A METHOD FOR IMPLEMENTING AI-BASED CONTROL IN A MANUFACTURING WORKCELL USING A BIOLOGICAL MODEL

by

Robert D. Borchelt, Ph.D.
Department of Industrial and Manufacturing Engineering
University of Wisconsin-Milwaukee
P.O. Box 784
Milwaukee, WI 53201
(414)229-5898
borchelt@convex.csd.uwm.edu

1. Abstract

Much attention has been given in recent years to the capabilities of artificial intelligence tools and their possible usefulness in the monitoring and control of manufacturing workcells. This paper presents a technique which allows the use of AI-based control approaches in manufacturing workcells without the necessity of "special" computers or exotic control systems. The technique advocated is explained using a biological model as an analogy.

2. Introduction

Control of manufacturing workcells involves the coordination and sequencing of complex tasks (among other things) in what is often a less than perfect domain. Strictly algorithmic approaches to control software have required rigid work environments with limited variability. Recognition of the limits of strictly algorithmic control methods has led to a desire to incorporate artificial intelligence into control systems.

A major problem facing industry engineers, when they attempt to use AI tools in a "real-world" cell, is that the AI tools are designed to operate in a personal computer (PC), workstation, or mainframe environment. The manufacturing system that they are designing is often controlled by machine controllers, programmable logic controller's, and/or perhaps a PC for data handling.

Machine controllers, PLC's, and even standard robot controllers are normally too limited to allow implementation of AI-based control. If the designers choose to give control of the cell over to a PC then they often feel that they must sacrifice speed and must "custom-design" interface hardware and software to allow the PC to control the cell.

Previous research in this area has focused primarily on a "single processor" model, where the software systems developed reside in a cell computer or other controller and utilize various methodologies to interact with the cell in a somewhat "real-time" manner. Often, AI portions of the control software are "off-line" and only called up when needed. The computing power needed to complete the AI tasks is commonly obtained by placing the cell in a "hold" status until normal operations can continue. While this is certainly acceptable in some situations, the necessity of "pausing" the system is a serious constraint which would limit the widespread application of AI in cell control. Even those systems which allow multi-tasking in the cell computer would experience serious degradation of response times if a significant AI task were taking place and tieing up system resources.

The obvious solution for this is to incorporate more than one processor in the control system. Research on distributed computing and parallel processing has shown the great power of multi-processor computing in solving problems requiring quick solutions or massive computation. One of the serious limitations that this research has pointed out is that problem solvers often don't know how to break their problem down into pieces which can be solved by separate processors. Even given a problem which can be decomposed, the task of deciding which processor gets what information when, and where each processor should send its results when it is finished is daunting.

3. A Model for Manufacturing Control

In an attempt to describe how multi-processor computing could be implemented to support control of manufacturing workcells, a biological analogy based on a simplified model of the human brain will be used. Imagine a manufacturing cell with a cell computer (for this discussion a PC), a programmable logic controller (PLC), and a robot controller, along with sensors and actuators necessary to complete a manufacturing task.

The PC could be considered as the cerebral cortex of the system, the PLC as the brain stem, and the robot controller as the specialized portions of the brain dealing with motor functions. The sensors would be similar to the sensory systems of the human body, and the actuators would be similar to the involuntary muscles and other lower-level systems of the human anatomy (i.e. those systems not requiring fine or gross motor skills).

Assuming such a model, the tasks which should be assigned to each processor can be discerned (part of the reason for using the model) based on the perceived level at which the human brain would handle a similar task.

Using this model, the PC should be responsible for controlling "higher level" problems such as assembly planning, error recovery, and other "knowledge intensive" tasks. It is a more sophisticated "thinker" than the other processors, and can support AI-based tools to handle these complex decisions. Because it is handling larger "problems" it will respond slower and should not be "interrupted" by "housekeeping details".

The PLC, on the other hand, should be responsible for handling "instinctual" or involuntary reactions like the implementation, and subsequent feedthrough of, an emergency stop. Another example might be the timing of a pallet transfer. The sequence of operations for a pallet transfer are well defined, simple, and easily accomplished by a PLC. In addition, the task of transferring the pallet from a main conveyor into a transfer station is not one that can be closely controlled by an automated system. The success and/or failure of a pallet transfer is much more dependent on the physical dimesions and tolerances of the material handling components than on the instructions sent to it buy the controller. If a problem develops in the transfer, there is not likely to be much that an "intelligent" controller can do about it. These "housekeeping details" are time consuming but predictable and do not require the higher level "reasoning" ability of the PC. Delegating these tasks to the PLC frees up the PC for its higher level duties. The parallel nature and speed with which a PLC can detect a situation and implement a response make it ideal for time-critical tasks. The "parallel" nature also makes a PLC well suited for handling the many details which must be simultaneously considered during normal operation.

Finally, the robot controller should be used to focus on the tasks for which it was originally designed. Namely, controlling the actions of the mechanical arm. While robot controllers often contain user definable inputs and outputs which can be used to sequence operations within a robot-centered cell, they are not well suited for the role of cell controller. Sequencing tasks slow down operation of the robot and the ability of standard robot languages to express sequencing concepts is limited at best. If task planning, gripper control, sensor monitoring, etc., is all given over to the other processors in the system, then the robot controller can concentrate on quick and accurate responses to movement commands. Writing robot programs in such an environment is greatly simplified as well.

Any specialized controller could be incorporated in a similar fashion. If a machine vision system is part of the sensor array, then the vision system computer should be focused on processing the video image and providing a simplified abstraction. The guiding principles are found in how the brain handles similar tasks (or at least how we think it does). The analogy is general enough to be applicable in most manufacturing control problems, but can be applied specifically enough to be useful.

4. Conclusions

Existing research often focuses on improving robot languages or robot controllers in order to implement AI-based control schemes in robotic workcells. Most robot purchasers are industrial companies which are unwilling to pay for the "added capabilities" which would be part of such an "improved" robot system. Whether this occurs due to ignorance of the potential advantages of such a system, or due to an actual lack of the need for these capabilities, is beside the point. The fact is that robot manufacturers will not offer these "features" until the customers start demanding them.

Having generalized about "most" robot purchasers, it is now necessary to address the aggressive and forward-thinking minority. Some companies <u>are</u> exploring AI-based control, and they have to do it with equipment manufactured today. It is possible to do so, and one approach has been explained here. The important concepts are not found in the specific discussion of which type of processor and/or device is given a certain task. The capabilities of each "robot controller" and "PLC" varies from one vendor to the next. Rather, the important concepts are found in the division of tasks among cooperating processors, with information being available where needed. The tradeoffs between speed of decision-making and complexity of the decision are important. It is not necessary for the "AI" portion of the control system to consider all decisions. On the other hand, it <u>is</u> necessary for the AI portion to understand the effects of those decisions and be able to notice when they cause errors. The AI portion of a controller should be able to over-ride the "lower-level" decision maker if it deems that to be necessary, just like the brain can make you "hold your breath" even though you need air. The technologies involved will vary from system to system, but the idea of structuring the levels of decision-making according to a biological model seems both feasible and easily understandable.

The resulting control system should be both powerful and fast, and would take advantage of the strengths of each of the controller platforms involved. The response times of the system would be appropriate to the tasks encountered rather than "uniformly fast" but required to be simple, or "uniformly slow" but "thoroughly considered". Overall, this model should provide a guideline for implementing multiple-processor computing in the manufacturing control environment.