Computer Integrated Manufacturing Systems Vol. 11, No. 4, p. 243–251, 1998

0951-5240/98/$ — see front matter

ELSEVIER

PII: S0951-5240(98)00013-5

Neural networks implementations to control real-time manufacturing systems

Gideon Cohen

RAFAEL, P.O. Box 2250, Haifa 31021, Israel

The main objective of advanced manufacturing control techniques is to provide efficient and accurate tools in order to control machines and manufacturing systems in real-time operations. Recent developments and implementations of expert systems and neural networks support this objective. This research explores the use of neural networks to control several manufacturing systems in real-time operations: robot manipulators, tool changes, conveyor systems and machine faults diagnosis. The main barrier to wide implementation of neural networks is the huge computation resources (times and capacities) required to train a network. This research represents the use of a multi-layer architecture of networks (input layer, several hidden layers and an output layer) to define single-valued inter-relationships between system participants and to avoid the need for long training processes. The use of neural networks to control the above-mentioned systems was evaluated from the following parameters: the architectures, network training methods, efficiencies and accuracies of networks to perform the task of control. Several conclusions related to neural network implementations to manufacturing systems were produced: (1) the multi-layer architecture fits the complexity of manufacturing systems; (2) neural networks are efficient to control real-time operations of machines; (3) machines which were controlled by neural networks performed accurate results; and (4) the use of several hidden layers can replace the need for long training processes and saves on computation resources.

Keywords: neural networks, multi-layer perception, manipulators manoeuvres, tools change management, closed lamp conveyor systems, machine faults diagnosis

Introduction

Neural networks have been considered recently as potential techniques to control manufacturing systems. Neural networks are bottom-up, case-based reasoning techniques. They are very powerful in their parallel processing, generalization ability and fault tolerance.

A neural network is composed of a set of artificial neurons (nodes) grouped in a number of layers (input, output and hidden layers). The output of a neuron of a specific layer (except that of the output layer) is fed as an input to its consecutive neuron. The layers and their neurons are framed and inter-related in a way that reflects the controlled system.

Several architectures of neural networks were considered: (1) networks as decision trees of threshold logic units; (2) several feed-forward networks; (3) multi-layer perceptron;, and (4) a meta-neural network.

The multi-layer perceptron architecture was chosen to be implemented in this research. This architecture represents itself as a network of numerous simple computing nodules (neurons), bound together via connections of different weights which determine the strength of the links. This structure provides a high level of flexibility and can be implemented to control a large variety of systems.

After a network is structured, the training stage is initiated. Network training generates the desired output for each individual input of a training data set. The training paradigm of interest is the back-propagation algorithm. For a given input, it generates the output by a forward pass. Then the difference between the actual output and the desired output is back propagated through the neural network (from the output layer to the input layer) to modify the weight matrices for the entire neural network.

In the reverse pass, training of the neural network takes place. Training the network means that all the weights are modified by a rule to minimize the difference between the actual output and the desired output.

The use of multi-layer architectures to define single-valued inter-relationships between system participants is presented in this research. This approach avoids the need for long training processes.

After the neural network is trained, it is tested against the records of a testing data set. For these records, the desired output is known. The generated output is checked against the desired output for that record. If there is a match, it is concluded that the trained neural network can classify the record correctly.

The neural networks technique was implemented in this research to control real-time operations of four manufacturing systems: (1) robot manipulators; (2) tool changes; (3) conveyor systems; and (4) machines faults diagnosis. Based on these implementations the match of neural networks technique to control real-time manufacturing systems was explored. Two criteria were used to evaluate this match: (1) accuracies of systems controlled by neural networks; and (2) the network's quick response to a variety of inputs. This response reflects the efficiency of the neural networks to control real-time manufacturing processes.

Implementation of neural network architectures to control manufacturing systems

Multi-layer architecture

A survey of several potential network architectures[1-7] leads to the conclusion that the multi-layer architecture fits most of the requirements of manufacturing systems control. The multi-layer architecture consists of an input layer, several hidden layers (in either serial or parallel order) and an output layer. Each layer consists of at least one node (artificial neuron). Node N_i at layer L_n is defined by four parameters: (1) an input vector, $\mathbf{I}_i = (i_1,\ldots,i_k)$; (2) a specific output a_i; (3) an activation function, f, represented in neural network terms by an arrow which links consecutive nodes; and (4) a training rate, μ_i. The input vector mimics the signals received by the neuron N_i from all the neurons (k neurons) in the previous layer. To each element of the input vector, a weight is associated that collectively makes a weight vector, $\mathbf{V}_i(V_1,\ldots,V_k)$. The weight vector mimics the strengths of synoptic connections between the neuron N_i and the other neurons. The inner product is defined by the notation:

$$I_i \times V_i = S = \sum_j i_j V_j, \text{ for } j = i \ldots k$$

This inner product represents the total weighted inputs (signals) received by the node N_i. The activation function, f, determines the level of excitation for the node N_i.

In the context of this paper, the multi-layer architecture represents either hierarchical structures of manufacturing machines or sequential stages of manufacturing processes. Since most of the manufacturing processes are recursive processes, their related network architectures are feedback-structured, the output layer is linked to the input layer.

Neural network training

The objective of the training stage is to train a network to react to a large variety of inputs and conditions. Several network training methods were considered: Hebb rules, Delta rules and back-propagation techniques.[2,3]

These training methods require huge computation resources to find the optimal weights of network's neurons. To save computation resources, the method of multi-layer architecture was implemented in this research. This method is based on several hidden layers to define single-valued inter-relationships between system participants. This approach results in accurate outputs for any pre-defined input.

Implementation of neural networks to control manipulators

Manipulator's maneuver control

A manipulator can be considered as a series of rigid parts in a kinematic structure.[8,9] Since a manipulator performs its tasks through the motion of its end-effector attached to its last link, the position and maneuver of this last link is of most significant importance.

The current position and orientation of the end effector is represented by the coordinates (x_n,y_n,z_n). Its initial coordinates are (x_o,y_o,z_o).

Manipulator's maneuver is resulted by the motions of its intermediate links, between the base link and its last link. The maneuvers of two consecutive links are generated by the maneuvers of the joints connecting the two links. There are n joints involved in the manipulator consisting of $n+1$ links. Each joint is driven by an individual actuator, which results in the displacement of the joint.

Modern manufacturing processes are based on accurate manipulators and end-effectors. Therefore manipulators' control techniques must fit high accuracy requirements.

A common technique to control industrial manipulators is based on a PID controller. Its operations relate to a reference control model. This concept is based on selecting an appropriate reference model resulting from robot kinematics. The adaptation algorithm is driven by the errors between the reference model outputs and the actual system outputs and modifies the feedback gains to actuators of the actual system. The implementation of a PID controller to non-linear manipulators provides a problem due to the linearity of the PID. This problem becomes more complex because operational parameters of manipulators 6 d.f are strongly interrelated. Therefore, the PID controller generates built-in dynamic errors. These errors are minimized by specific control parameters which are worked-out by many runs of the PID controller.

Neural network implementation to control manipulators' maneuvers

Network architecture
Input layer: The input layer is a matrix $[M \times 6]$, M = number of manipulator joints. The matrix consists of six parameters which specify the current position of each joint (rotation angle of $x_{i=1}$ axis about the z_{i-1} axis, the distance along the $z_{i=1}$ axis and rotation angle about the x_i axis) and the (x,y,z) coordinates.

Hidden layers: There are $M+1$ hidden layers. M layers represent the transformation matrices for each joint. The $M+1$th layer represents the 'arm matrix'. The transformation matrices are defined at the stage of network set-up by each joint position parameter defined by the input layer. The 'arm matrix' is defined at the stage of network set-up by multiplications of joint transformation matrices.

Output layer: The output layer represents the next location of each joint. Since these locations serve as inputs to their consecutive locations, the output layer is linked to the input layer. A feedback structure results. The input layer is updated by the new joint locations through this linkage.

Network training
Training a network, in the context of manipulators' control, is the set-up of all inter-relationships of the hidden layers.

Joint positions are defined at the input layer by their nominal values, their upper and lower bounds. Within their bounds the related 'arm matrix' varies insignificantly. When the accumulated changes in the related 'arm matrix' become significant, e.g. reach a defined limit, a new input matrix is defined to relate to a new 'arm matrix'.

At the stage of training a network, a large variety of computer runs are performed to define single-valued correlations between Euler-angles, (x,y,z) coordinates and 'arm matrices'. Since the Euler-angles and the joint positions of a specific robot vary in a known and limited range, the training stage can be accomplished in a short time period.

Network performances
Many manipulator trajectories controlled by neural networks were performed and evaluated in this research. These trajectories were varied in their Euler angles and in their initial positions. The performed trajectories were evaluated by two criteria: (1) required computation times to accomplish a specific trajectory; and (2) accuracy of a performed trajectory compared to the reference of a 'real world' trajectory.

Computation times
Table 1 gives the computation times required to accomplish a defined manipulator's maneuver. The computation times are the results of three control methods[10]; expert systems, numerical control and neural networks. 'Real world' maneuvers were used as references to evaluate the efficiency of these control methods.

Two types of manipulators were considered. Slow manipulators (represented by light loading ratios, e.g. long lead times to accomplish manipulator tasks) and high speed manipulators (represented by heavy loading ratios, e.g. short lead times to accomplish manipulator tasks).

Several conclusions can be derived from the results presented in *Table 1*. (1) The computation times required to control trajectories with neural networks are very close to the computation times required by expert systems and shorter than the computation times required by numerical control algorithms, by a factor of 2–6. (2) The efficiency of expert systems and neural networks to control manipulators is not affected by extended loading ratios. The computation times required by numerical algorithms are directly related to manipulators' loading ratios.

Trajectories accuracy
Figure 1 represents comparisons between manipulators' maneuvers in the 'real-world' and the same

Table 1 Normalized computation time of control techniques ('World time' is used as a reference)

	Loading ratios								
Control technique	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
'World time'	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Numerical control	0.13	0.14	0.16	0.1	0.1	0.1	0.09	0.06	0.06
Exp. sys.	0.03	0.02	0.03	0.03	0.03	0.02	0.02	0.02	0.02
Neural networks	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02

maneuvers performed by manipulators controlled by neural networks. The main conclusion which can be drawn is that 'real world' trajectories are repeated very accurately by manipulators controlled by neural networks.

Tool changes management

Tool changes management

Modern manufacturing floors are characterized by a large variety of products and small batches. Therefore, tools must be ordered in their magazines in a flexible manner to support frequent tool changes. This research explores the use of neural networks to determine the most flexible order of tools in their magazine.

Optimization problem formulation

Figure 2 illustrates schematically the tool change process. In the 'starting position' of the comb cassette it is possible to insert or remove an individual tool. In the first 'ejection position' it is possible to change as many as four tools and in the fully ejected position up to eight tools.

The automated change of cassettes is possible with a manually operated electronic lift truck or automatically with an unmanned transport system. The tools are alternately removed and inserted cassette-wise.

The optimization of the tool changes leads to a set of tool cassettes ordered in a way which results in minimum required steps of a chain magazine to generate a 'meeting' between 'home position' location and a cassette in its chain magazine.

The notation of the explored objective function is:

$$\min Z = \sum_{\substack{i=1 \\ j=1}}^{M} \left\{ |a_i - a_j|, |M - (a_i - a_j)| \right\} \times \gamma_{ij}$$

where:

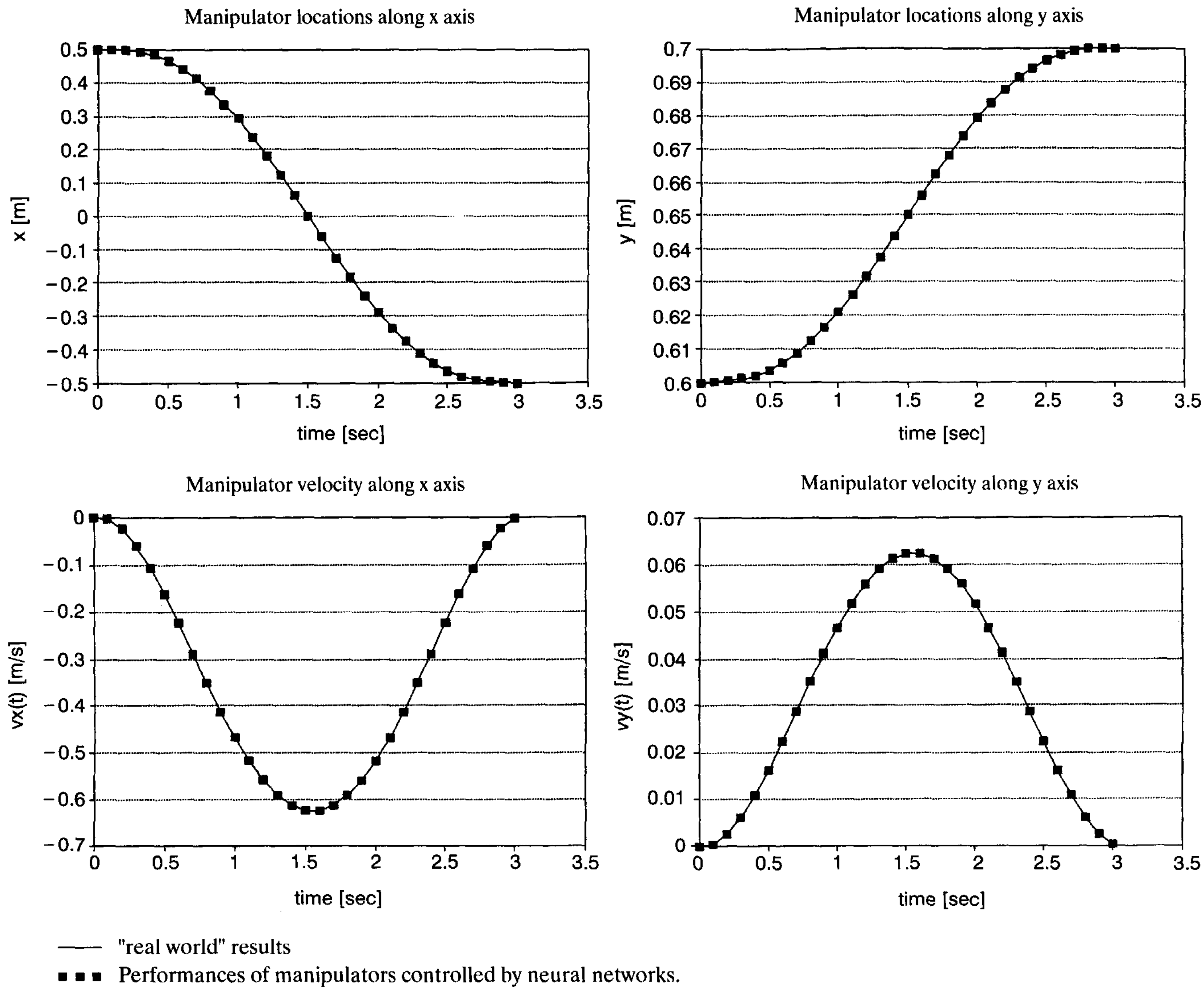

Figure 1 Manipulator maneuvers in the 'real world' compared to maneuvers controlled by neural networks

Z = number of required steps of the explored chain magazine;

M = number of different tool cassettes which can be installed in a specific chain magazine;

a_i = defined location at the chain magazine of a tool cassette which should be replaced;

a_j = defined location at the chain magazine of a new tool cassette which is required to match current manufacturing process.

Solution of the optimization problem

Since the chain magazine can move in either a clockwise or counter-clockwise direction, non-linear algorithms must be employed to solve the optimization problem. 'Toolbox' algorithms[3] are the most common methods used to minimize the number of required tool-cassette replacements for a specific manufacture process.

'Toolbox' algorithms relate to predefined manufacturing processes and their required sequence of tools, to determine the optimum order of a set of tools. 'Toolbox' algorithms require extensive computation times and huge memory capacities. Therefore 'Toolbox' algorithms are limited in their implementations to small problems.

Neural network implementation to control tool changes

Network architecture

Input layer: The input layer, in the context of tool changes, consists of scheduled manufacturing processes. Each manufacturing process is identified and represented by a pre-defined code to provide a compact structure of a layer.

Hidden layers: The hidden layers consist of tools needed to perform all potential manufacturing processes. Each manufacturing process is represented by a specific tools set. Each tools set is linked to a specific manufacturing process specified at the input.

Output layer: The output layer represents the tools which are required to execute the scheduled manufacturing processes. The tools are ordered in their chain magazine in a way which results in the shortest route of the chain to perform all the required changes.

Network training

Training a network, in the context of tool changes, is the set-up of hidden layers single-valued inter-relationships. This set-up consists of four stages: (1) definition of the inter-relationships between manufacturing processes and their related tools; (2) imple-

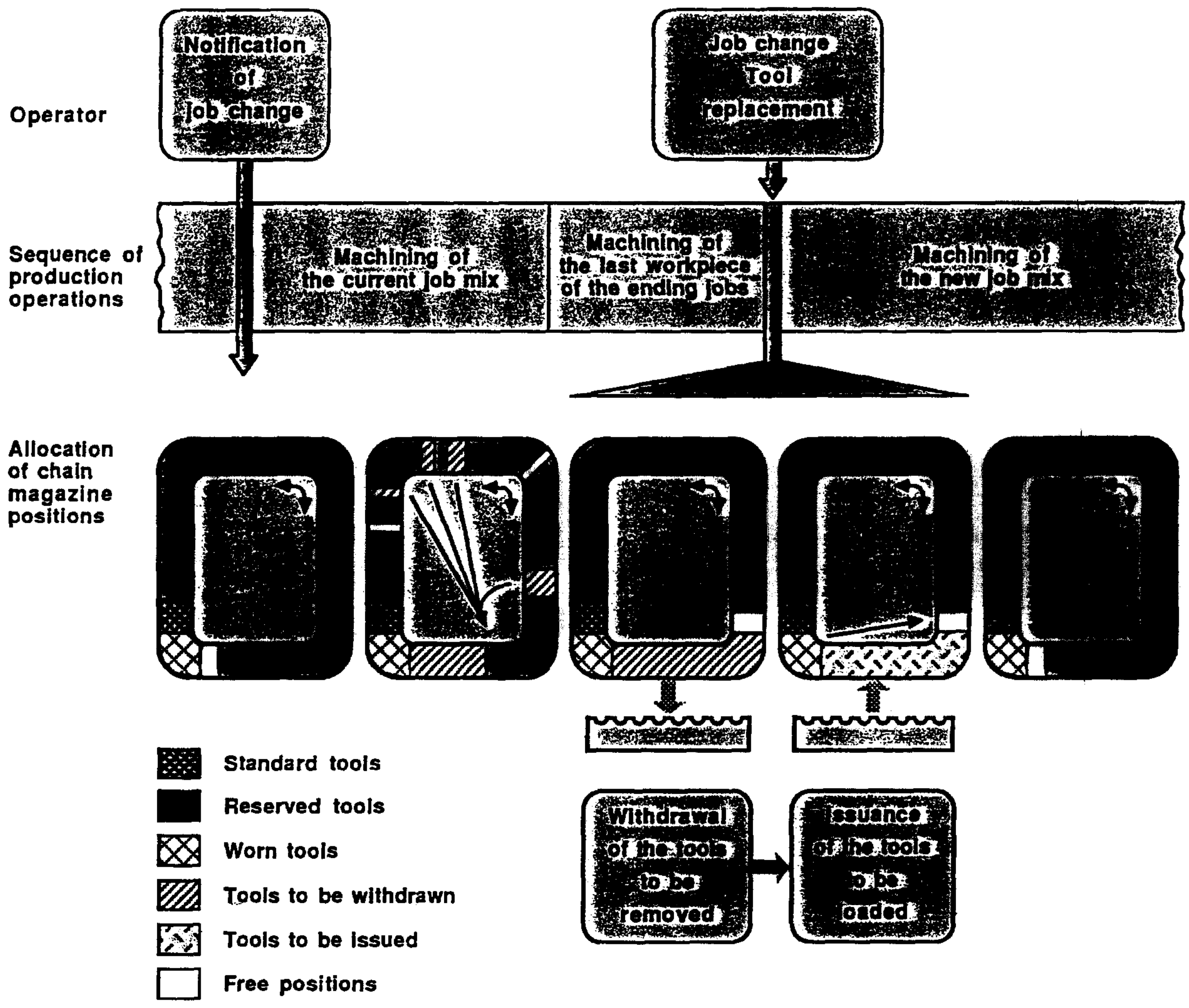

Figure 2 Tool change operational sequence

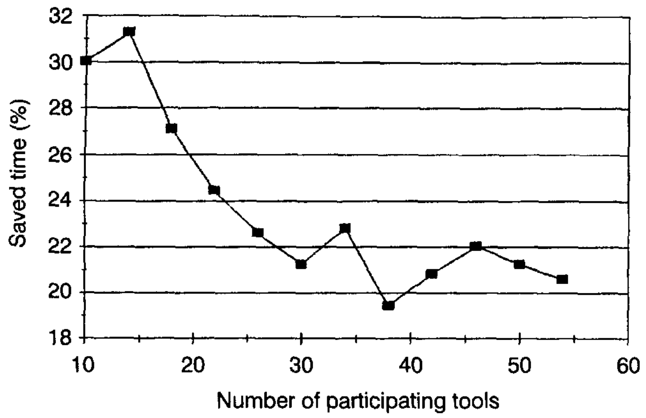

Figure 3 Saved time is inversely related to the number of participating tools

mentation of an algorithm to search the required tools for the performance of defined scheduled manufacturing processes; (3) estimations of expected frequencies of tool use; (4) conclusions of tool order to provide minimum steps of a chain magazine to generate a 'meeting' between 'home position' location and a cassette in its chain magazine.

Network efficiency

Network efficiency to control tool changes is evaluated in terms of saved time during required tool changes, compared to tool changes based on a random order. *Figure 3* shows the saved time achieved by the network which was developed in this research. The saved time is inversely related to the number of participating tools.

When the number of participating tools is extended, the complexity of training a network becomes higher and requires huge resources of computation (time and capacities). Since these resources cannot be provided, the network remains untrained to change large numbers of tools.

Closed loop conveyor systems

Closed loop conveyor systems

The efficiency of material handling facilities are of most significance in the efficiency of the entire manufacturing systems.[11–13] A typical manufacturing system can be considered as work stations ordered in a specific way and combined by a material handling system. The conveyor belt is a material handling system which is widely used to transport materials between machines. The conveyor belt system consists of a conveyor, loading station(s), work station(s) and unloading station(s).

Conveyor systems are referred to as queuing systems, characterized by a stream of parts arriving at a service station according to a stationary Poisson stream without after-effects, with parameter λ, and are served at several service channels, numbered in order $1,2,\ldots,n$.[14,15] Each channel contains one server and the service time distribution for each of the channels is a negative exponential with parameter μ. The waiting space in front of each of the service channels is fixed, say for $N_1,N_2,\ldots,N_n$ parts in channels, $1,2,\ldots,n$ respectively. The incoming part first goes to channel 1. If channel 1 is filled to its capacity, the part goes to channel 2, otherwise it takes its position in channel 1. Each incoming part tests each of the n channels in the prescribed order until it finds some channel that can accommodate it. In the case when the incoming part finds all the channels filled to their capacity, it leaves without being served and is 'lost' to the system. The conveyor system referred to in this research consists of the following elements (see *Figure 4*):

- Three conveyor belts to transfer parts. Conveyor 1 moves in a direction that is opposite to the direction of the two other conveyors. All three conveyors move at a constant velocity, 9 m/min.
- Loading station integrated with Conveyor 1 ('ENT' in *Figure 4*). The loading station is equipped with a bar-code reader to identify the entering parts.
- Unloading station integrated with Conveyor 2 ('Exit' in *Figure 4*).
- Four work stations integrated with Conveyor 3.
- Seven transfer stations. The transfer station aim is to transfer parts between two consecutive conveyors. A transfer station consists of two small conveyors and a step motor. The small conveyors move in an orthogonal direction relative to the direction of the main conveyor. The small conveyors move at a constant speed, equal to the main conveyor speed of 9 m/min. When a part reaches a transfer station, it is raised by the step motor to be separated from the main conveyor.
- Stoppage pins to control the parts maneuvers on the conveyors. A stoppage pin is raised to stop a part in front of a station. The stoppage pin is lowered to generate a continuous platform and to allow a part to continue on its way to the next station.
- All the aforementioned elements are combined in a central processor which in this research is equipped with the developed neural network. The inputs received from the conveyor system elements initiate the neural network reasoning process to work out the required operational parameters.
- Four waiting stations to extend the waiting areas in front of the stoppage pins.

Neural network implementation to control closed loop conveyor

Network architecture

Input layer: The input layer represents the current position of the conveyor system. It consists of several elements which represent the status of the conveyor system components (work stations, waiting stations, conveyor belt, etc.). The input layer indicates either '0' (to represent empty stations or unutilized belt) or '1' (to represent occupied stations or utilized belt).

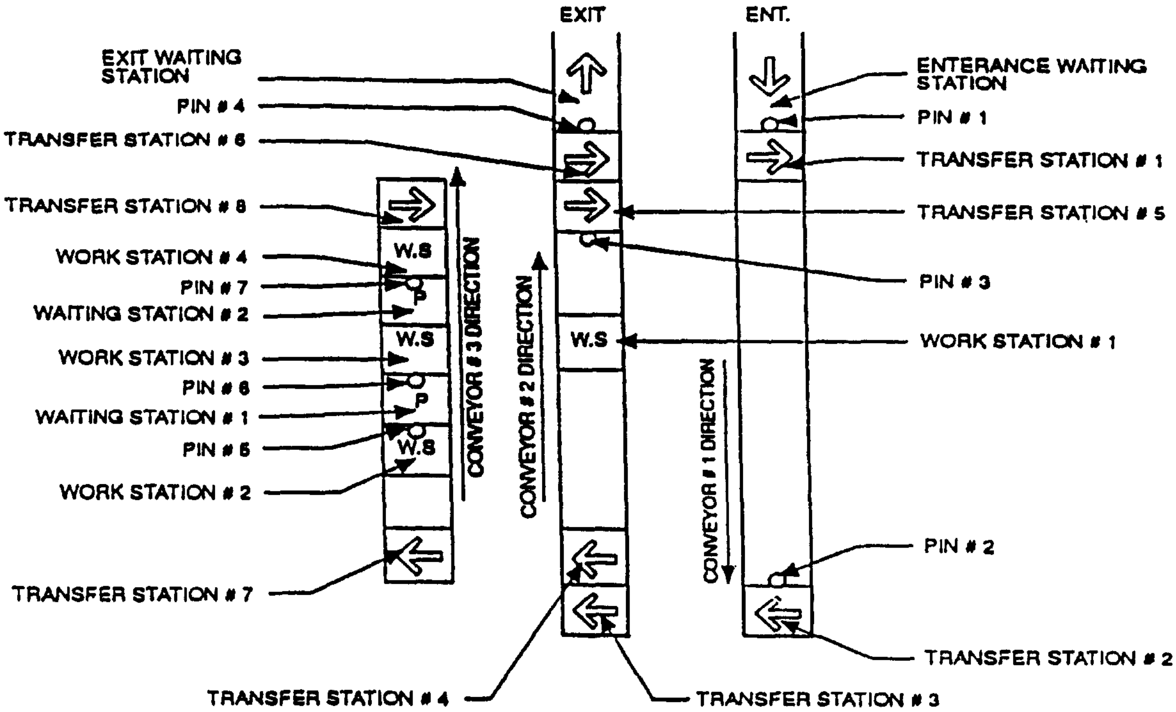

Figure 4 Conveyor system structure

The input layer indications are changed by electronic signals which are generated and sent by the conveyor component, the status of which changed. The electronic signals are converted to numeric data to update the input layer.

Hidden layers: The links between the nodes of the hidden layers represent the dynamics of the conveyor system, which is defined by the system's operational rules.[15]

Output layer: The output layer represents the system's next status. It results from the current input layer manipulated by the relevant operational rules, represented by the hidden layers.

Network training

Network training, in the context of closed loop conveyors, is the set-up of the links between the nodes of the hidden layers themselves and between the hidden layers and the input layer.

Each reaction of the network is initiated by an electronic signal sent from one of the system's participants to the central computer. This signal indicates that the system's status is changed. The input layer is updated and initiates the search of the hidden layers to find a specific layer which represents the relevant operational rule. When the relevant hidden layer is met, the system's next status is generated and represented by the output layer.

Network efficiency

The performances of a conveyor system controlled by a neural network were compared to conveyors which were controlled by two different control techniques:[15] (a) a control technique based on Johanson rules; and (b) a control technique based on the 'First in, first out' (FIFO) rule.

Table 2 shows the performances of a conveyor system associated with parts which need very short manufacturing process time ($t = 5$ min). *Tables 3* and *4* relate to conveyor systems associated with parts which need longer manufacturing process times (15 and 25 min).

Table 2 Conveyor system performances

	Performance (min)		
Control technique	**Process time (a)**	**Waiting time (b)**	**System efficiency ($a/(a+b)$)**
Johanson rules	35	10	77.7
Expert system	75	90	45.4
Neural network	80	100	44.4
Random entries and FIFO rule	85	110	43.5

Parts need short process time ($t = 5$ min). System operation time = 60 min.

Table 3 Conveyor system performances (t = 15 min)

Control technique	Performance (min)		
	Process time (a)	Waiting time (b)	System efficiency ($a/(a+b)$)
Johanson rules	105	30	72.4
Expert system	145	110	56.8
Neural network	160	140	53.3
Random entries and FIFO rule	175	170	50.7

System operation time = 180 min.

The most significant conclusion that can be evaluated from the results represented by *Tables 2–4* is that the importance of intelligent control methods, either expert systems or neural networks, is directly related to manufacturing process times.

If a manufacturing process time is short, system stations are very often free for new parts. Therefore, parts can maneuver in their basic routes and smart reasoning processes to work out complicated routes are insignificant. If a manufacturing process time becomes longer, system stations are often occupied for new arriving parts. These parts must perform complicated routes until a required station becomes free. Smart reasoning processes are of major importance in working out these routes efficiently.

Machine faults diagnosis

Machine faults diagnosis—its role in manufacturing systems

Machine breakdowns have a significant effect on manufacturing systems efficiency.[16–19] Most of the modern CNC machines are equipped with built-in self-fault diagnosis capabilities. These capabilities are based on operational control parameters (temperature, oil level, air pressure, accuracy, etc.). If one of these operational parameters is found to be beyond its pre-defined approved operational range, an indication is signaled to attract the operator's attention.

The most advanced CNC machines are equipped with limited capabilities of software algorithms to perform automatic shut-downs, whenever a specific operational parameter reaches beyond its operational range. Comprehensive control techniques are required to improve machine faults diagnosis capabilities and to extend their ability to cure machines. Such comprehensive control techniques consist of two elements: diagnostic trees to formulate machine structures and fuzzy multiple attribute decision making to execute smart diagnosis processes. Neural networks can combine the benefits of these techniques.

Neural network implementation to perform machine faults diagnosis

Network architectures

Input layer: The input layer represents potential faults of machines. The initial indication of this layer is ('0', '0', ..., '0') to indicate that no faults have occurred. When a specific fault appears, its indication at the input layer is changed into '1'.

Hidden layers: The hidden layers consist of three layers. The first layer represents machine components which may share a specific indicated fault. The second layer represents pre-defined approved operational ranges of machine components. The third hidden layer represents current operational values of these components. If the current operational value of

Table 4 Conveyor system performances (t = 25 min)

Control technique	Performance (min)		
	Process time (a)	Waiting time (b)	System efficiency ($a/(a+b)$)
Johanson rules	175	50	77.7
Expert system	205	110	65.0
Neural network	220	130	62.9
Random entries and FIFO rule	235	170	58.0

System operation time = 300 min.

Table 5 Diagnosis time related to main malfunction symptoms

Machine's malfunction symptoms	Diagnosis time related to malfunction symptoms (s)
Temperature beyond a defined upper limit	3.2
Temperature below a defined lower limit	3.7
Engine's oil level below operational lower limit	3.9
Air pressure beyond a defined upper limit	3.2
Air pressure below a defined lower limit	3.5
Reduced machine's response velocity	3.1

a specific component is beyond the bounds of its pre-defined approved operational range, this specific component is diagnosed to be the reason for the machine's malfunction.

Output layer: The output layer represents the component which is diagnosed to be faulty.

Network training

Training a network, in the context of machine faults diagnosis, is performed in two stages. (1) Machine components are linked to the input layer. If a specific fail occurs, an electronic signal is sent from the fault component to update the input layer's indication. (2) Generation of hidden layers single-valued inter-relationships.

Network efficiency

The criteria to determine network efficiency to diagnose faults is the lead-time between the first appearance of the machine's malfunction symptoms and the accomplishment of the diagnosis of the related fault. *Table 5* gives the diagnosis times related to a machine's main malfunction symptoms.

The achieved results given in *Table 5* lead to the conclusion that the use of neural networks to diagnose machine faults meets the requirement of the machine's real-time operations (response time is less than 5 s).

Summary and future outlook

The structures of multi-layer neural networks fit the complexity of manufacturing systems. The structure of several hidden layers provides an efficient tool to represent complex single-valued inter-relationships between a large variety of manufacturing machines and processes. The feedback structure which links the output layer of a network with its input layer fits the representations of machines which perform incrementally.

The use of hidden layers to represent the single-valued inter-relationships between manufacturing machines and processes is a flexible technique (any two nodes can be linked). It results in quick and accurate responses of the network.

These conclusions can support further implementations of neural networks to control machines in real-time operations.

References

1 **Cherug, P, Banghan, K, Beaudry, M and Trempwan, P** 'Mobility, an intelligent network in action'. *IEEE Conference Publication* (1991) pp. 24–28.

2 **Dutta, S and Shekhar, S** 'Bond rating: a non-conservative application of neural networks'. *Proceedings of IEEE International Conference on Neural Networks* (1988) pp. 443–450.

3 **Hertz, Z, Krogh, A and Pulmer, R G** Introduction to the theory of neural computation. Redwood City, CA, Addison-Wesley: 1991.

4 **Lawrence, J** Introduction to neural networks and expert systems. Nevada City, CA, California Scientific Software: 1992.

5 **Medsker, L and Liebowitz, J** The design and development of expert systems and neural networks. New York, NY, Macmillan Publishers: 1994.

6 **Neural Wave Inc** Neural computing: a technology handbook for Professional II/Plus and Neural Works Explorer. Technical Publications Group, Pittsburgh, PA: 1993.

7 **Rakes, T** 'Research in the neural network/expert systems area', *Decision Line* Vol 22 (1991) 11–14

8 **Fu, K S, Gonzales, R C and Lee, C S G** Robot arm kinematics. Robotics control, sensing, vision and intelligence (19910.

9 **Kreamer, W C** Measuring robot and sensor accuracy. *Proc. Robots West Conf.*, Anaheim, CA (1984).

10 **Cohen, G** Real time expert systems as a necessary tool to achieve shop floor efficiency. *Conference Proceedings: Annual Israel Conference on Advanced Production Technologies* (1991).

11 **Ho, K L, Law, S and Law, H W** Multi-purpose conveyor system. *Proceedings of 29th SICE Annual Conference*, Volume II (1991), 641–643.

12 **Merkel, N** 'Automation of belt conveyor systems', *Tech. Inf. Process. Autom-Electr. Power Install.* Vol 20 (1990) 3–6

13 **Pourbabai, B** 'Performance modeling of closed loop material handling system', *European Journal of Operational Research* Vol 32 No 3 (1987) 340–352

14 **Bastani, A S** 'Closed-loop conveyor with breakdown and repair of unloading stations', *IIE Transactions* Vol 22 No 4 (1990) 92–96

15 **Cohen, G** 'Expert system to control and to design closed loop conveyor systems', *Expert Systems with Applications* Vol 7 No 4 (1994) 483–494

16 **Dallimonti, R** 'Smarter maintenance with expert systems', *Plant Engineering* Vol 41 (1987) 51–56

17 **Davis, R** 'Reasoning from first principles in electronic trouble-shooting', *International Journal of Man–Machines Studies* Vol 19 (1983) 403–423

18 **Davis, R** 'Diagnostic reasoning based on structure and behaviour', *Artificial Intelligence* Vol 24 (1984) 247–410

19 **Roberge, P R and Adamson, M S** 'Development of a deep knowledge diagnostic expert system using fault tree analysis information', *Canadian Journal of Chemical Engineering* Vol 69 No 1 (1991) 76–80