a complicated manufacturing system that is precise enough and is suitable for the analysis and design of a controller based on conventional control theory and methods.

Once a system condition changes and performance deteriorates, traditional control system design techniques have few means to deal with these. The traditional controllers usually can do nothing but repeat poor control quality all the time, unless it is modified by human intervention. This is mainly due to the lack of effective procedures to utilize system sensory information in real time and produce the proper modification action accordingly to rectify the problem [10]. This situation therefore points out the need for intelligence in manufacturing systems.

B. Global Trends

Recent rapid progress in technology provides new approaches and imposes new requirements for future manufacturing systems. The new requirements relevant to IMS include the following:

- The market for products is becoming increasingly internationalized.
- There is a need to improve planning and control algorithms, which may include AI techniques.
- Knowledge-based systems and AI have reached the realm of applicability.

To meet the new requirements, certain trends can be predicted for future manufacturing systems, such as

- human-free factory;
- remote-controlled intelligent factory;
- self-fault diagnosis;
- information fusion;
- natural language control and programming; and
- distributed intelligent subsystems to handle the relevant problem-solving activities.

These trends mainly focus on finding new ways of increasing productivity and system flexibility and developing new technologies that will embed appropriate intelligence in manufacturing systems. The aim is to achieve the following:

- shorter product life cycles;
- increased emphasis on quality and reliability;
- more customized products;
- new materials;
- growing use of electronics;

- pressure to reduce inventories;
- just-in-time production;
- point-of-use manufacture; and
- greater use of computers in manufacturing.

Other important trends in technological systems are to break down rigidity and exponentially expand complexity by decomposing systems into smaller, more manageable units that behave as minisystems in their own right, but which can be coordinated as subsystems of a central intelligent supersystem [11]. In robotic systems, for example, microprocessors provide the technical foundation for a decentralization of control by embedding "intelligence" in the sensors, operating units, and peripheral elements of the larger systems.

To resolve manufacturing control problems and meet global trends, embedding intelligence in manufacturing systems has been recognized to be a very effective approach. As a result, many of intelligent systems techniques have been explored and developed, and attempts to integrate them with traditional manufacturing systems have been made.

III. INTELLIGENT SYSTEM TECHNIQUES

Intelligence is the result of learning. Intelligent systems are therefore those that are capable of learning, responding, and reacting to the physical world without or with minimal human assistance. Intelligent techniques are those which can be adopted and integrated to comprise intelligent systems that are able to carry out sophisticated tasks unaided and learn from experience in a changing environment.

The essence of intelligent systems is the capability to collect knowledge of the situation gained at execution time and correlate it with other knowledge to take effective actions for achieving goals. Intelligent systems can be composed of sensors for perceiving both the external and internal environments, actuators for acting on the world, and computer hardware and software systems for providing connections between the sensors and actuators as pictured in Fig. 1.

World Models. In general, a world model stores representations of the internal and/or external world at a high level of abstraction. The world model is the intelligent system's best estimate of the state of the world. It can be used to generate expectations and predictions. Thus, it can provide answers to requests for information about the present, past, and probable future states of the world. By all means, the world model must be kept up-to-date to have reasonable estimates of the state of the world all the time.

Signal Processing. Signal processing concerns the techniques used to enhance and transmit useful information from measurements acquired by sensors in a digital or analog form. It can take real-time, high-speed information, such as radio, sound, or video signals, and manipulate it for a variety of purposes. For the last 25 years, there has been enormous progress in the development and applications of signal processing, especially digital signal processing. Its impact is being felt in applications as diverse as stereo systems, cars, personal computers, and cellular phones. This is mainly due to the availability of more powerful hardware chips and software algorithms at rapidly reducing costs.

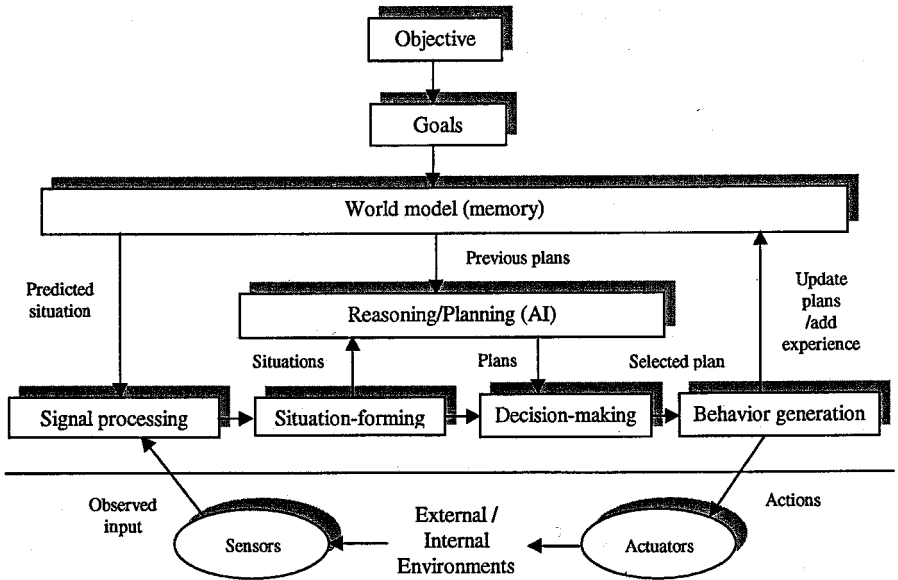


FIGURE 1 The elements of an intelligent system and their relationships.

In Fig. 1, the signal-processing unit represents the hardware and software developed to deal with those signals from various sensors. The main goal of the signal-processing unit is to extract meaningful and useful information (such as features, objects, and their relationships) from the sensory measurements for the intelligent system to further manipulate.

Situation-Forming. Soon after the signal-processing unit has retrieved meaningful and useful information, the situation-forming unit will have the capability to put those bits and pieces together to form possible current internal and/or external situations. Criteria, which are set in advance, can be used to match the retrieved information with situations. Of course, it would be a fuzzy process to form situations based on those sensory measurements that are not as simple as black and white.

The possible situations then will be fed into the reasoning/planning unit. Plans corresponding to those situations will thus be generated or retrieved and reviewed by the decision-making unit.

Reasoning/Planning. Planning implies an ability to predict future states of the world. Planning enables a system to take a high-level goal and decide which subtasks must be accomplished to move the system toward the goal. The reasoning/planning unit monitors the execution of task plans and modifies existing plans whenever the situation requires. Predicting the future state of the world often depends on assumptions as to what actions are going to be taken and what reactions are to be expected from the environment. In a real-time system, plans must be regenerated periodically to cope with changing and unforeseen conditions.

Before making the final decision on an action, planners shall direct relevant sensors through sensing actions to improve the estimates of the environment until

the sensing information is sufficient and unambiguous. Then, planning agents will determine the best actions.

Decision-Making. The decision-making unit evaluates both the observed state of the world and the predicted results of hypothesized plans. It may compute costs, risks, and benefits for both observed situations and planned activities. The decision-making unit thus provides the basis for choosing one action rather than another.

Behavior-Generation. Once the decision-making unit has selected an appropriate plan according to the current system situation, the behavior-generation unit will generate a sequence of system actions corresponding to the selected plan. Eventually these actions will be interpreted into hardware-dependent instructions for system actuation and control.

A. Software Aspect—Knowledge Manipulation

Because the decision-making process in the advanced manufacturing system environment is becoming increasingly complicated, difficult, and overwhelming to humans, AI has been widely adopted to assist human efforts [12]. In the area of manufacturing control, expert systems support the operator to handle complex situations and the maintenance staff to diagnose causes of failure. Knowledge-based systems are the main part of autonomous decision-making systems that show new solutions in robotics and automation. In addition, neural networks allow fast processing of extensive video data. Fuzzy technology provides a basically new approach in handling of uncertain data in general. All these approaches will revolutionize applications and systems that deal with high complexity and uncertainty [13].

I. Machine Learning

Machine learning deals with computational methods for acquiring new knowledge, new skills, and new ways to organize existing knowledge. The major aim is for the machine to learn to classify unseen objects that are not contained in the training set. To date, learning concepts from examples is still the most common and best-understood form. For learning, a language is normally needed to describe the relationships between objects and classes. This language is called "hypothesis language", because it is used for stating hypotheses about what the target concept could be.

Various approaches to machine learning can be characterized by the description languages they use. Roughly, there are two kinds of descriptions: relational descriptions and attribute-based descriptions.

The relational description (also called structured description) describes an object in terms of its components and the relationships between them. A relational description of the concept of an arch may state the following: an arch is a structure consisting of three components (two posts and a lintel), each of the three components is a block, both posts support the lintel, and the posts do not touch. Such a description is relational because it talks about the relationships between components [14].

In an attribute-based description, an object is described in terms of its global features. Such description is a vector of attribute values. An attribute description may, for example, mention the attribute's length, height, and color. Attribute-based descriptions are a special case of relational descriptions, in which all relationships are unary.

2. Neural Networks (Models of the Brain)

Neural networks (NNs) were defined by Kohonen in 1988 as

massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations which are intended to interact with objects of the real world in the same way as biological nervous systems do [15].

Originally, the inspiration came from the nervous systems of higher organisms, most notably from the human brain. It is assumed that a neural network is characterized by [16] the following features.

- Parallel architecture: it is composed of many self-contained, parallel interconnected processing elements or neurons.
- Similarity of neurons: each basic processor is described by a standard nonlinear algebraic or differential equation.
- Adjustable weights: there are multiplicative parameters, each associated with a single interconnection and they are adaptive.

A neural network does not execute a series of instructions as conventional programming languages. It responds, in parallel, to the inputs that are presented to it. The result is not stored in a specific memory location but consists of the overall state of the network after it has reached some equilibrium condition. For a very wide range of applications in engineering and information technology, NNs offer a complementary and potentially superior approach to that provided by conventional computing. This is because the conventional approach requires a sufficient understanding of a problem to write appropriate computer programs for acceptable solutions. In situations in which understanding of a problem is incomplete, computer programs are necessarily based upon assumptions and approximations that are sometimes hard to justify. But when NNs are applied to a problem that is not fully understood, the need to make assumptions or approximations hardly ever arises because NNs essentially learn by experience and therefore can be trained to implement complex tasks for which conventional approaches tend to fail. This characteristic therefore makes NNs very suitable to be implemented in manufacturing where the models of systems are highly complicated or incomplete [15].

The current applications of NNs in manufacturing cover group technology for part classification and part family formation, engineering design for product design processing, monitoring and diagnosis, process modeling and control, quality assurance, scheduling for resource allocation, and process planning.

3. Expert Systems (Models of the Mind)

Expert systems is a branch of AI. It offers a way to capture and encode knowledge from experts. An expert system generally consists of three main parts:

- the explanation generator and user interface;
- the inference engine; and
- the knowledge base [17].

An expert system is different from conventional programming because it does not process data sequentially. Instead, the sequence in which rules are executed is determined by the information available and desired. An expert system uses rules, but such a system does not learn these rules by itself. A knowledge engineer determines them in meetings with the experts and manually writes each individual rule into the program.

Since the mid-1970s, AI methods have been continuously developed and applied by the manufacturing industry. Expert systems have been the most successful implementation of AI. There have been an increasing number of applications in the manufacturing environment including design, machine fault diagnosis, system configuration, interpretation of data, and monitoring.

4. Fuzzy Logic

Fuzzy logic was invented in the 1960s by Professor Lotfi Zadeh of the University of California, Berkeley, and has become the most successful and easy way to implement engineering systems yet found [18]. It provides a scheme and a calculus for dealing with vague or uncertain concepts. Fuzzy logic has been applied successfully in many areas in which conventional model-based approaches are difficult or not cost-effective to implement. It applies more humanlike way of thinking in the programming of computers. Basically, fuzzy logic can be characterized by the following.

- Fuzzy logic is a multivalued logic. It allows any degree of value from 0 to 1 to be assigned in a fuzzy set, such as gray instead of black or white. This is different from crisp logic, where given items are either members of a given set or they are not.
- There must be a set of rules, where the conjunction “and” calls for the minimum membership of the topic being considered to yield an output membership value for decision-making purposes.

Fuzzy logic is easy to implement, easy to understand, and relatively cheap to develop and maintain. The ease of development and understanding is at the heart of the growth of fuzzy logic. Because fuzzy logic is based on human reasoning, it is easy to write down the fuzzy rules that are required to operate a machine or process with no complex mathematics involved. The task is easy to do and when complete, it is transparent to others. It was also shown that fuzzy logic adapts well to control nonlinear systems and systems with no modeled parameters, which have long proved unstable when attempts were made to control them by conventional continuous linear controllers.

Fuzzy logic is now routinely used for the entire span of engineering applications and products, such as motor cars, industrial processes, cameras, automatic washing machines, and other consumer devices.

5. Genetic Algorithms (Models of Evolution)

A genetic algorithm (GA) is reminiscent of sexual reproduction in which the genes of two parents combine to form their children. When it is applied to solve problems, the principle is that we can create an initial population of individuals representing possible solutions to a problem we are trying to solve. All of these individuals have certain characteristics that make them more or less fit as members

of the population. The most fit members will have a higher probability of mating than the less fit members to produce offspring that have a significant chance of retaining the desirable characteristics of their parents. This method is very effective for finding optimal or near-optimal solutions to a wide variety of problems, because it does not impose many of the limitations required by traditional methods [19].

A GA is an elegant generate-and-test strategy that can identify and exploit regularities in the environment and converges on solutions that are globally optimal or nearly so. For many systems that cannot be adequately modeled or whose state space is too large, a genetic algorithm should be considered, which can efficiently search for the system's optimal parameter values without having to rely on prior knowledge of the performance space.

In addition, GAs have been used in many other innovative ways, for instance, to create new indicators based on existing ones, to select good indicators, to evolve optimal trading systems, and to complement other techniques such as fuzzy logic.

6. Hybrid Systems

There is now a growing realization in the intelligent system community that many complex problems require hybrid solutions. Increasingly, hybrid systems combining genetic algorithms, fuzzy logic, neural networks, and expert systems are proving their effectiveness in a wide variety of real-world problems.

Every intelligent technique has particular properties (e.g., ability to learn or explanation of decisions) that make it suitable for particular problems and not for others. For example, although neural networks are good at recognizing patterns, they are not good at explaining how they reach their decisions. Fuzzy logic systems, which can reason with imprecise information, are good at explaining their decisions but they cannot automatically acquire the rules they use to make those decisions. These limitations have been a key driving force behind the creation of intelligent hybrid systems where two or more techniques are combined in a manner that overcomes the limitations of individual techniques.

Hybrid systems aim to combine the power of each technique to overcome complicated problems. For instance, if there is a complex application, which has two distinct subproblems, say a signal processing task and a serial reasoning task, then a neural network plus an expert system can be used to solve these separate tasks. It is therefore very important to put forward different views and approaches for hybrid systems and to identify the possible problems to be faced in bringing the systems together [20].

Due to the power of combining the advantages of each individual technique, the use of intelligent hybrid systems, such as neurofuzzy and neurogenetic systems, is growing rapidly. There are already a number of successful applications in many areas including process control, engineering design, financial trading, credit evaluation, medical diagnosis, and cognitive simulation.

B. Hardware Aspect—Information Retrieval

1. Sensors, Intelligent Sensing, and Perception

A sensor is an entity that responds to a physical stimulus and transmits the resulting impulse [21]. Sensor technology concerns measurement (input) and

information processing (perception algorithms). Sensors can therefore provide an intelligent system with useful information from internal and external worlds. The information may include visual signals, tactile signals, force, torque, position, velocity, temperature, vibration, smell, taste, and pressure.

Perception compares observations with expectations derived from an internal world model. Sensory processing algorithms then may compute distance, shape, orientation, surface characteristics, physical and dynamical attributes of objects, and regions of space.

Because every type of sensor has specific areas of operation and certain failure modes, a single sensor can only provide partial information and is therefore often limited in its ability to resolve ambiguities and detect errors or failure. To compensate for this limitation, different sensors should be selected and integrated for achieving the specific objectives.

One of the key factors in making manufacturing systems intelligent is to sense and abstract suitable information from an unknown or dynamically changing environment. The type of environment that might be encountered and its representation should direct the choice of sensors. Basically the sensory information can be categorized as sight, hearing, touch, taste, and smell.

Charge-coupled device (CCD) cameras, sonar sensors, optoelectronic sensors, optical amplitude detectors, triangulation sensors, and lidar (light detection and ranging) sensors all can provide the sense of sight for machines. Machines do not have to gather the same data that we do, but they do need to be able to come to some similar conclusions. Actually, what can be seen and how it is seen are only a part of sight. Processing and interpretation are what make vision so useful [22].

Touch sensors include not only tactile but also pressure and temperature sensors. Microswitches provide a simple binary output to indicate the presence or absence of an object. Force/torque sensors are often incorporated to determine the force/torque applied at the points of interest. Tactile array sensors generate images for high-level interpretation, such as shape recognition.

The sense of taste provides a nonvisual and nonauditory means of liquid or solid substance identification. This may include chemical analysis of various materials and liquids to provide a breakdown of the contents.

In terms of the sense of smell, some electronic "noses" have been created. Essentially, the electronic noses are dumb versions of people's noses. The polymers act as spongy sensors, absorbing scent vapors and matching them with models contained in computer programs. A machine equipped with smoke and gas leak detectors can sniff trouble before it strikes. Many current electronic noses incorporate a specific sensor designed for a particular task. The aim of the next-generation electronic nose is to identify many smells, by relying on many broadly tuned sensors. Basically, the development approach is to mimic some aspects of olfaction, or scent reception, in humans or animals. In the future, the output from sensor arrays could be analyzed with artificial intelligence, such as neural networks, thereby allowing an array to learn different smells on its own. That would improve this electronic nose's versatility. Over time, electronic nose technology may evolve toward nature's overall solution: a combination of specifically and broadly tuned odor sensors.

The function of hearing is to provide a communication interface that identifies the direction and location of objects, as well as audio signals from speech.

Moreover, a new type of technology is emerging, the so-called biochemical sensors or bio-sensors. In effect, these bio-sensors are created by duplicating the materials in a living system, which can be an animal or a human being. The aim is to sense any specific things in its environment. In most cases, this involves chemical reactions. However, it will probably be some time before this technology is widely useful and/or available [21].

2. Sensor Integration and Information Fusion

Since having successfully developed sensors for sight, sound, touch, smell, and taste long ago, the technology world is now turning to how to smartly integrate them for various purposes. Sensor integration and information fusion involve the sophisticated software network that utilizes the combination of outputs of multiple sensors to provide more information to the system than could be realized from the inputs of the individual sensors themselves.

No matter how good a sensor is and how efficient the signal-processing algorithms are, uncertainty in sensor data cannot yet be totally eliminated. The use of multiple and different sensors can help to overcome the shortcomings and limits of each individual device. The integration of multiple sensor data can reduce uncertainty if an individual sensor fails [23]. Besides, to generate an accurate description of the environment requires the integration of the data from a number of sensors. However, many important issues are involved in multisensor systems, such as establishing a common framework to include different sensors, setting up communications between the various sensors, and planning strategies for sensor integration.

The number of different types of sensors intelligent systems are able to possess is not infinite but is severely limited. This also limits the type of communication an intelligent system can possibly receive and thus limits the information and the knowledge that the system can have about its environment. The selection of sensors depends on a combination of what needs to be controlled, the physical limitations of what can be sensed, and the particular control strategy to be used.

Typical methods for integrating data from multiple sensors can be categorized as the qualitative approach, the quantitative approach, and the mixed approach. The qualitative approach uses rule bases, such as expert systems, to describe and interpret sensor information. However, this approach cannot handle numbers easily and does not incorporate uncertainty with any exactness. The quantitative approach requires numbers. The information from sensors is quantified and mathematically modeled, so that statistical methods can be adopted.

Traditional sensors, such as proximity sensors, can perform adequately when correctly set up. However, they do not tolerate even small changes in their environment. To construct intelligent systems, intelligent sensors that can apply inherent knowledge of the objects to adapt to changes are needed. To develop intelligent sensors, both hardware and processing algorithms and their integration will remain important and active research fields for many years to come. An intelligent sensor, which contains built-in processing, calibration, self-test, diagnosis, and communication capabilities is and will still be a highly active research topic.

IV. APPLICATION OF INTELLIGENT SYSTEMS TECHNIQUES IN MANUFACTURING—DEVELOPMENT OF AN INTELLIGENT WORKCELL ROBOT SYSTEM

In batch production, workcell setting is subject to changes. These changes may be of a deliberate nature as in the introduction of a new product or of a forced nature as in machine breakdowns or material shortages. Whatever the reason, changes are inevitable in manufacturing industries [9]. These changes, which include changeover, accurate calibration, and re-setups, ultimately lead to low machine utilization. Therefore, for a small batch of products to be made economically, manufacturing systems should be capable of automatically adapting online to variations.

To deal with the workcell changes efficiently, robots have been integrated with computer-assisted design (CAD)/computer-assisted manufacturing packages, partially manual offline inspection, machines, sensors, bar codes, pallets, and accurate parts feeders over time to achieve better automation. However, there still is a deficiency in flexibility and intelligence for robots to automatically adapt to workcell changes.

Therefore, with the preceding issues in mind, an intelligent robot that provides a solution to overcome the difficulties caused by all these workcell changes, which exist in many current manufacturing systems, was developed at the Centre for Advanced Manufacturing Research, University of South Australia. The system consists of

- a CAD simulation package for workcell background knowledge generation (world models);
- a machine vision system for object recognition and location determination;
- a three-dimensional (3D) force/torque (F/T) sensor for monitoring and controlling the interaction between the robot and its environment;
- a six-axis industrial robot;
- a common database; and
- neural networks and C programs for proper decision making and performing of tasks.

This intelligent robot was enabled to automatically adjust itself to handle dynamic workcell changes that include malfunctioning machines, newly introduced or removed machines, and product changes. With such capabilities, a significant amount of time in system installation, re-setup, calibration, and programming can be saved. Experiments have been carried out and the results are very promising.

A. Introduction

The future manufacturing system is expected to possess much greater flexibility than such systems here now. Today if a machine is broken down or a machine is introduced into or moved out of a workcell, the workcell has to be halted until setup, installation, accurate calibration procedures, and programming have all been completed. Moreover, in any integrated manufacturing environment, both mechanical and electronic systems actually drift with time for reasons that are often indeterminable. Obviously, it is bothersome and time-consuming to perform accurate calibration all the time for various products or different layouts.

Clearly, an intelligent system capable of handling dynamic workcell changes automatically is important for the future of manufacturing. The challenges and major difficulties of developing such a system are

- How to identify the necessary subsystems.
- How to design and develop the subsystems.
- How to embed intelligence in the system.
- How to integrate and coordinate subsystems properly to achieve the maximum system flexibility and intelligence.

This research targeted the development of an intelligent workcell robot. The CAD-based simulation was developed to provide the background knowledge, robot workcell simulation data for the robot to compare with the physical world information sensed through machine vision, and a F/T sensor for proper actions. The machine vision was developed to realize object recognition and location determination. Neural networks were adopted and trained together with the 3D F/T sensor designed and built in this research to monitor and control the interaction between the robot and its environment. A common database was constructed to store all the relevant subsystem and system information.

With the achievement of this system, accurate calibration procedures for the workcell robot to perform tasks are no longer necessary. The robot workcell performance will not be halted among machines that are running whenever other machines are faulty, in maintenance, introduced, or removed. Thus this robot system can save a significant amount of time in installation, re-setup, calibration, and programming whenever there are workcell changes. This system then culminates in high machine utilization, especially for small-sized batch production manufacturers. Disturbances such as machine failures or parts shortages will not cause large delays in delivery any more. Most importantly, a company does not need to be committed to a single product strategy but can alter its path to meet competitive pressures.

B. System Hardware Overview

The hardware configuration [24, 25] of the present intelligent workcell robot system is shown in Fig. 2. This intelligent robot system consists of the following hardware features:

3D Force/Torque Sensor. Sensing is crucial to a supervised intelligent robot's ability to interact with the environment. How well this function can be performed depends on the types of sensors and receivers that the robot has. In the robot presented here, a neural network-based F/T sensor was integrated [26]. It provides 3D F/T information for robot control while the robot is interacting with the environment, such as positioning or assembling components.

Multifunction Input/Output (I/O) Board and Processing Circuits. The Lab-PC+ is a multifunctional, analog, digital, and timing I/O board for IBM personal computers (PCs) and compatibles. It is useful for signal analysis, data logging, temperature measurement, and direct current voltage measurement. The processing circuits designed were used to process the sensed information. The signal

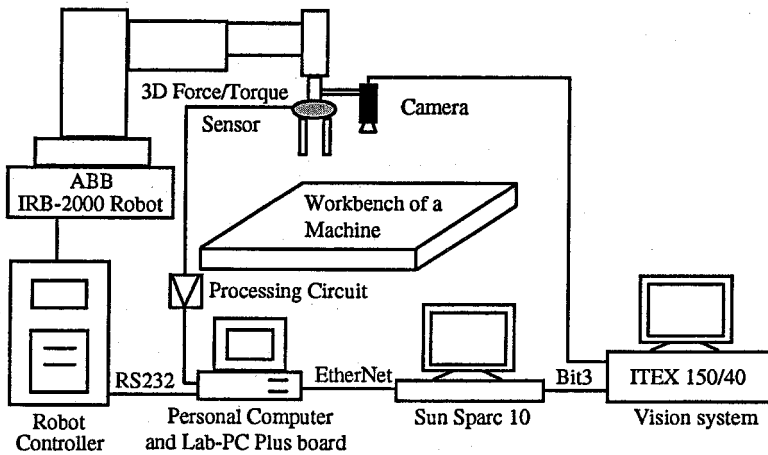


FIGURE 2 Physical layout of the intelligent workcell robot system.

from the 3D F/T sensor was amplified, converted into current for long distance transmission, transmitted, converted back to voltage, digitized, and then received by the Lab-PC+ installed in a PC. The information was then processed by neural networks or C programs for robot control output.

Machine Vision System. An ITEX 150/40 vision system was adopted. It includes a monochrome camera with a monofocal lens, an IMA-150/40 advanced image manager with 4 megabytes of reconfigurable memory, an AM-VS variable scan acquisition module, a DM-PC pseudocolor display module, a CM-PA programmable accelerator, and a BIT3 model 466 Sbus-VME adapter. The camera was attached on the robot final axis as shown in Fig. 2. It can then pan on tilt freely as the robot twisted its wrist.

Robot System. A six-axis industrial robot, ABB IRB-2000, with a model M92 controller was employed. Its loading capacity is 10 kg and the speed of its movements can be assigned and changed for various applications. This robot arm was therefore controllable at the F/T level with the implementation of the 3D F/T sensor at the wrist.

Host Computers. There are two different host computer platforms. The first is a Sun Sparc 10 workstation that provides multitasking environments for running the workcell simulation and vision system, for simulating the machine controller, and for coding C programs. The second platform is a 486 PC that provides the environments for writing C programs, running the multifunction I/O board, and executing control programs that act as the system coordinator and controller.

Interface among Hardware Subsystems. The mutual communications among subsystems are necessary and very important for system integration and coordination. The internal communications among subsystems are categorized as follows:

- Personal computer and Sun Sparc 10: Ethernet was established for the link between the PC and the Sun workstation.

- Personal computer and robot controller: An RS232 serial line linked the PC and the robot. It transmitted commands issued from robot host programs to the robot controller. The C callable library CommTools [27] was adopted and further developed.
- Sun Sparc 10 and vision system: The BIT3 Sbus-VME adapter was the link between the Sun Sparc 10 station and the vision system. It is a memory-mapped device that interconnects two buses. It allows data, peripherals, and bus resources to be shared between computer systems.
- 3D F/T sensor and personal computer: An RS232 serial line and the multifunction I/O board, Lab-PC+, provided the interface for C programs to access the sensed signals.

C. System Software Overview

An important part of any computational architecture is the software that runs on the hardware platforms. In this system, the software provides system modeling, planning, sensing, and acting. The architecture of the system's functional modules is shown in Fig. 3.

The goals of the software modules were to provide

- a flexible and friendly environment for coding, debugging, and running programs;
- a friendly simulation environment for generating necessary background data for proper system performance; and
- different library modules to be combined for various applications.

This intelligent workcell robot consists of the following software features:

CAD Package. A commercial CAD package, Advanced Solid Modeler (ASM [28]), was further developed to simulate camera and robot workcell operations [29]. It can thus provide the background knowledge and simulation results for the robot to compare with the physical world information for proper actions. The camera viewing parameters were set and tested for reflecting the way a real camera operates. These parameters helped ASM to output two-dimensional (2D) perspective gray-scale images that are highly similar to those from a real camera. These images can then be processed by the vision system module to generate useful data for workpiece recognition, such as image codes, before system online running takes place. The ABB IRB 2000 robot workcell was modeled as shown in Fig. 4.

The robot's collision-free moving paths for safely entering machines were simulated and generated in this simulation environment. The model-based knowledge was stored in the common database. Afterwards, this knowledge was used in object recognition and guiding the robot to safely move among machines.

Image—Processing System for Object Recognition and Position Orientation Determination. This is an image—processing, analysis, and recognition system developed under the Sun Sparc 10 platform in C programming language. It processed images from both ASM and the real world camera to generate image codes. By comparing these image codes, the right workpiece that was randomly located on the workbench can be recognized. This system can also derive the

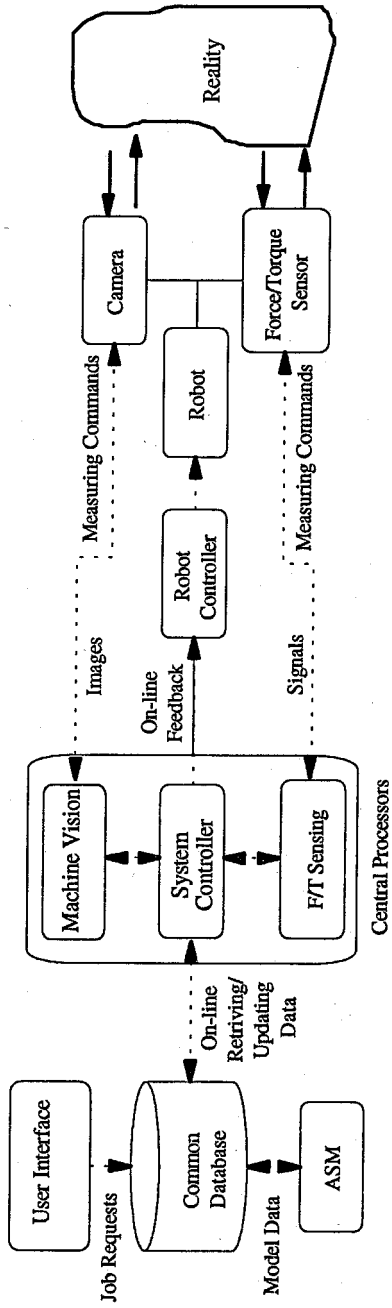


FIGURE 3 System software functional modules.

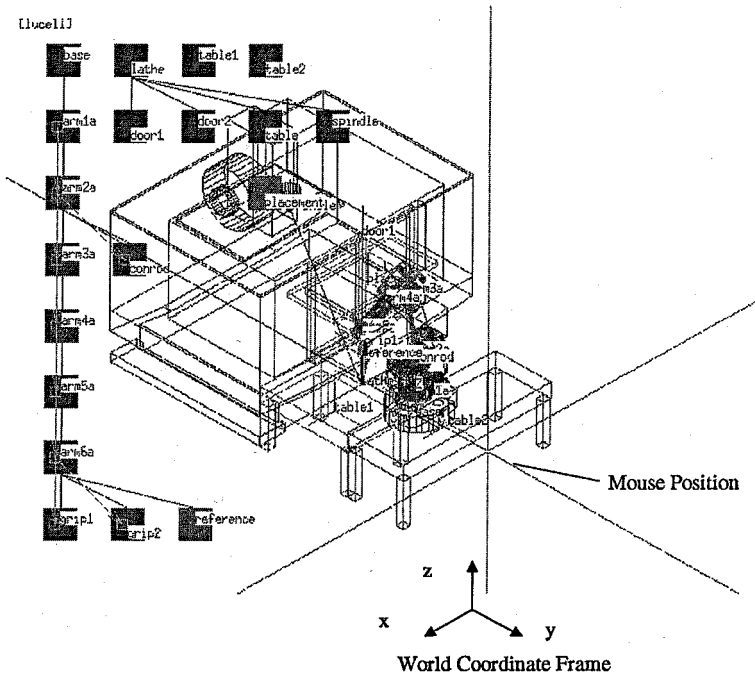


FIGURE 4 Robot workcell model.

workpiece position and orientation from images, thus allowing the robot to adjust its position and orientation to pick up the right object and send it to the requested destination. The processing flow chart of this image system is shown in Fig. 5.

This is a workpiece recognition and location determination module. It recognizes a 3D workpiece by analyzing 2D perspective images and also derives their 3D coordinates. For the robot to know the location of the workpiece in the world coordinate system, coordinate transfer is necessary. The coordinate systems are shown in Fig. 6 and the meanings of the terminologies are explained below:

Image zero point (zp): This point is the zero position in the camera image.

Gripper (g): It means the reference point on the top of the gripper.

Camera (c): The origin is located at the center of the lens.

$\vec{r}_{g \rightarrow zp}$: The vector from gripper reference point to image zero point.

$\vec{r}_{zp \rightarrow 0}$: The vector from image zero point to the center of gravity of the object located on workbench.

Therefore, the amount for the gripper to move to grip the object randomly located on the workbench is

$$\vec{r}_{g \rightarrow 0} = \vec{r}_{g \rightarrow zp} + \vec{r}_{zp \rightarrow 0} \quad (1)$$

Coordinate transfer that includes translation and rotation needs to be done for the robot gripper to properly grasp the workpiece. The orientation of an object is the angle between the x axis of the coordinate system and the axis of the least moment of inertia of an object in this system. The position value is the vector between the coordinate zero point and object's center of gravity.

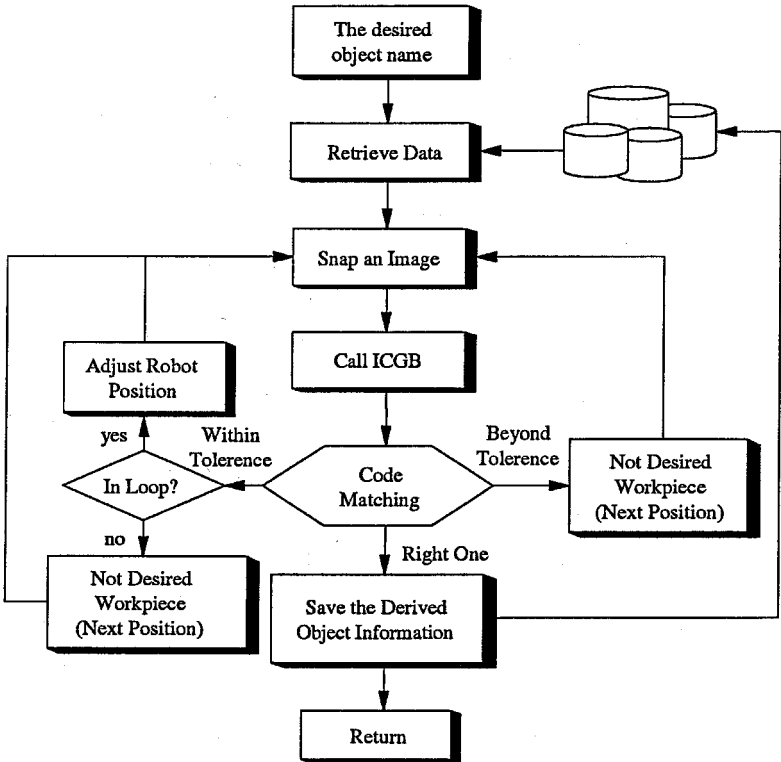


FIGURE 5 Image processing flow chart. ICGB, image code generation block.

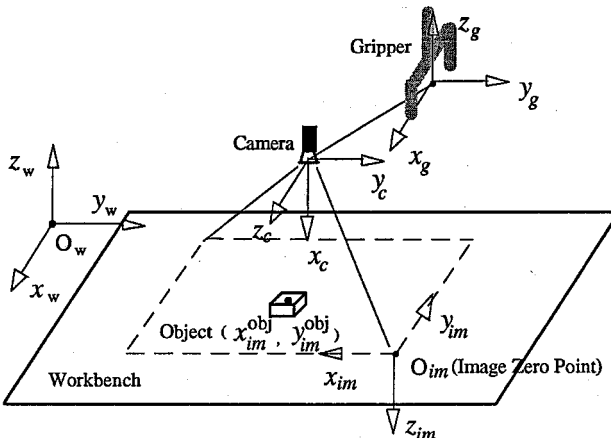


FIGURE 6 Coordinate system for robot and image system.

Communication among Subsystems. To enable all subsystems to function properly, communication and interface are extremely important. The software interfaces used in this system are as follows:

- **CommTools:** This provides a set of user callable functions that facilitate the transfer of data between the personal computer and the ABB robot controller. CommTools allows a user to write application programs in C to accomplish the serial (RS232C) data transfer.
- **PC-network file system (NFS):** This turns the PC into an NFS client that can access files on any NFS server on the network. It gives a user full access to the system's command set, file system, and resources.
- **Image-processing system and robot control:** A data file was used to exchange the message between the image-processing and robot control systems. The communication was between the Sun workstation and the PC via ethernet.
- **3D F/T sensor and robot control system:** Neural networks and C programs were developed for robot F/T control while the robot is interacting with environments.

Common Database. Image codes, workpiece information, and other relevant data are all housed in a common database. It is located under the Sun workstation and is available to all subsystems to access.

Robot and Vision Initialization Data. The data for initializing the robot and vision systems were located in the PC. Data include the robot identification number, communication port number, robot moving velocity, velocity percentage, coordinate calibration value, robot home position, and vision scanning area.

Force/Torque Sensing Module. When the robot had reached the destination with reference to the offline simulation, the F/T sensing module automatically took control to detect and adjust position errors if this function is assigned. It consists of neural networks and mathematics equations that have already been converted into C programs for monitoring the signals from sensors and sending the proper response to the acting programs. In this system, many various sensing functions have been developed to accommodate different interacting conditions.

Acting Programs. An acting program module coordinated the timing and working sequence of the image module, the F/T sensing module, and robot control. It also handled the system information passing, data extracting and updating, and system level decision-making. To obtain the simulated robot path, to transfer the path among different coordinate systems, and then to send the information to the robot controller for controlling robot movement were the responsibility of this module as well.

The information generated by the subsystems was used to let the robot know all the job requests and workcell status, such as which machine should be served, what workpiece should be searched and loaded, which machine is faulty, and where the new machine should be added in.

D. System Working Principle

The system performance can be divided into two categories: offline simulation and online running. The system offline simulation sequence is shown in Fig. 7 and the online running procedure is shown in Fig. 8.

The offline simulation and preparation had to be completed to provide the robot system with basic decision-making knowledge before the online system was executed. The offline tasks include designing products, deciding raw materials, simulating camera functions for generating images, and simulating robot movement for a collision-free path. The workcell simulation always needed to be done to generate enough information for the robot before workcell changes.

For online running, first of all, the robot read in the requested jobs from a job queue center collecting all the requested jobs. The robot then checked the machine status that requested the task. If the machine was down, the robot ignored this job request and read in the next one. If the machine was ready, the job was then executed.

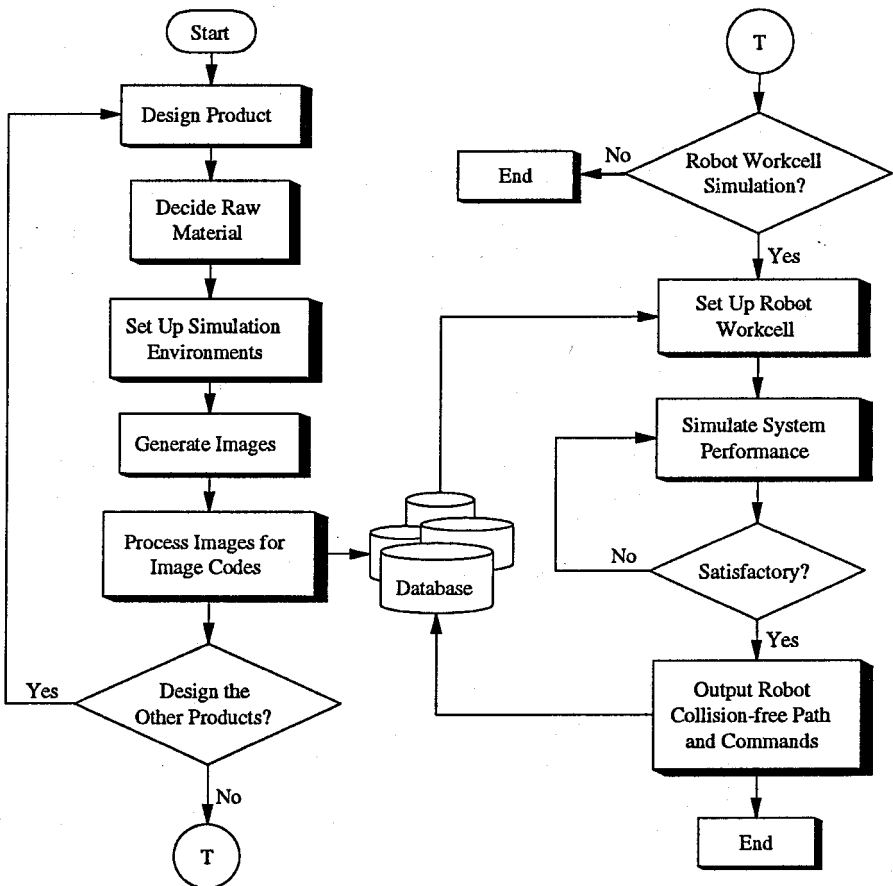


FIGURE 7 System offline simulation sequence.

The job requests were issued after offline simulation was completed. All the sequential jobs and the functions needed for completing the requested jobs are assigned in a data file. The robot then was actuated by the data file, which can be located on a remote site. The task, workpiece, F/T sensing function, and workbench were all assigned in the data file as well.

E. System Programming

Robots are widely used in manufacturing industries primarily for their flexibility and rapid response to changeovers. To make use of their advantages, a quick programming method is always needed. Today online programming is still the most common technique used to program industrial robots such as the use of teach pendants [30]. A major limitation of this method is that the actual robot is required to go through the desired sequence of actions. When the task is complicated or frequently changed or there are many robots, this method becomes very time-consuming.

In this present system, a task-oriented programming method was developed. It was designed to be user friendly, offline, hardware-independent, and implemented without the need of visual aids or special care in setting up spatial relationships between workcell components. Unlike many current programming systems, the additional visual and sensing functions were integrated in the intelligent workcell robot system presented here for online detecting and compensating for workcell uncertainties and dynamic changes. The uncertainties included modeling errors, location errors (position and orientation) of each workcell component, and robot relative position errors. The dynamic changes involved machine status, products, and workcell layout. The proposed programming system allowed programmers to deal with tasks rather than positions of the robot or its configurations as in the case of the graphics-level programming approach. The task was specified using the designed task-oriented commands such as 'load' and 'unload' in the user interface. The system then interpreted the programmed commands and determined the sequence of actions to be taken without human involvement. However, in its present form, only material-handling operations that are always performed by robots in a workcell were developed.

The developed task-oriented programming system can basically be divided into three levels, namely, task specification, hardware-independent program generation, and hardware-dependent program generation and execution as explained below.

I. Task Specification

The specifying of tasks using the developed task-oriented commands is the only stage that users needed to deal with in programming the system. All the data relevant to the task are then retrieved automatically from the common

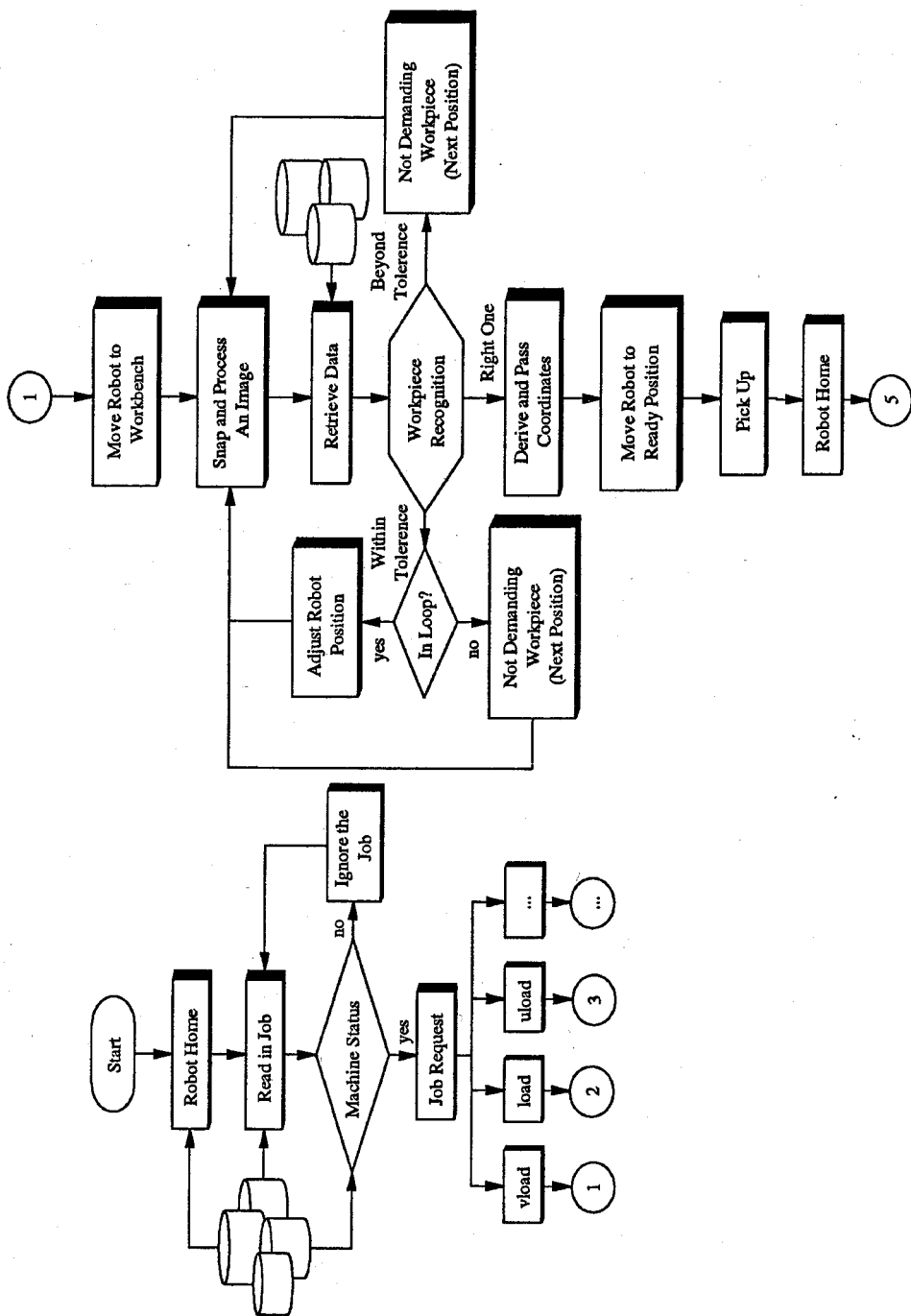


FIGURE 8 System online working procedure.

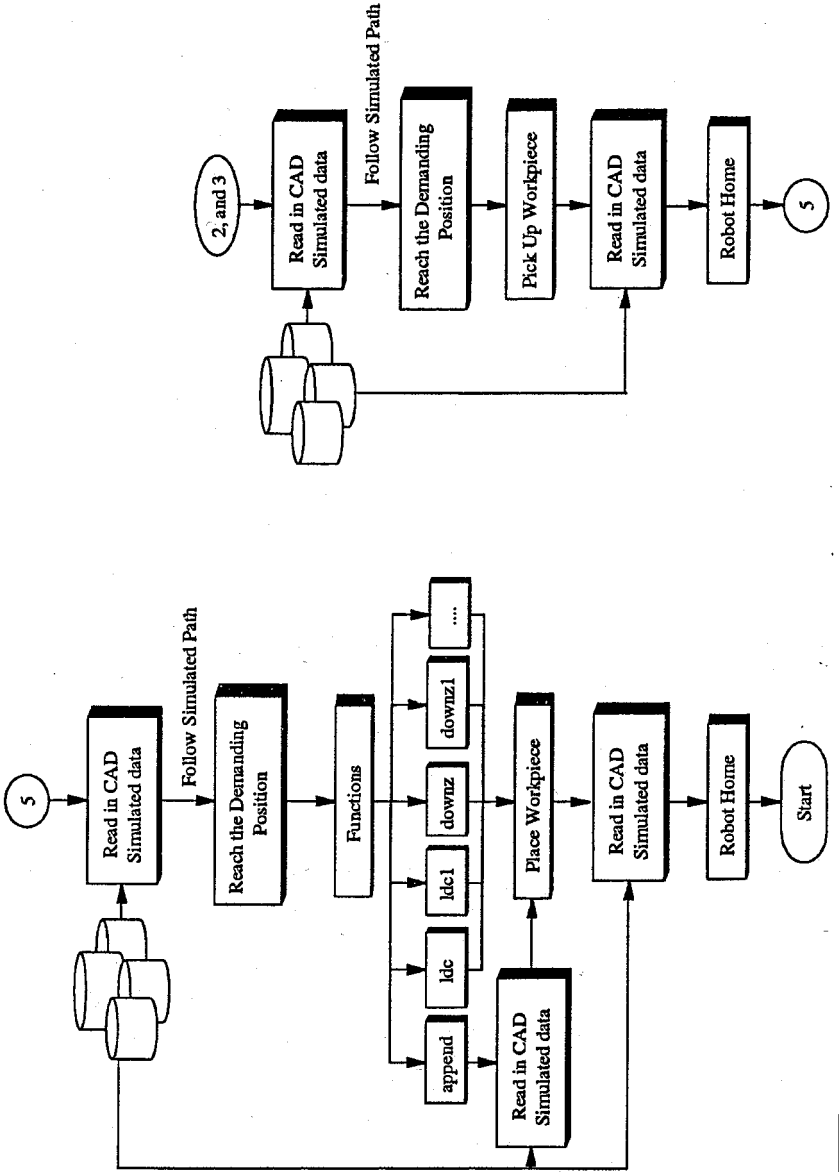


FIGURE 8 (Continued)

database after tasks are issued. The followings are examples of the developed commands.

Machine	Jobno	Task	Part	Place1	Place2	F/T function
<i>mc1</i>	<i>1</i>	<i>vload</i>	<i>alblk</i>	<i>bencha</i>	<i>vice</i>	<i>nxnyc1</i>
<i>mc2</i>	<i>2</i>	<i>unload</i>	<i>blk</i>	<i>storage1</i>	<i>chuck</i>	<i>downz</i>
<i>mcv01</i>	<i>3</i>	<i>load</i>	<i>albar</i>	<i>storage2</i>	<i>nil</i>	<i>append</i>
		<i>verify</i>	<i>nil</i>	<i>pk1</i>		<i>ldc1</i>
				<i>pk2</i>		<i>pxpyc</i>
				<i>nil</i>		<i>nil</i>

where

'machine' is the machine to be served.

'jobno' is the job number.

'task' is the task that the robot needs to perform.

'vload' is to use the machine vision to identify the part randomly located on the workbench and to derive its position and orientation for loading.

'unload' is to unload a part from a machine to a storage place.

'load' is to load a part from a fixed position to a machine.

'verify' is to use the machine vision system to calibrate the orientation of the appointed machine.

'part' is the name of the part to be processed.

'place1' is the position where the robot loads the part.

'place2' is the position where the robot unloads the parts.

'F/T function' is to appoint the force/torque sensing function for detecting and calibrating the environment.

'nxnyc1', 'downz', etc., are various functions developed for different environmental conditions to detect the way to approach the destination.

'append' is to instruct the robot system to employ the updated data to perform the task without using the sensing functions again.

To program the system, the developed commands needed to be issued in the correct sequence. To enable the system to detect the environment, compensate for the errors, and update the database, a specific sensing function needed to be assigned according to the environmental characteristics since workcell uncertainties existed. If the robot was requested, for example, to use the machine vision to identify the part 'alblk' from workbench A and loaded the part onto the 'vice' of the machine 'mc1', the intelligent robot system could be programmed in this way:

```
mc1 1 vload alblk bencha vice nxnyc1
```

Obviously, this programming system met its promises, which were to be offline, hardware-independent, user friendly, and implemented without the need of visual aids or special care in setting up spatial relationships between workcell facilities.

2. Hardware-Independent Program Generation

The system read the programmed tasks and interpreted them into hardware-independent programs containing subroutine calls for the robot, the machine vision module, and the force/torque sensor to perform the required operations.

The following is extracted from a hardware-independent program.

vload:

```

    rb_to_mc          /* inform the machine about the coming task and check
                       its status */
    app_loc           /* instruct the robot to approach the workbench */
    write_vjob        /* inform the vision module about the coming job */
    chk_job           /* receive the response from the vision module */
    cmpr_job          /* take action according to the vision feedback */
    read_verified_data /* read the calibrated machine orientation data */
    zero_sig          /* reset the force/torque sensor readings */
    app_mc_end        /* approach the machine vice */
    ft                /* sense the environment for positioning errors */
                    /* if 'nil' is issued, this function will not be performed */
    de_mc_end         /* reverse the app_mc_end */
    mc_go             /* inform the machine that robot has left */

```

3. Hardware-Dependent Program Generation and Execution

The hardware-independent subroutines were then further interpreted into hardware-dependent programs using the developed interpreter. The computer, connected to the robot controller, then followed the sequence of the output programs to coordinate the subsystems and performed the required task. If the simulation data describing the workcell were needed, the system controller would automatically retrieve the data from the common database.

The robot used the vision system to scan through the assigned workbench to identify and pick up the right demanded workpiece that was randomly located and load it to the right machine if 'vload' was issued. The robot moved to the requested workbench, snapped, and processed an image. Image codes for all workpieces in the snapped image were then generated and compared with the image codes associated with the demanded workpiece. If the right workpiece was potentially found, the position and orientation of the robot that is attached to the camera were adjusted. The workpiece would then be snapped in the center of the image to avoid serious distortion problems. The potential right workpiece means that the image code is different from that demanded, but within the predefined tolerance. When the right workpiece was identified, its position and orientation were derived and passed to the acting program module and then to the robot controller for properly picking up the workpiece. If the right workpiece was not found, the robot kept on searching by following the predefined searching path.

If 'load' was assigned, the robot went to the requested location, which was a fixed position, to pick up the demanded workpiece and sent it to the destination. The CAD-simulated path stored in the common database was extracted and followed.

If 'unload' was requested, the robot unloaded the workpiece from the machine that issued the request to the destination. The CAD-simulated path was extracted and followed again. After the CAD-simulated path had been followed and completed, the selected F/T function took control. These functions consist of the neural networks and C programs. They were implemented to automatically detect and compensate for the errors that existed between the CAD simulated results and the physical world settings to make sure that the workpiece was located properly.

The robot went back to home position after the requested task had been completed and then read in the next job. The path simulated in ASM must be ensured to be collision-free for both approaching and withdrawing from machines. All the functions were modularized while they were programmed. Hence, it is easy to modify the functions or add new functions.

The tasks could be issued from any computer text editor, which is the user interface of the present system. This programming method did not need visual aids or special care in setting up the spatial relationships of workcell facilities while one was programming this intelligent workcell robot system. This programming system could decompose the planned task to robot-independent programs and therefore it was not specific to any particular robot type. The system then interpreted the program into robot-dependent programs, which were executed by the computer sending instructions to the robot. The common database provided the robot moving paths and simulation results that were generated by the offline simulation environment in advance for approaching workcell facilities and understanding the environments.

F. System Online Performance

The following is one of the examples that were carried out in our laboratory. This example focused on the behavior of the system when new changes occurred in the workcell and aimed to illustrate the following system capabilities:

- searching and identifying the demanded workpiece that is randomly located on the workbench;
- adapting to any machine introduction without interfering with existing system performance;
- adapting to any machine breakdown or faulty condition without interfering with the existing system performance; and
- on-line detecting, compensating, and recording of errors that exist between the real—world setting and simulated model.

The task issued first is executed first in this system. A machine priority setting can be introduced to allow the robot to be interrupted and accessed by a machine that has higher priority. The assumption made for this example was that initially there was only one machine in the workcell. The following sequence was then designed and used to demonstrate the system capability.

A data file was implemented to issue job requests to the workcell robot. The job requests issued in this example are described below.

A job was issued for machine 1:

Job Command	⇒	use the vision system to load an aluminum block to machine 1 by ldcl;
Program	⇒	<i>mc1 1 vload alblk bencha vice ldcl.</i>

The vision system was functioning and coordinated with the robot to search for and identify the requested aluminum block and load it to machine 1. The position error that existed between the simulation model and the real-world setting was

then detected, compensated for and recorded. For example, 'ldc1' was designed for placing a block onto a corner and then recording the final approaching path. The recorded approaching path for compensating for simulation errors can then be used while the 'append' command is issued.

Machine 2 was prepared to be introduced to the workcell and its installation position and other necessary information were simulated. System data were updated for machine 2 and finally machine 2 was introduced. A new job request was issued for machine 1 and machine 2:

Job Command \Rightarrow pick block 1 from fixed position 1 to machine 2
by downz;
Program \Rightarrow *mc2 2 load blk pk1 chuck downz.*

The robot picked up block 1 from a fixed position and loaded it to machine 2:

Job Command \Rightarrow unload aluminum block from machine 1 to storage
place 1 by downz;
Program \Rightarrow *mc1 3 unload alblk storage1 vice downz.*

The robot unloaded the aluminum block that had been machined in machine 1 to storage place 1:

Job Command \Rightarrow unload block 1 from machine 2 to storage
place 2 by ldc;
Program \Rightarrow *mc2 4 unload blk storage2 chuck ldc*

The robot unloaded block 1 that had been machined in machine 2 to storage place 2:

Machine 1 was down and was reported to the designed 'Fault Report Centre' while the robot was serving machine 2 for unloading block 1;

Status Report \Rightarrow machine 1 is down.

The job was then issued for machine 1 again:

Job Command \Rightarrow use vision system to load aluminum block to machine 1
by ldc1;
Program \Rightarrow *mc1 5 vload alblk bencha vice ldc1.*

The robot ignored the last job request because machine 1 was reported to be down and read in the next job.

Machine 1 was fixed. Its status was updated to be ready in FRC. A job is issued for machine 1:

Job Command \Rightarrow use vision system to load aluminum block to machine 1
by ldc1.
Program \Rightarrow *mc1 6 vload alblk bencha vice ldc1.*

The robot then coordinated vision system to load aluminum block to machine 1.

If no F/T sensing function was selected and issued in the job request, then no F/T function would be executed for online error compensation.

The job requests were sequentially executed after the robot workcell was activated. In this example, new machine had been introduced, machine status had also been changed randomly and the workcell robot did successfully perform the requested tasks. The F/T sensor functions for online detecting, compensating, and recording errors between the real world setting and simulated model were also testified.

In this system, image codes and the related data were saved for object recognition purpose. To save image codes instead of images can result in saving computer memory space and the time for recognizing objects [29].

G. Discussion and Conclusion

In this development, the CAD simulation was built to provide the background knowledge of the workcell for the robot to compare with the physical world for proper actions. The vision system was developed to realize object recognition and location determination. Neural networks were adopted and integrated together with the 3D F/T sensor, which is designed and built in this research to control and monitor the interaction between the robot and its environment. A common database was constructed to store information. The integrated intelligent robot that can automatically accommodate the frequent workcell changes was thus constructed. Satisfactory results had been achieved in the experiments for different machines and storage places, such as using vision system to search and load workpiece, going to fixed positions to pick up or load workpiece and unloading the workpiece to the storage places, etc. The robot had also demonstrated its capabilities in dealing with the randomly changed machine status, such as introducing new machines, relocating machines, and handling breakdown machines.

Thus, with the present development, time-consuming accurate calibrations are no longer bothersome for proper system performance because the proposed system can automatically detect and compensate for workcell changes and simulation errors.

V. CONCLUSION

In the human mind, the stimulus-response mechanism is the basic mechanism that permits humans to learn. Fundamentally, an intelligent system (IS) should be a stimulus-response system. The stimulus is the sum of the communications entering through the senses (sensors). The brain (central processor) extracts information from this stimulus and represents it as a situation. Next, the IS selects a response rule (related response module) stored in the memory, appropriates it to the situation, and performs the response of this rule to permit the system to get nearer to the state of its objective. In the memory, the IS has accumulated response rules that it has generated from earlier experiences and from generalizations based on other response rules.

Obviously, development and integration of appropriate intelligent techniques, which include both software and hardware with manufacturing systems, will be some of the most important challenges that affect future manufacturing systems.

Exploration of the deeply rooted problems in today's manufacturing systems has begun. In the future, the system environment is envisaged to be more user friendly and integrated with the capability of the subsystems to be further advanced. Clearly it is now and will still be the trend to apply human, animal, or insect sensing, navigation, and control methods to manufacturing systems.

REFERENCES

1. Badiru, A. B. *Project Management in Manufacturing and High Technology Operations*, 2nd ed., Wiley, New York, 1996.
2. Tzafestas, S. G. *Intelligent Robotics Systems*. Dekker, New York, 1991.
3. Fritz, W. Intelligent systems and their societies, 1997. Available at <http://www.anice.net.ar/intsys/whaisint.htm>.
4. Lu, T. F. *The development of an intelligent workcell robot*. Ph.D. Thesis, 1997.
5. Hayashi, H. Intelligent manufacturing. *IEEE Spectrum* 30(9):82-84, 1993.
6. Jurgen, R. K. Global R&D for a 21st century manufacturing system. *IEEE Spectrum* 27(11): 122-124, 1990.
7. Solberg, J. J. and Kashyap, R. L. ERC research in intelligent manufacturing systems. *Proc IEEE* 81(1):25-41, 1993.
8. Bosnik, J. R., Hemati, N., Pourki, F., and Siegler, S. Robotics in intelligent manufacturing: IMPAQT activities. In *Proceedings of the 5th IEEE International Symposium on Intelligent Control*, 1990, Vol. 2, pp. 1205-1210.
9. Solberg, J. J. Engineering Research Centers: The Purdue Experience. In *Seventh IEEE/Components, Hybrids and Manufacturing Technology Society (CHMT) International Electronic Manufacturing Technology Symposium, Proceedings*, 1989, pp. 206A-206D.
10. Jamshidi, M. and Geng, Z. Learning control of robot manipulators. *Expert Syst. Applicat.* 4:297-304, 1992.
11. John, M. Organizational foundations of intelligent manufacturing systems—the holonic viewpoint. *J. Comput. Integrated Manuf. Syst.* 8(4):237-243, 1995.
12. Zhang, H. C. and Huang, S. H. Applications of neural networks in manufacturing: A state-of-the-art survey. *Int. J. Prod. Res.* 33(3):705-728, 1995.
13. Rembold, U., Lueth, T., and Ogasawara, T. Intelligent manufacturing components for future CIM systems. In *IFAC Intelligent Manufacturing Systems*, Vienna, Austria, 1994, pp. 1-10.
14. Bratko, I. and Urbancic, T. Transfer of control skill by machine learning. *Eng. Applicat. Artif. Intell.* 10(1):63-71, 1997.
15. Zhang, H. C. and Huang, S. H. Applications of neural networks in manufacturing: A state-of-the-art survey. *Int. J. Prod. Res.* 33(3):705-728, 1995.
16. Warwick, K., Irwin, G. W., and Hunt, K. J. *Neural Networks for Control and Systems*. Peter Peregrinus, London, 1992.
17. The Impact of Expert Systems. Available at <http://www.morebusiness.com/other/v2n8.html>.
18. Fuzzy Logic. Available at <http://www.tecquip.co.uk>.
19. NeuroGenetic Computing. NIBS Pte Ltd. Technical Report TR-960307. Available at <http://sunflower.singnet.com.sg/~midaz/Neuroga.htm>.
20. Medsker, L. R. *Hybrid Intelligent Systems*. Kluwer Academic, Boston/London/Dordrecht, 1995.
21. Chapman, P. W. *Smart Sensors, An Independent Learning Module from ISA*, 1996.
22. Rosheim, M. E. *Robot Evolution—The Development of Anthrobotics*. Wiley, New York, 1994.
23. Gray, K. O. and Caldwell, D. G. *Advanced Robotics and Intelligent Machines*. The Institution of Electrical Engineering, London, 1996.
24. Lin, G. C. I. and Lu, T.-F. Construction for an intelligent product assembly and inspection system. In *The 3rd International Conference on Automation Technology, AUTOMATION '94*, Taipei, Taiwan, ROC, 1994, Vol. 3, pp. 13-17.
25. Lin, G. C. I. and Lu, T.-F. CAD-based intelligent robot workcell. In *The 3rd International Conference on Computer Integrated Manufacturing (ICCIM)*, Singapore, 1995, pp. 437-444.
26. Lin, G. C. I. and Lu, T.-F. Neural network for active compliance mechanism calibration. In *International Conference on Data and Knowledge Systems for Manufacturing and Engineering*, Hong Kong, 1994, pp. 497-502.

27. ABB (Asea Brown Boveri). *CommTools User's Guide*. ABB Robotics, Sweden 1992.
28. *QikDraw Systems ASM Tutorial*. QikDraw, Adelaide, South Australia, 1994.
29. Lin, G. C. I., Lu, T.-F., and Zhang, D. CAD-based intelligent vision and robot system. In *IS&T/The International Society for Optical Engineering (SPIE) Symposium on Electronic Imaging: Science and Technology*, Vol. 2423, No. 14, 1995.
30. Shirinzadeh, B. and Tie, Y. Object-oriented task planning and programming system for layered robotic assembly and disassembly operations. In *International Conference on the Robots for Competitive Industries*, 1993, pp. 502-509.