PERIODICA POLYTECHNICA SER. MECH. ENG. VOL. 42, NO. 2, PP. 153-168 (1998)

AN ON-LINE INTELLIGENT ADAPTIVE CONTROLLER FOR MANUFACTURING OPERATIONS BASED ON AN OPEN ARCHITECTURE

Oscar HERNÁNDEZ

Technical University of Budapest
H-1521 Budapest, Hungary

Received: October 28, 1998

Abstract

In the development of the unattended and self-adjust machining system, where the human operator must be replaced by a Computer Numerical Control, the control system should properly process information with its varying environment in an intelligent way. The intelligent CNC must be able to recognize, in real time, major problems of operation connected to the machining process like chipping problems or tool breakage. To do that, the CNC intelligent system must be able to receive, process, and analyze inputs from multiple types of external sensor attached to it. This approach can only be possible on the base of an Open Architecture.

The author of this paper proposes the use of fuzzy logic to develop an intelligent adaptive controller for manufacturing operations based on an open architecture that must be able to face problems of control, monitoring and fault diagnosis in real time.

Keywords: Open Architecture Controller, automated manufacturing operations, monitoring, artificial intelligence, hard real time systems.

1. Introduction

Most of NC and CNC's installed today are for Original Equipment Manufacturer – OEM – application that means a lack of flexibility because they are offered as a complete and unique package for a particular need where the user has hardly a possibility to integrate his own software solutions [1]. They do not offer any possibility to low level control access and they are closed solutions. In addition, there are difficulties associated with using proprietary technologies such as vendor-dictated pricing structures, non-common interfaces, higher integration costs and the requirement of specific training for troubleshooting and operation [2].

This lack of flexibility has led to develop an open architecture controller for manufacturing operations.

The three top car makers in the U.S. (Chrysler, Ford and General Motors) announced they would cooperate in Open Modular Architecture for Control project (OMAC) in the definition and implementation of open control technologies for manufacturing applications [3].

Open System Architecture for Control within Automation Systems (OSACA) in Europe has declared its intention to specify an architecture for open control systems which is manufacturer independent. In addition, this project will improve the competitiveness of the manufacturers of machine tools and control systems [4].

Finally, six major manufacturers in Japan (which are three machine tool builders: Toshiba Machine Co., Toyoda Machine Works, Ltd. and Yamazaki Mazak Corp.; IBM Japan, Ltd.; Mitsubishi Electric Corp., and SML Corp.) have formed the Open Systems Environment for Control (OSEC), declaring they would jointly develop a PC-based open computer numerical controller for manufacturing operations [5].

The OMAC, OSACA and OSEC projects, all of them, were created to face the problem of vendor dependence for automated manufacturing applications with the same goal: to define standards for an open and modular architecture that must be vendor independence, scaleable and easy to maintain. Each project has its particular limitations but all of them are oriented for manufacturing applications in high-automated factories. Even when they are PC-based solutions they are oriented to work in a high-coupled network environment making its implementation more difficult for the high technology involved.

Despite to these orientations (high-automated and high-coupled network environment), we have identified that small shop floors are outside of their scope for the high cost involved. They do not include solutions either for low automated factories or small and manual operated shop floors (with high possibility of retrofit).

Based on this, we propose an on-line Intelligent Adaptive Controller (IAC) for manufacturing operations based on an Open Architecture for small and low automated shop floors where the budget of investment is not too much but competitiveness needs are high. Our proposal has been designed to be applied to high-automated shop floors as well.

2. Intelligent Adaptive Controller for Manufacturing Operations, its Objective

The main goal of our proposal is the definition of a PC-based Intelligent Adaptive Controller for manufacturing operations based on an Open Architecture. The Intelligent Adaptive Controller must be able to monitor and control automated manufacturing processes on line executed by machine tools or robots as well. The intelligent adaptive controller is able to detect any abnormal state of operation of the manufacturing process and it is able to adjust the operating parameters in order to protect the manufacturing system against catastrophic failure [6].

The open architecture supports the more effective integration of user-defined special purpose modules. Other applications using AI techniques can also be easily added to the proposed open architecture by simple integration of new modules, such as ANN, expert systems, vision, etc.

It is important to remark that our proposal does not define an open architecture for CNC systems. Instead of it, it uses a particular CNC (based on a selected open architecture) adding new feature and new functions all of them controlled by a computer.

3. Architecture of IAC

Our Intelligent Adaptive Controller has a centralized controller architecture (due to its PC-based proposal), but can be improved, including the necessary modules and hardware interface, to work in a network environment.

The modularity of the system is achieved by three basic modules, they are: Acquisition, Processing and Control (see *Fig. 1*). Note that each main module integrates atomic and functional sub-modules. For example, acquisition force and force processing, acquisition speed and speed processing, acquisition of current and current processing.

Depending on particular needs of implementations different types of sensors can be used [7]. For example: force sensors, acoustic emission sensors, current sensors, etc. Simple sensors such as proximity switches are generally categorized in the Event Collector element.

Each processing module depends on a specific need and different features that must be analyzed, such as statistic features or time domain analysis.

The integration of a new feature must appear in pair with its respective processing module, for example: acquisition of Acoustic Emission signal and frequency domain processing of these signals in order to have a tool wear diagnostic.

The control module includes two sub-modules, one of them for monitoring actions (supervisory module) and the other for decision making (intelligent module) process. If the supervisory module detects an abnormal situation the intelligent module takes place modifying the appropriate operating parameter(s) of the manufacturing system.

The Complex Machine can be any machine tool with at least two degrees of freedom. It includes the possibility to use robots that perform manufacturing operations.

The communication between acquisition, processing and control modules is carried out through main memory.

The intelligent adaptive controller has been developed in the base of a fuzzy logic system. The author of this work has included an interpretation module to the basic modular structure of the fuzzy logic system, see *Fig. 2*. The module added, uses the output of the fuzzy system in order to

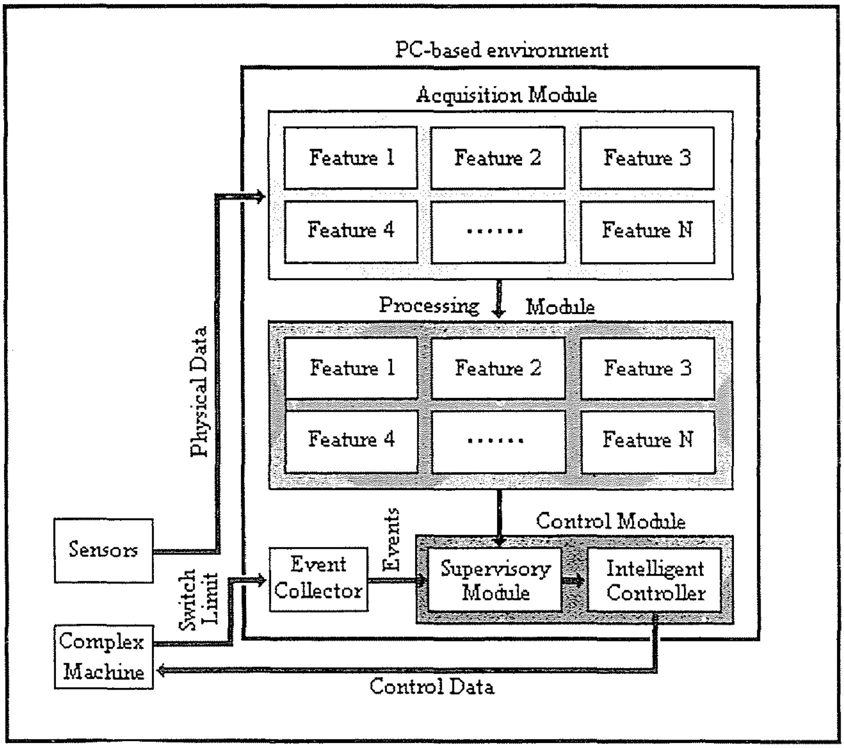


Fig. 1. General reference model defining a PC-based Intelligent Adaptive Controller for manufacturing operations based on an Open Architecture

execute any control action that must modify the manufacturing parameters contained in the Data Base module.

The parallel execution of tasks (acquisition, processing and control) is necessary in order to meet the time constraints of any hard real time system. A PC-based real time operating system has been selected to manage and control the parallel execution of multiple tasks.

4. Implementation

We have identified an important area of research where robots are used for machining parts (see *Fig. 3*), to produce finish surface or in assemble activities. The use of Artificial Intelligence techniques can be used to improve the performance, using monitoring and diagnostic systems for this purpose [8], [9].

The author of this paper proposes the development of an intelligent

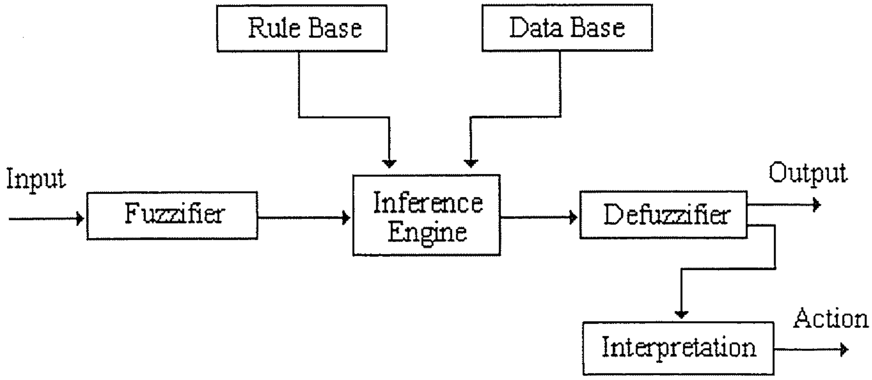


Fig. 2. Improved fuzzy logic structure. An Interpretation module has been added, it translates the outputs, from the fuzzy logic system, to control actions

adaptive controller for robotised manufacturing operations based on an Open Architecture (see *Fig. 4*), in order to build an intelligent machining system [10]. The intelligent Adaptive Controller must be able to recognize tool wear condition, tool breakage and abnormal states of operation on-line. The use of Fuzzy Logic (our suggestion) can improve the final results of manufacturing operations and can add some features of intelligence for applications that require decision making in real time. Fuzzy Logic can be used to produce some diagnostics about the machining process and tool wear condition as well.

The main modules of the open architecture developed in this work are:

- Data acquisition
- Signal processing
- Fuzzy logic
- Decision making
- Human interface (Display)
- Analysis of performance

In the development of these robotised manufacturing operations we have used different hardware as well as software elements. The elements included in this hardware platform (*Figs. 3* and *4*) and their purposes are:

- An SCARA robot with 4 degrees of freedom to perform manufacturing operations.
- An industrial PC: used to run the Real Time Operating System under DOS environment, connecting the robotised manufacturing operation with monitoring and control tasks.

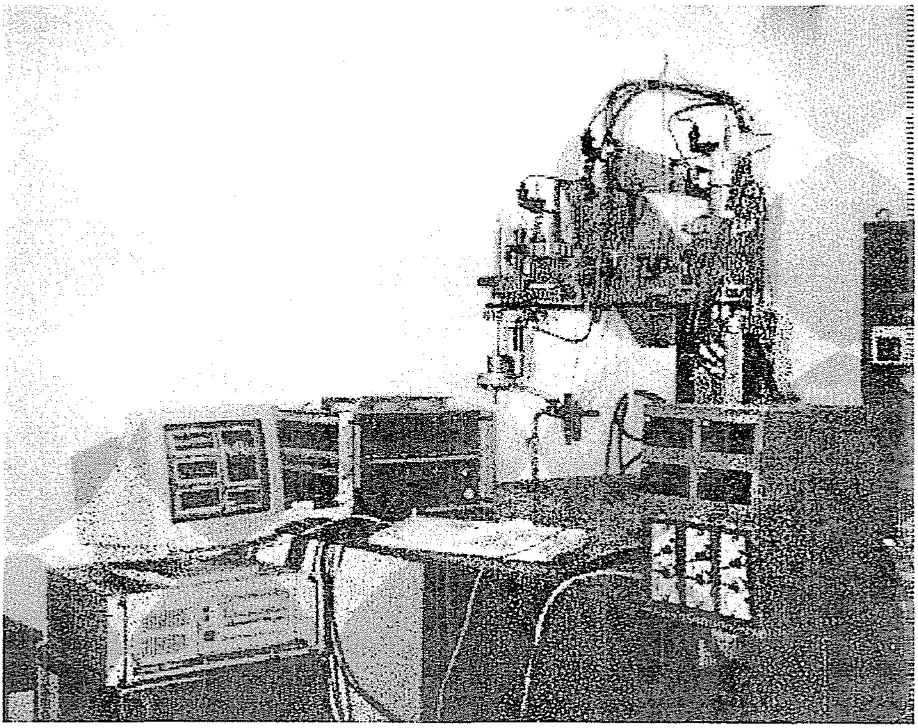


Fig. 3. Hardware platform for robotised manufacturing operation that includes one SCARA robot, an industrial PC, AT-6450 and NI-DAQ PC-LPM-16 cards, amplifiers, and force sensors

- An AT6450 card that is used to send and receive information to/from the servo controllers. The card is attached to the industrial PC and it sends command lines to the card.
- A NI-DAQ PC-LPM-16 card that is used to collect data from the force sensor. Three channels are used (F_x, F_y, F_z).
- A piezoelectric sensor (KISTLER) that is used to sense the force applied to the end mill tool during the cutting process.
- Amplifiers and power supplies used for sensor and motors as well.

In order to face the time constraint imposed by the real time intelligent adaptive controller the author has decided to use a PC-based real time operating system. The modules mentioned above have been programmed in this environment.

In addition, to increase the portability of the system, all the modules developed in this work have been programmed in Borland C.

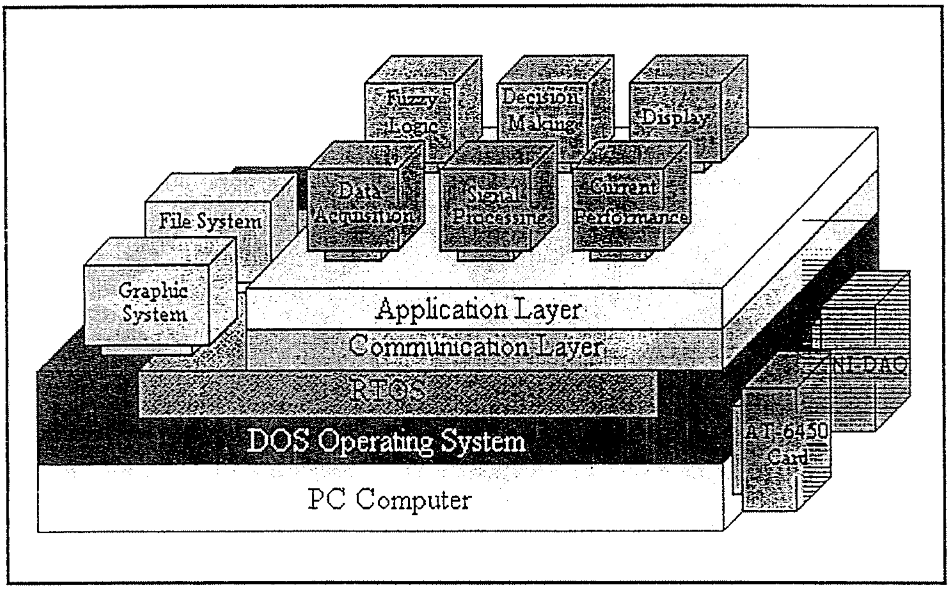


Fig. 4. Intelligent adaptive controller for robotised manufacturing operations based on an open architecture. Different modules that work in real time have been developed, including the human interface

5. Data Acquisition

The data acquisition module has been built using the communication primitives that the PC-LPM-16 card offers to software developers.

The communication primitives form the basic interface base between the computer and the force sensor. Amplifiers (see *Fig. 5*) for each force component of the sensor have been used in order to increase the accuracy of any measure coming from the data acquired. The output of each amplifier is directly attached to the PC-LPM-16 card, on the other hand its inputs are directly connected to the force sensor.

The primitives used in the development of the data acquisition module and their purpose are:

getDeviceToUse 0: retrieves the device number to be used by the communication interface

Get_DAQ_Device_info (deviceNumber, infoType, infoValue): allows to retrieve parameters pertaining to the device operation

AI_Clear (deviceNumber): clears the analog input circuit and empties the FIFO memory

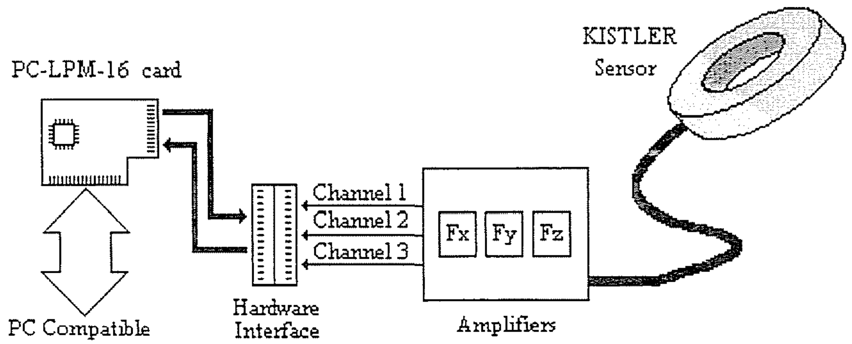


Fig. 5. Hardware configuration of the data acquisition system. A PC-LPM-16 card is used for acquisition purposes.

AI_Read (deviceNumber, chan, gain, reading): Reads an analog channel (initiates an A/D conversion on an analog input channel) and returns the unscaled result

AI_VScale (deviceNumber, chan, gain, gainAdjust, offset, reading, voltage): Converts the binary from the AI_Read call to actual input voltage

The functions call and their purpose have been extracted from the NI-DAQ Function Reference Manual for PC Compatibles of National Instruments. The parameters of any function call are in its general form.

6. Signal Processing

The use of minimum and maximum limits to define the permissible domain of normal and stable state of operation has been implemented by SZABO and WU [11] for monitoring of the tool condition in milling operations. Szabo and Wu, in their paper, use the cutting force as input signal of the monitoring system that provides estimation of the tool conditions. The monitoring indices for tool breakage detection are calculated in the time domain.

The author has decided to develop a modification of this strategy in order to define a dynamic threshold. An on-line dynamic threshold strategy, based on a moving average method, is used to define bottom and upper force limits used by the monitoring system, the system measures any deviation of the process and calculates the new dynamic thresholds within a constant period of time (a feature of the real time operating system.) A dynamic threshold strategy is able to calculate a new threshold when a minimal variation in the manufacturing process is detected.

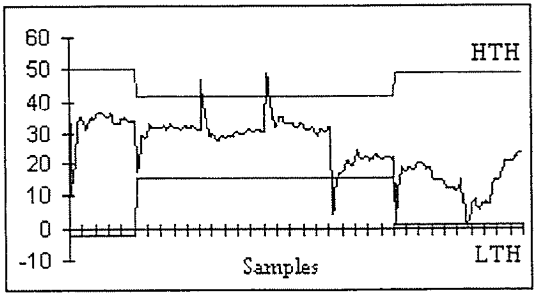


Fig. 6. Average of the window with 100 samples was used to produce the dynamic upper and bottom threshold limits (*Eqs.* (1) and (2) were used to calculate HTH and LTH limits).

Moving average of the samples (contained in the previous window) and standard deviation have been used to define the dynamic threshold for each window of size k.

The upper and bottom dynamic threshold are defined as follows:

$$HTH_j = \overline{x_j} + 2\sigma_j \,, \tag{1}$$

$$LTH_j = \overline{x_j} - 2\sigma_j \,, \tag{2}$$

where:

$$\overline{X_j} = \frac{1}{k} \sum_{i=k*(j-1)+1}^{k*j} f(x_i) \,, \tag{3}$$

$$\sigma_j = \sqrt{\frac{1}{k} \sum_{i=k*(j-1)+1}^{k*j} [\overline{x_j} - f(x_i)]^2} \tag{4}$$

are the average of the samples and the standard deviation of the window j.

LTH_j : j_th bottom threshold

HTH_j : j_th upper threshold

j : j_th window of size k, $j : 1, 2, \ldots, n$

k : window size

$f(x_i)$: i_th sample of the J_th window.

The *Fig. 6* shows the case when during the machining process a tool breakage is detected by the monitoring system. Observe the peaks in the figure indicating an increase of force over the upper dynamic threshold (HTH).

The difference between the peak and the threshold can be used to have a good tool wear diagnostic before the tool breakage occurs. Some experiments have shown that before a tool break occurs a continuous increase of cutting force is detected and the force signal is near to the upper dynamic threshold.

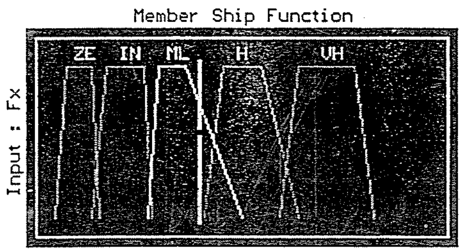
The computational algorithm that calculates the moving average is as follows:

```
void WindowFixAvg (float *p, int *Last, int *Index,
          float Buffer, float *Avg, float *StdDev, int N,
          Boolean *FirstWindow)
{
   if (*Last < N)/* First N Samples */
      *Avg += *(p + (*Last)++) = Buffer;
   else { /* The Window is full with data */
      if (*FirstWindow) {
         *Avg /= N;
         *FirstWindow = False;
         *Avg = GetAverage (p, N);
         *StdDev = sqrt (Variance (p, N, *Avg));
   }
   else
      if (*Index < N) /* Getting new data */
         *(p + (*Index)++) = Buffer;
      else {
         *Index = 0;
         *Avg = GetAverage (p, N);
         *StdDev = sqrt (Variance (p, N, *Avg));
      }
   }
   exit_task ();
}
```

7. Fuzzy Logic

Taking care of the time constraint a fuzzy logic system with its set of rules working on-line has been developed for supervise and control functions of the robotised manufacturing operation [12]. The fuzzy logic system (supervisory module in *Fig. 1*), is used to classify any event of the robotised machining system.

The membership functions [13] [14] of our implementation have a trapezoidal representation (see *Fig. 7*). Our definition includes 4 points of the trapezoidal function. Their limit values for each membership function are read from a text file.




```
typedef struct {
        char Name [MaxName];
        float BPleft,
            UPleft,
            UPright,
            BPright;
    } MSF, *PtrMSF;
typedef enum MSV {Z,IN,L,ML,H,VH};
```

Fig. 7. Representation of a membership function, its graphical user interface representation and its software implementation

The input to the fuzzy logic are the three components of force (received from the force sensor) and the resultant force as well.

The centroidal method is used by the defuzzification process in order to fire the respective rule. If more than one rule are fired, the system selects the rule that has much more activated area.

In addition we have created a fuzzy logic look-up table (*Table 1*) with 6 membership functions and 4 entries in order to define and simplify the interpretation of fuzzy rules. The fuzzy logic look-up table has been built based on experiments and data files containing different samples under different machining conditions that include tool breakage, tool wear and tool sharp.

Table 1. Fuzzy logic look-up table that defines some actions to execute for a particular operation state. Force components (F_x, F_y, F_z) and their respective resultant force are the inputs. The set of membership functions is defined with the values of VH (very high), H (high), ML (more or less high), L (low), IN (insignificant), ZE (zero)

	VH	H	ML	L	IN	ZE
F_x	Stop	Reduce	Warning	Continue	Increase	NP
F_y	Stop	Reduce	Warning	Continue	Increase	NP
F_z	Stop	UP+	Warning	Up-	Continue	NP
R	Stop	Danger	Reduce	Warning	Continue	NP

8. Decision Making

Any modification of operating parameters [15] is dictated by the control module of the Intelligent Adaptive Controller for robotised manufacturing operations (see *Fig. 1*) in the way of actions. Those actions are translated to their respective command lines that are sent to the respective hardware interface (AT-6450 card).

In order to control the system's performance (when an abnormal operation state is detected by the monitoring system) any value over the upper dynamic threshold as well as the force resultant are analyzed [16]. If the difference between the input and its respective threshold (see *Fig. 6*) is classified by the fuzzy logic system as High or Very High the system must be stopped. For other situations, where the differences are not so big, the fuzzy logic will indicate the right action that must be executed on the right direction (x, y or z-axis).

The next sub-set of fuzzy rules has been constructed in order to face some manufacturing problems. Note that each fuzzy rule has its associated action that must be indicated and carried out by the interpretation module shown in the *Fig. 2*.

1. if ((Fz is H)) then (Output is H) /* Stop */
2. if ((Fz is ZE) and ((Fx is ZE) or (Fy is ZE))) then (Output is ZE)
3. if ((Fx is H) and (Fy is ML)) then (Output is ML) /* Reduce X */
4. if ((Fx is ML) and (Fy is H)) then (Output is ML) /* Reduce Y */
5. if ((Fx is ML) and (Fy is ML)) then (Output is ML) /* Reduce X,Y */

The control system takes in account the z-axis as the most sensitive one (rule 1) because it holds the force sensor. The z-axis has the highest priority in the system and its evaluation is executed as soon as possible in order to avoid any crash or collision of the gripper with its environment [17] as well as any catastrophic failure of the robotised manufacturing system.

Commands that stop the movement of the robot or commands that reduce the speed of the motors are among the control actions that can be executed in order to protect the gripper and the manufacturing system as well. Command lines that move up the position of the gripper are also included. These commands protect the sensor and the gripper against collisions.

We want to remark our interest in any value over the upper dynamic threshold (for control actions) because it is a consequence of an abnormal operation state, however, we can include control actions when the value is below or near the boundary of the upper dynamic threshold in order to have a better diagnostic of the manufacturing operation as well as a good tool life diagnostic. Do not forget that before a tool breakage occurs a continuous increase of cutting force can be detected by the supervisory module (*Fig. 2*).

9. Human Interface

The entire necessary visual interfaces have been programmed in Borland C in order to create a versatile and portable version of this Human Machine Interface. The presentation on the screen of any value as well as any graphic plotted was programmed using the primitives of the graphical library that is part of Borland C.

10. Analysis of Performance

The current performance of the Intelligent Adaptive Controller for robotised manufacturing operations has been created as the interpretation of the fuzzy logic output and the combination of the resultant force (calculated from the x, y and z force components) acting on the force sensor, see *Fig. 8*. Depending on the analysis of the current performance the Intelligent Adaptive Controller performs the respective action on each axis (not always necessary).

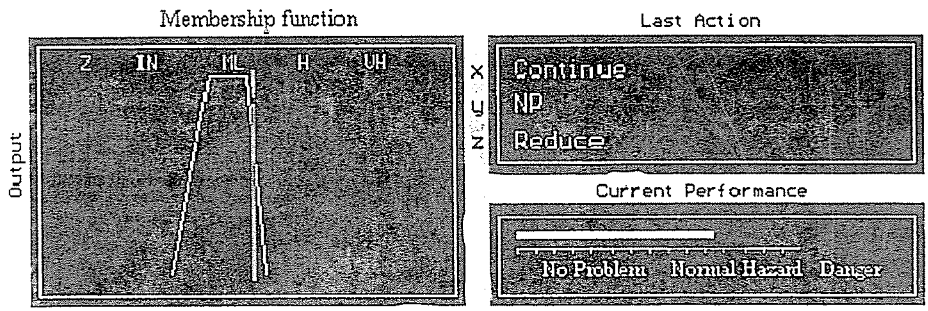


Fig. 8. The current performance is derived from the output of the fuzzy logic system. Observe that the system includes the last action performed on the respective axis.

The current performance includes 4 states of operation, they are:

(1) No Problem: Normal and stable operational state
(2) Normal: Normal operational state with significant increase of force
(3) Hazard: Warning operational state, this state identifies a possible hazard to the manufacturing system. The intelligent adaptive controller must adjust parameter(s).
(4) Danger: danger operational state, the intelligent adaptive controller must stop the robotised manufacturing operation as soon as this state is detected by the supervisory module.

11. Results

We have found that a moving average strategy for monitoring process defining dynamically bottom and upper threshold is better than a single and fixed threshold strategy based on the average of the samples acquired. Moving Average can be used for monitoring smooth and accurately the behaviour of the force signal meanwhile the dynamic threshold is used to define bottom and upper limits on-line.

This strategy offers a good opportunity to use the fuzzy logic system with the deviation between the force signal and the dynamic threshold, as inputs, in order to execute the right control action on time when an abnormal condition of operation is detected by the supervisory module.

Unfortunately, there is one problem, the size of the window must be fixed as small as possible in order to reduce the execution time of data processing within the window.

The identification and classification of event fails only in the case of short and quick transient states of operation, see *Fig. 9*. The use of fixed window size to define dynamic bottom and upper threshold limits can be improved by extending the dynamic capability to the size of the window, it can also be dynamically calculated.

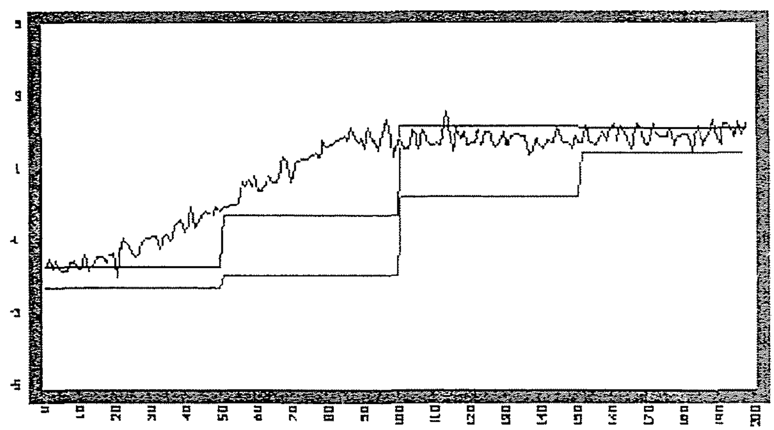


Fig. 9. The presence of short and quick transient states fail the fixed window average. To avoid this problem the size of the window must be as small as possible.

In order to have good results and face the execution time constraint the author of this paper suggests that the value of k must be within the interval $25 < k < 500$.

The benefits of the Intelligent Adaptive Controller for manufacturing operations based on an open architecture are:

- Low cost: Any standard PC can be used (industrial PC's are suggested for extreme working conditions).
- Interoperability: A simple PC-based real time operating system has been selected, if a new operating system is needed minimal modifications are necessary. The real time operating system runs in a DOS environment, it consumes few resources and no additional hardware is necessary.
- Portability: All functions, modules and general purpose user interface have been written in Borland C.
- Scalability: Modules can be modified, added to the system or removed from the system in order to improve the performance or to increase its capabilities.
- Reusability: Modules of acquisition, processing and user interface can be used by other applications. Perhaps some modifications must be necessary.

12. Conclusions

The competitive advantage of any company can be achieved by the incorporation of its own proprietary engineering knowledge in their controllers, resulting in manufacturing systems more versatile and flexible.

In the future, it will no longer be acceptable to take the approach of replacing existing controllers with newer and better models when a few new functions need to be added.

When a clear definition and implementation of Open Architecture becomes available the opportunities to develop a dynamic open factory, that will respond to the market demands in a short time, will become a reality. When this happens all the activities in the factory will be carried out by autonomous and intelligent devices, that will eliminate the human intervention reducing the rate of errors as a consequence.

The role of human beings would be confined to activities such as supervising, maintaining, and upgrading machines and equipment; shipping and receiving supplies and finished products; providing security for the plant facilities; programming, upgrading, and monitoring computer programs; and monitoring, maintaining and upgrading hardware.

Acknowledgement

The author of this work thanks to the Mexican Council of Science and Technology (CONACYT) and Institute Technologic of Monterrey (ITESM-CQ), for their sponsorship of this research project. Special thanks to Dr. Sándor Markos, Dr. Tibor Szalay, Professor László Monostori and Professor János Somló for their helpful comments during the development of this work, and to my supervisor Professor Mátyás Horváth.

References

[1] SPERLING, W. – LUTZ, P. (1997): Enabling Open Control Systems, *Robotics and Manufacturing*, Vol. 6, *ISRAM'97*, Montpellier, France.

[2] MAMORU, M. – TAKAAI, N. – HIDEYUKI, O. – MAKOTO, H. – KATSUYA, T. (1997): An Open Architecture CNC CAD-CAM Machining System with Data-Base Sharing and Mutual Information Feedback, *Annals of the CIRP*, Vol. 46, No. 1.

[3] Introduction to OMAC Users Group, *Objectives*, http://www.arcweb.com/omac/introduction.htm.

[4] Open System Architecture for Controls within Automation Systems, Introduction, http://www.osaca.org/sub-site/introduction/introduction/htm.

[5] Open System Environment for Controller Architecture, Introduction, http://www.sml.co.jp/osec/draft1.0/welcome.html.

[6] WRIGHT, P. K. – BOURNE, D. A. (1988): Manufacturing Intelligence, Addison-Wesley.

[7] SHRIDER, T. M.: Evolutionary Control Via Sensorimotor Input and Actuation, The ACM's First Electronic Publication, http://www.acm.org/crossroads/xrds4–3/evolution.html.

[8] RONG, Y.: Fuzzy-Logic Based Diagnostic Monitoring of Robot Operation, Department of Technology, Southern Illinois University at Carbondale, Carbondale, IL 62901–6603.

[9] NOVAK, A. – WIKLUND, H. (1993): Reliability Improvement of Tool-Wear Monitoring, *Annals of the CIRP*, Vol. 42, No. 1.

[10] DAVIES, A.: (August 1994): The Intelligent Machine, *Manufacturing Engineer*, pp. 182–184.

[11] SZABO, B. – WU, S. M. (1990): On-Line Techniques for Tool Condition Monitoring in Milling Operations, *IFAC* 11^{th} *Triennial World Congress*, 59–64, Tallin, Estonia.

[12] PASSINO, K. M. (June 1995): Intelligent Control for Autonomous Systems, pp. 55–62.

[13] FANN, W. R. – HSU, P. L.: Fuzzy Adaptive Control of Milling Processes, Department of Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, 30039 R.O.C.

[14] LANGARI, R.: Synthesis of Nonlinear Control Strategies from Fuzzy Logic Control Algorithms, Department of Mechanical Engineering, Texas A an M University.

[15] SOMLÓ, J. – NAGY, J. (1977): Advances in Computer-Aided Manufacture, D. McPherson, ed. North-Holland Publishing Company, pp. 293–305.

[16] HERNÁNDEZ, O. – MARKOS, S.– SZALAY, T. (1997): Fuzzy Logic Controller in Real Time Based on Open Architecture for Robotics Applications, *OSIN 1997 International Symposium*, pp. 103–110.

[17] PHAM, T. C. (1987): Robust Adaptive Axis Control of Manipulator, *Robotics and Computer-Integrated Manufacturing*, Vol. 3, No. 3, pp. 285–293.