International Journal of Production Research,
Vol. 43, No. 1, 1 January 2005, 147–168

Taylor & Francis
Taylor & Francis Group

Multilevel intelligent scheduling and control system for an automated flow shop manufacturing environment

M. A. S. MONFARED† and J. B. YANG*‡

†School of Engineering, University of Alzahra, Tehran, Iran
‡Manchester School of Management, UMIST, Manchester M60 1QD, UK

(*Received October 2003*)

Current manufacturing scheduling and control systems are incapable of coping with complex system dynamics inherent in real-world situations and, hence, human intervention is required to maintain real-time adaptation and optimization. A unique feature of biological intelligent systems is that they build and improve over their communication, decision-making and control structures in real time autonomously. A challenge is now emerging in the design of manufacturing systems where on-line adaptation and optimization become increasingly important. This paper reports on the development of a new integrated intelligent scheduling and control system for an automated manufacturing environment using a multilevel approach. At the first level, a conventional scheduling and control system is considered, then at the second level, a new fuzzy logic mechanism is developed to enable the conventional system to improve and perceive the changes of system parameters adaptively. A new perturbation mechanism is embedded in the third level to implement on-line optimization for coping with the more complex structural changes of system dynamics. The final level is composed of artificial neural networks that can learn from experiences provided by the perturbation mechanism. The approach is designed to improve system intelligence gradually to cope with various forms of systems dynamics. A fully automated flow shop manufacturing system is taken to demonstrate this approach.

Keywords: Automated flow shop; Intelligent systems; Scheduling; Control; Fuzzy logic; Neutral nets

1. Introduction

In recent years, competition in advanced manufacturing industry has intensified the pressures on all aspects of manufacturing systems. It is therefore essential that powerful techniques and technologies be developed that can constantly improve lead time, flexibility and responsiveness. To achieve these capabilities, manufacturing systems manifesting greater intelligence and autonomy are of significance. One of the features of intelligent systems is that they have real-time built-in capability to communicate with system's environment, perceive changes and adapt to such changes.

Intelligent systems need to be integrated in various aspects of often complex manufacturing tasks (e.g. product design, process design, operations, quality control

*Corresponding author. Email: jian-bo.yang@umist.ac.uk

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online © 2005 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/00207540412331282033

and inspections, and scheduling and control systems). The ultimate purpose of such integration is to achieve a fully intelligent manufacturing system where a customer could interactively communicate with a large-scale intelligent system in order to enquire about feasibility of ordering a product, to estimate delivery time and price, to place orders, to carry out product design and process design, to procure material, to implement production scheduling and control, and finally to arrange the delivery of products. However, the immediate purpose of designing intelligent manufacturing systems must be set more realistically. This is because current manufacturing systems are still heavily based on human intelligence.

To achieve an intelligent manufacturing system, significant effort must be made to improve the current techniques in manufacturing science and engineering (Askin and Standridge 1993, Gershwin 1994). One of the key limiting elements is that the working mechanism behind human intelligence has not yet been fully understood. Therefore, instead of trying fully to replace human intelligence with machine intelligence, the immediate effort ought to increase machine intelligence as much as possible and to build manufacturing systems where human direct interventions are minimized.

Integrated intelligent scheduling and control systems have been investigated in process industries and in highly repeated manufacturing systems where material flow can be considered as continuous flow (Gershwin 1994, Bodington 1995). However, it is of increasing concern to build integrated intelligent scheduling and control systems in cases where the discrete nature of manufacturing can not be neglected and where the manufacturing systems are constantly exposed to dynamically changing environments (Ramadge and Wonham 1987, Ho 1989, Suri 1989, Buzacott and Shanthikumar 1993, Antsaklis 1994, Werbos 1994, Prabhu and Duffie 1999, Trentesaux *et al.* 2000, Akturk and Ozkan 2001, Kumar and Jaikumar 2002, Abdelhameed and Talbah 2002).

Chen *et al.* (1996) proposed an intelligent cell control architecture in order to handle the stochastic nature of an automated manufacturing system. Their new architecture not only provides feasible control actions, but also continuously improves control decisions. The framework for real-time scheduling and control was tested on a hypothetical flexible manufacturing cell. Diaz *et al.* (1991) studied the scheduling and control problem in a steel plant in an integrated fashion. They proposed the use of programmable logic controllers (PLCs) and digital control systems with distributed structures (DCs) at the lowest system level, a computer system for global control of the plant, and a business computer system for aggregate master planning. The investigation showed that a fully automated scheduling and control system is still hard to obtain due to uncertainty associated with the steel plant system (Kumar and Jaikumar 2002).

To build an intelligent manufacturing system, one promising research direction is to develop hybrid methodologies by incorporating concepts and techniques from operational research, control theory (Monfared 1997, Kogan and Khmelnitsky 2000, Lu 2001, Cervin *et al.* 2002).

The present paper explores a new framework for dealing with scheduling and control of an automated manufacturing system. The purpose of this research is to show how non-stationary dynamics in system parameters and structures can be adequately modelled. Specifically, we consider a system that moves from one stationary condition to another where the time elapsed in any condition is considerably long to allow the system to detect the changes in the system condition.

A novel fuzzy predictive control system will be developed to cope with parameter dynamics. We also develop a perturbation procedure to provide automatically new data for structural dynamics. By using various feed forward neural networks, we then proceed to illustrate that parameter and structural dynamics can be learned on a real-time basis. The proposed integrated scheduling and control system hence illustrates an intelligent system capable of on-line adaptation and optimization in a non-stationary stochastic environment.

The paper is organized as follows. In section 2, a manufacturing system is modelled that illustrates the different components of the physical system. A fuzzy predictive control system is modelled in section 3 in which parameter dynamics can be detected and recognized. In section 4, a perturbation procedure is developed as an optimization tool to cope with structural dynamics. In section 5, the applicability of artificial neural networks (ANNs) is investigated to illustrate an enhanced intelligent system using data provided by optimization tool. In section 6, the integration methodology developed and applied in this paper is re-examined and clarified. The paper concludes in section 7.

2. Physical system description

The design of an integrated intelligent scheduling and control system requires that both scheduling and control problems be treated in an integrated way as is considered in the following automated flow shop manufacturing system.

2.1 *System configuration*

A flow shop manufacturing system (or cell) is illustrated in figure 1. The cell is formed by connecting nine machines as follows: three robots for loading and unloading tasks, one conveyor belt for handling incoming parts, three rotary spiral racks for storing parts, one heat treatment machine for strengthening the physical properties of the parts through a heat treatment operation, and one automatic guided vehicle (AGV) for handling finished parts.

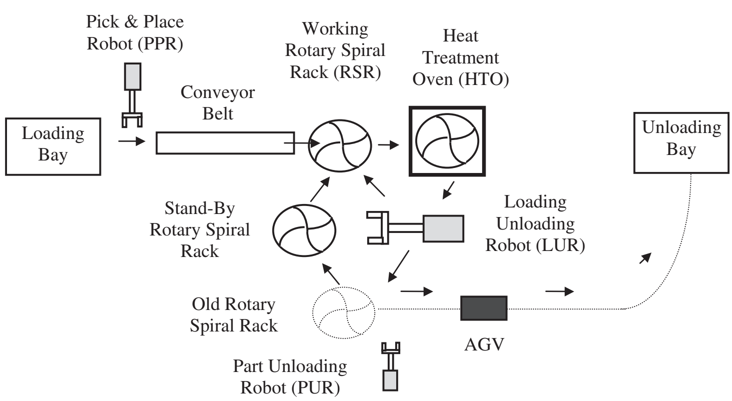
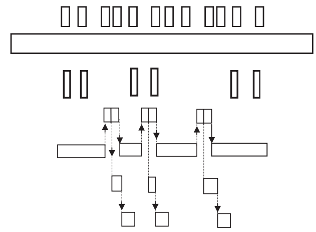


Figure 1. Fully automated flow shop.

Table 1. Logical sequencing of a working cell.

	Equipment	Unit	Timing	Operating time
M1	Pick and place robot	Part	Fixed	
M2	Conveyor belt	Part	Fixed	
M3	Rotary spiral rack	Part	Fixed	
M4	Load/unload robot	Batch	Fixed	
M5	Heat treatment oven	Batch	Variable	
M6	Part unloading robot	Batch	Variable	
M7	Automated guided vehicle	Batch	Fixed	

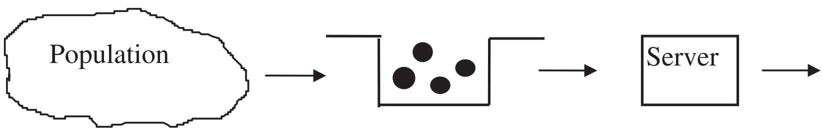


Figure 2. General view of the queuing system.

The working logic of the system is as illustrated in table 1. Parts arrive randomly at the loading bay, and then the pick and place robot (PPR) is activated to pick and place the parts onto the conveyor belt. As the parts reach the end of the conveyor belt, they will automatically feed into the working rotary spiral rack (RSR) which stores the parts. The working RSR is moved to the heat treatment oven when the oven is free and sufficient parts have been collected.

To allow a new batch to be processed, the following sequences are taken. The old batch sitting in the heat treatment oven will be taken out using the loading/unloading robot (LUR) and located next to the part-unloading robot (PUR). The LUR then moves the working RSR situated in front of the conveyor belt to the oven. The LUR also moves the stand-by RSR to the front of conveyor belt. In the meantime, the PUR unloads the old batch's parts into the AGV.

After all parts have been unloaded into AGV, two actions are activated. The first is the activation of the AGV to move the parts to the unloading bay and return back for the next operation. The second action is taken by the LUR to move the emptied old RSR to the stand-by position. All machines are of on/off type but the conveyor belt is on all the time during the production. The time required for performing all operations is fixed except for the HTO processing time and the part unloading time. These two operations are random variables because of the random nature of the number of parts in a batch. The randomness makes this dynamic system different from the so-called conventional automated systems.

2.2 *Simple model*

The manufacturing system described in figure 1 and table 1 can be simplified as a queuing system, as illustrated in figure 2, where parts arrive randomly, accumulate in a queue and then receive service by a single server. Since the parts are served in a batch-wise fashion, this queuing system is classified as a batch service queuing process (or single batch production system).

Tremendous efforts have been put into the analysis and control of the above single-stage, single-product production (or service) systems. The batch service system was originally modelled by Deb and Serfozo (1973) by using dynamic programming and stochastic queuing system theory. They proved that there is an optimal threshold control policy that trades off between holding cost and operational cost. The policy is that a batch of parts (or customers) will be produced (or served) if and only if the number of parts (or customers) reaches or exceeds a certain threshold or control limit, denoted by Q. The single batch processing system provides a simple and straightforward production control system. However, the design of such a system is full of challenge, as will be discussed below.

2.3 *Stochastic scheduling and control system*

In a single batch production system, such as a heat treatment oven, parts arrive dynamically in accordance with a certain distribution (e.g. Poisson). The parts are collected as a batch until the number of parts becomes equal to or larger than a threshold. The processing time also changes dynamically in accordance with a certain distribution (e.g. negative exponential). Such a single batch processing system can also be found in a transportation system, in the form of a shuttle bus service between an airport terminal and a remote parking lot (or a transit bus station). In either case, if the parts (or passengers) arrive deterministically, an optimally scheduled service might be the best policy.

Deb and Serfozo showed that at any given time when there are parts (or passengers) waiting in the system, a certain control limit offers an optimal scheduling and control policy. In other words, a batch of X parts is processed if and only if X exceeds a threshold limit Q. The working logic of the control limit policy, as is considered in the literature, is demonstrated in figure 3.

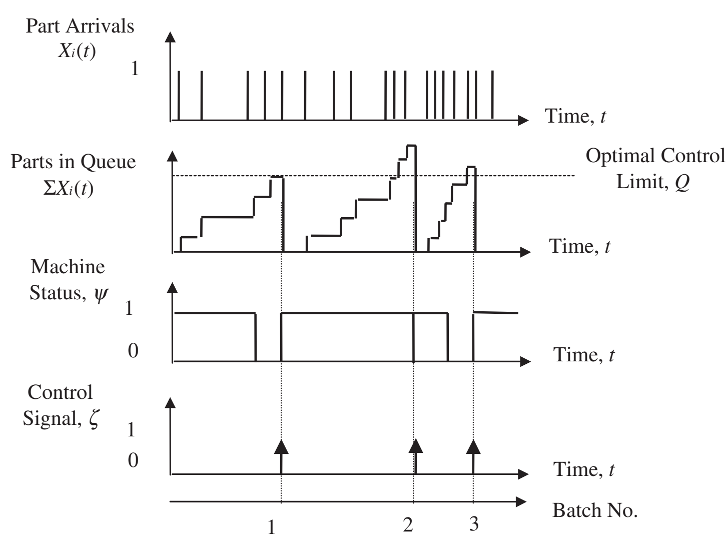


Figure 3. Illustration of the working system.

2.4 *Computation of the optimal control limit*

The threshold policy is general and applicable to any single batch processing system. This is based on the calculation of an expected continuously discounted cost function over an infinite length of time, $V_\alpha(Q; X)$, or an average cost per unit over an infinite time, i.e. $V(Q)$ (for details, see Deb and Serfozo 1973). The computational method for an optimal control limit has also been developed by Deb and Serfozo when the following conditions hold:

- Cost function $C(t)$ consists of the production cost $P(t)$ and the holding cost (or penalty cost) $H(t)$.
- Production cost of X items is assumed to be $R + cX$, where R and c are non-negative constants. Operational costs are discounted with the rate of α. (Note that the control limit policy holds for both discounted cost function and for non-discounted (or average) cost function.)
- Holding cost of X items per unit time in the system is assumed to be hX, where h is a constant.
- Customers arrive with Poisson distribution.
- Batch processing time follows an exponential distribution with a parameter of μ, e.g. $B(t) = 1 - e^{-\mu t}$.
- Batch size is considered unlimited (this assumption although of less practical value facilitates the derivation of the computational algorithm as described by Deb and Serfozo 1973).

Under the above conditions, it is possible to apply the following expressions for a finite number of integers Q to compute the optimum one. In this case, the total expected continuously discounted cost over an infinite horizon will be minimized. The expressions are as follows:

$$\text{If } X <= Q - 1,\ V_\alpha(Q\,; X) = hX/\alpha + h\lambda/\alpha^2 + a^{Q-x}[R + h\lambda/(\alpha+\mu)^2 + cQ + hQ/(\alpha+\mu) + G(Q) - (h/\alpha)(Q + \lambda/\alpha)]$$

otherwise

$$V_\alpha(Q;X) = R + h/\lambda(\alpha+\mu)^2 + cX + hX/(\alpha+\mu) + G(Q),$$

where in any case

$$a = \lambda/(\alpha+\lambda),$$

$$b = \mu/(\alpha+\mu),$$

$$\delta = \lambda/(\alpha+\mu),$$

$$\beta = \lambda/(\alpha+\mu+\lambda),$$

$$A(Q) = R - (hb/\alpha - c)(Q + \delta b\beta^Q),$$

$$B(Q) = 1 - a^Q + (1-b)\beta^Q,$$

$$G(Q) = h\lambda/\alpha^2 + (hb/\alpha - c)Q - R - h\lambda/(\alpha+\mu)^2 + A(Q)/B(Q).$$

The above expressions have been applied to the following example, where $R = 100$, $h = 1$, $c = 1$, $\alpha = 0.30$, $\lambda = 10$ and $\mu = 10$, and it has been found that the

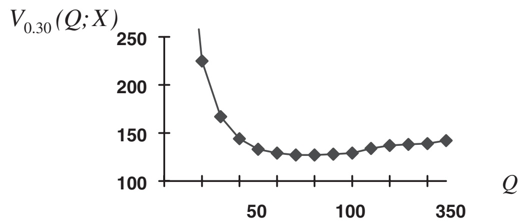


Figure 4. Q versus $V_{0.30}$ $(Q; X)$.

$\min_Q\{V_{0.30}(Q;X)\}=127.4$, and accordingly the optimal control limit of Q is 74. This is shown in figure 4, where the Q-axis depicts the control limit Q, and the V-axis the expected discounted cost $V_\alpha(Q;X)$.

The validity of the above procedure for calculating the optimal policy is subject to two assumptions, one of which is that the system dynamics is stationary (i.e. the mean value, the standard deviation and the type of distribution are fixed). The second assumption is that the system dynamics associated with the inter-arrival times and processing times can both be represented by exponential distributions.

2.5 *System dynamics*

This paper is concerned with developing models and techniques for calculating the optimal limit for non-stationary and non-exponential dynamical systems. Building such models is full of challenge due to the complexity of non-stationary dynamics. Three different types of dynamics are taken into account in the rest of this paper, as summarized in the following three cases.

- Case 1: Distribution is exponential but the mean is not fixed. This is referred to as *parameter dynamics* in this paper.
- Case 2: Distribution is not exponential (e.g. Uniform, Normal, etc.) but it is stationary (or fixed) enough to allow its full perception. This is referred to as *structural* or *distribution dynamics*.
- Case 3: Patterns of distribution are not known and the system dynamics constantly changes. This is called *complex dynamics*.

In the following sections, a new fuzzy scheduling and control system, an on-line perturbation procedure and ANNs are investigated to cope to some extent with parameter dynamics, structure dynamics sand complex dynamics, respectively.

3. New fuzzy scheduling and control system

3.1 *New two-level control system*

For a simple non-stationary system with a fixed distribution but a variable mean arrival λ, the calculation procedure as shown in section 2.4 can be repeated to obtain the corresponding control limit Q^* (figure 5). To cope with such parameter dynamics

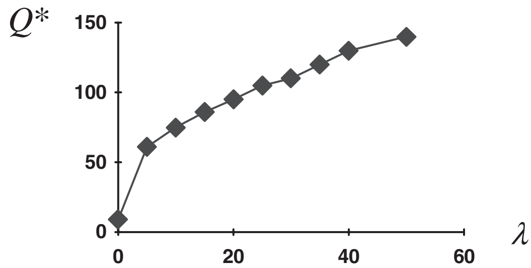


Figure 5. λ versus Q^*.

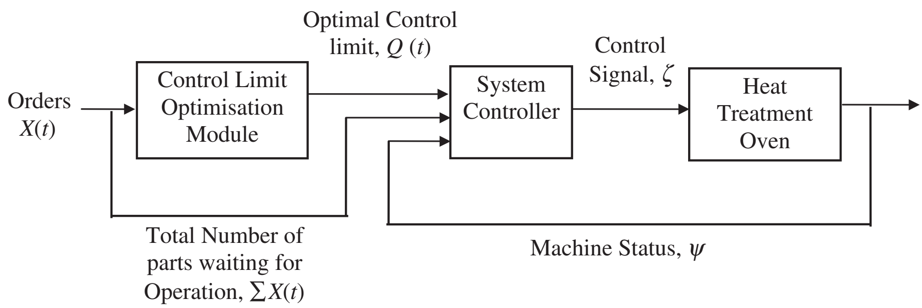


Figure 6. New control structure.

(i.e. the Case 1 dynamics), an adaptive procedure is developed to perceive the changes of λ continuously and then adjust Q^* on-line accordingly.

The model as shown in figure 6 concentrates on the design of a two-level control system, where a new control limit optimization module is added to the original system developed by Deb and Serfozo (1973). This module is used to implement the adaptive procedure for calculating Q^* by detecting the changes of λ. The first box in figure 6 illustrates the Control Limit Optimization Module. This module is producing optimal control limit, $Q(t)$ using real-time data of incoming orders, $X(t)$. The second box illustrates a System Controller (e.g. a PLC). It uses three types of data as input: $\Sigma X(t)$, $Q(t)$ and ψ and produces an output control signal, ζ to govern the heat treatment oven, the third box in figure 6. By eliminating the first box in figure 6, we will have a typical conventional control system.

The control system adopted here is a forward control system in the sense that no feedback is maintained between the controller and the control limit optimization module. This is in accordance with the nature of the problem, where the 'scheduler' attempts to find the optimal control limit policy based on the information acquired from the order data streams to be used by the 'controller'.

3.2 *Mathematical structure of the proposed optimizations module*

The proposed optimization module can be considered as an autonomous input–output function whose input is the stream of parts (or passengers), and whose output is the optimal control limit, as illustrated in figure 7. The sub-functions following the blocks in figure 7 are described by the mathematical equations as follows.

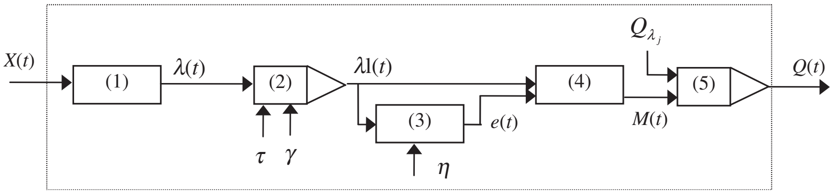


Figure 7. Fuzzy predictive control system.

It can be written that:

$$\lambda(t) = \sum_{k=t_{i-1}}^{t_i} X(k)/(t_i - t_{i-1}) \tag{1}$$

$$\lambda l(t) = \left(\sum_{n=0}^{\tau} \mathrm{e}^{-n\gamma}\right)^{-1} \sum_{n=0}^{\tau} \mathrm{e}^{-n\gamma} \lambda(t-n) \tag{2}$$

$$e(t) = \eta \lfloor \lambda l(t) - \lambda l(t - \Delta t) \rfloor \tag{3}$$

$$M(t) = f(\lambda l(t), e(t)) = \{\mathrm{Poisson}(\lambda_j | \lambda l(t)); j = 1, 2, \ldots, e(t))\tag{4}$$

$$Q(t) = g(Q_{\lambda_j}, M(t)) = \sum_{j=1}^{e(t)} \{Q_{\lambda_j}\}\{\mathrm{Poisson}(\lambda_j|\lambda l(t))\} \sum_{j=1}^{e(t)} \{\mathrm{Poisson}(\lambda_j|\lambda l(t))\}. \tag{5}$$

The above mathematical equations describe the dynamic behaviour of the system. Equation (1) represents a moving average for the incoming parts, within the batches of $i-1$ and i, and equation (2) represents an exponential filtering of the incoming arrival rate over a long period (τ). The reason for adopting an exponential weighted average is that it gives higher weight to recent data. This helps to refresh the population parameter faster, which is an advantage in an intelligent system.

Equation (3) illustrates the uncertainty band (or the error signal) which is an integer corresponding to the difference of the long-term arrival rate, $\lambda l(t)$. When no changes are detected in the arrival pattern of incoming parts, the error signal in equation (3) is equal to zero. This implies that the initial control limit value is optimal. However, if $|e(t)| > 0$, then changes in Q are detected using the fuzzy function given by equation (4), which results in a new control limit value, as calculated by equation (5).

The $M(t)$ in equation (4) is a fuzzy set containing $e(t)$ Poisson probability functions each of which belongs to the set to some extent. When parameter changes take place we know that the distribution is still Poisson but with unknown parameter. The parameter can be any number so that we can well assume that we have a fuzzy set that defines our uncertainty. As the learning proceeds, the fuzzy set becomes non-fuzzy eventually, i.e. a given Poisson distribution with certain λ. The control structure developed in figure 7 is a predictive control system since it predicts the distribution parameter and uses this prediction to control the system optimally.

Deb and Serfozo (1973) developed a set of closed-form expressions for finding an optimal threshold (i.e. as reported in section 2.4) if the system meets the following two conditions: (1) the dynamics of the system can be captured in terms of stationary random variables, where their first and second moments are fixed, and (2) the random variables are exponentially distributed. Real-time stochastic systems, however, hardly comply with these conditions.

The integrated intelligent scheduling and control system as described above relaxes the first condition. A novel adaptive fuzzy control algorithm was developed, in which the fuzzification of the perceived mean of the arrivals (i.e. $\lambda l(t)$) was used to enhance the performance of the predictability of actual λ under the uncertainty involved. The novel feature of this predictive fuzzy control procedure is that probability theory is used to handle both the fuzzification and defuzzification procedure. However, the full account of the fuzzy predictive control system is not the focus of this paper and has been reported elsewhere (Monfared and Steiner 2000). The design of an intelligent control system that relaxes the second condition is now considered as follows.

3.3 *Statement of the problem: structural dynamics*

The system illustrated in figure 6 consists of a machine (e.g. a heat treatment oven), a controller and an optimization module. The controller decides when the machine should be turned on and off (i.e. $\zeta = 1$ and 0). The optimization module, on the other hand, estimates the optimal threshold (i.e. $Q(t)$ or Q^*). The real-time control rule is readily obtained (i.e. $\zeta = 1$ if $\Sigma X(t) \geq Q^*$ and $\zeta = 0$ otherwise, where ζ is the oven control signal, $X(t)$ is the number of parts which have arrived at time t, and ψ is the machine working status).

The controller input variable (i.e. Q^*) was considered fixed by Deb and Serfozo (1973) and is computed once for a given stationary condition. In this paper, Q^* is considered variable to reflect upon non-stationary dynamics in the system's parameter. In the latter case, Q^* is determined adaptively and a priori knowledge regarding the mathematical relationship between Q^* and $X(t)$ and hence λ is required. The data regarding $Q^* - \lambda$ must be computed beforehand, and located in the control algorithm using a conventional method (e.g. look-up tables) or a neural network method. However obtaining this data is theoretically formidable if the variable is not exponentially distributed. It is therefore necessary to design a control algorithm which does not rely on the a priori $Q^* - \lambda$ relationship.

4. On-line perturbation procedure

In the rest of this paper, it is assumed that the stochastic distribution of a system changes from time to time. This is called structural or distribution dynamics (Case 2 dynamics). It is therefore desirable to design an intelligent system, which can perceive such changes of dynamics and then adapt to the changes in real time. It is sensible to develop a procedure that can offer control limit policy irrespective of distributions.

The fact that Deb and Serfozo's expressions cannot cope with general probabilistic patterns is not surprising as the major part of the model was developed using queuing theory, which suffers from a similar difficulty. However, the problem may be handled by designing a learning system. The key point in the development of such an on-line learning procedure is that even if there is no a priori knowledge about the distribution it is still possible to start the control system with an assumed threshold level (i.e. Q_1), with no sense of its optimality. The performance of the system can then be observed and estimated (i.e. V_1).

With the single data so far generated (i.e. $Q_1 - V_1$), it is not possible to verify the optimality of the assumed Q_1. It is, however, possible to disturb Q from the Q_1 by ΔQ, i.e. Q_2 and to observe the differences, which this may cause on cost values, i.e. V_2. The disturbed control limit may either downgrade or upgrade the previous performance, as this will be verified by completion of the experiment and comparison of V_1 and V_2.

The experimental on-line optimization process will eventually lead to exploration of $Q - V$ plane as more data are generated. The optimization procedure conducting the experiments can be a standard technique such as the Golden Section search approach or a modern search technique such as the genetics algorithm (Yang 2000). When the perturbation ΔQ is bounded, the stability and convergence of the system are maintained.

4.1 *Estimation of V*

In accordance with the basic assumption suggested by Deb and Serfozo, we have

$$P(t) = R + cX$$
$$H(t) = h * X$$

and since X is a random variable and c, h and R are fixed parameters, therefore both $P(t)$ and $H(t)$ are also random variables. (Note that $P(t)$ is calculated when a new batch becomes ready for processing, and $H(t)$ is calculated at each time as the holding cost increases at every additional waiting time.)

It is due to this random nature of $P(t)$ and $H(t)$ that it is desirable to estimate the average cost of the system for the perturbation procedure. The estimation of V is conducted as follows:

$$C(t_i) = P(t_i) + H(t_i) \tag{6}$$

$$\sum_{t_i=1}^{t} C(t_i) = \sum_{t_i=1}^{t} P(t_i) + \sum_{t_i=1}^{t} H(t_i) \tag{7}$$

$$\bar{C}(t_i) = \sum_{t_i=1}^{t} C(t_i)/t \tag{8}$$

$$\Delta C(t_i) = \bar{C}(t_i) - \bar{C}(t_i - \Delta t) \tag{9}$$

$$\Delta\,\bar{C}(t_i) = \left[\sum_{t_i=1}^{t} \Delta C(t_i)\right]/t \tag{10}$$

$$V = \bar{C}(T) \text{ when } T = \text{constant} \tag{11}$$

The results shown in figure 9 were obtained using ths simulation results of the system given in Monfared 1997. It is clear that the cost per batch (i.e. $C(t_i) = P(t_i) + H(t_i)$ is a random variable but its average value (i.e. $\bar{C}(t_i) = \sum_{t_i=1}^{t} = C(t_i)/t$) continuously improves over time and get stabilized (note the thick line in figure 8).

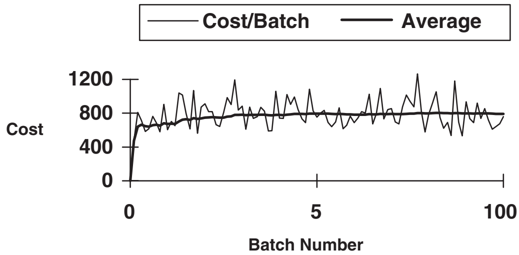


Figure 8. Cost per batch ($C(t_i)$) and average cost per batch ($\bar{C}(t_i)$).

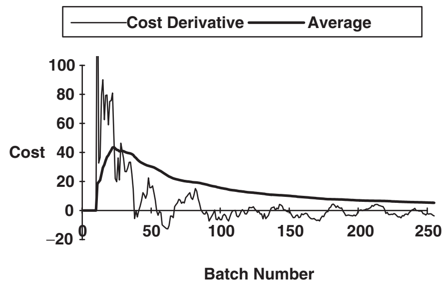


Figure 9. Increment of average cost per batch ($\Delta C(t)$) and its smoothed value ($\Delta\bar{C}(t)$).

However, it is necessary to check when the system becomes stabilized as it is at that point that V is estimated by $\bar{C}(t_i)$. For this purpose, it seems appropriate to use the increment of the average cost function, or $\Delta C(t_i) = \bar{C}(t_i) - \bar{C}(t_i - \Delta t)$. An important property of $\Delta C(t_i)$ is that it gradually reduces to near zero. To use this property better, a filter has been used which is defined by $\Delta\bar{C}(t_i) = [\sum_{t_i=1}^{t} \Delta C(t_i)]/t$. The behaviour of $\Delta C(t_i)$ and its filtered version i.e. $\Delta\bar{C}(t_i)$) are shown in figure 9.

A necessary condition that the system is stabilized is defined as follows:

$$\Delta\bar{C}(t_i) < \Delta\bar{C}(t_i - 1)$$

which implies that $\Delta\bar{C}(t)$ approaches zero if enough time elapses, or

$$\lim[\Delta\bar{C}(t)] = 0 \text{ if } t \to \infty.$$

The above condition does not indicate where to stop the process. The number of the batches needed to obtain V depends upon the system under consideration. For the given system, it has been experimentally found that the system can be relatively well stabilized after 300 batches of productions. Note, however, that to automate the working condition of this system, an adaptive mechanism such as the one developed in section 3.1 can be adopted. Such an adaptive mechanism is obviously error based. Here, again we have assumed that when structural dynamics take place it is not short lived. It elapses enough time to enable our perturbation module to capture its dynamics. We relax this limitation in section 5 by adopting a more intelligent control structure using neural networks.

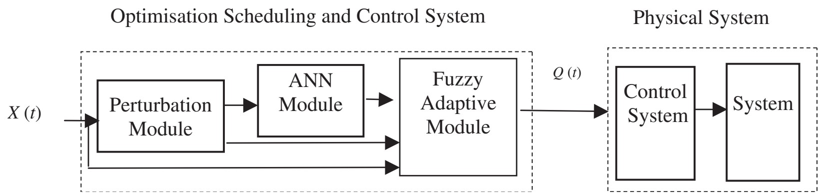


Figure 10. Integrated intelligent scheduling and control system architecture.

5. Neural networks for capturing complex dynamics

Structural dynamics such as changes in probability distribution are usually unknown a priori. The system must be able to detect such changes automatically and activate the perturbation procedure to extract data required for optimal control. One way to detect a change in distribution is to monitor some of its statistical measures (e.g. mean, variance) and then compare the measures with those of certain known probability distributions to establish whether a new change has taken place. Once a new change is detected, then the perturbation procedure is automatically activated for generating a new control limit Q^*.

Results obtained using the above perturbation procedure are needed for controlling the system. However, the implementation of the on-line perturbation procedure causes disturbance to the system, which is undesirable. It is, therefore, preferable if the invocation of the procedure can be minimized. Furthermore, to apply the perturbation procedure effectively, the change caused by the perturbation should last long enough to reach a stable state. In an unstable environment where the change cannot stabilize, a well-trained ANN can take over to provide an effective control policy. This is where Case 3 complex dynamics take place. In this section, the development of an ANN module will be investigated. This module needs to be integrated in an architecture proposed in figure 10 to produce an integrated intelligent scheduling and control system.

In this multilevel architecture, a new Optimization Scheduling and Control System is developed. The input to this system is $X(t)$, i.e. the stream of real-time part arrivals. This system provides control limit $Q(t)$ as output that feeds into Physical System. All cases of dynamics are accommodated within this system.

The Fuzzy Adaptive Module as illustrated in section 3 accommodates Case 1 parameter dynamics. The Perturbation Module accommodates Case 2 structural dynamics as illustrated in section 4. Case 3 complex dynamics is hence accommodated by the ANN Module using data from the Perturbation Module in a way that is discussed here.

5.1 *Building and working of ANN module*

To illustrate how ANNs can work within the proposed architecture, different models were examined. The simplest form of ANN associates the mean arrival rate λ to the optimal control limit values Q^*. If only Case 1 parameter dynamics were involved, this simple model was appropriate. However, in a non-stationary stochastical system where complex dynamics are involved, more elaborate and complex ANN models are to be developed.

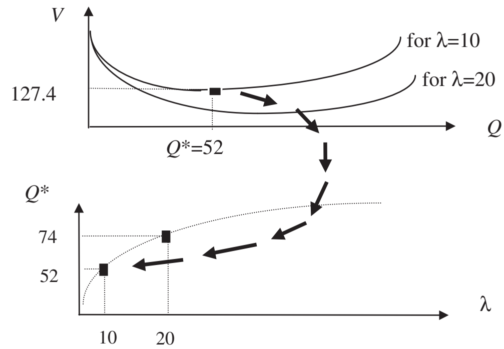


Figure 11. Illustration of linking the perturbation procedure to an artificial neural network.

Table 2. Training data for two sets of experiments.

	First experiment			Second experiment		
Mean arrival	a	b	$\sigma=(b-a)/\sqrt{12}$	a	b	$\sigma=(b-a)/\sqrt{12}$
10	5	15	2.89	0	20	5.78
20	15	25	2.89	0	40	11.56
30	25	35	2.89	0	60	17.34
40	35	45	2.89	0	80	23.12

Training data for ANN obtained by the perturbation procedure can be an on-line process, as described below. Every time the perturbation procedure is activated for a given arrival rate λ, it produces an optimal control limit Q^*. For the system described above, for example, $Q^*=52$ for $\lambda=10$, which makes one point on the $Q^*-\lambda$ plane (figure 11). In this way, when a number of such points are generated the $Q^*-\lambda$ curve is further developed. The ANN is well suited to model any non-linear function and can improve the function with the accumulation of knowledge.

To investigate the possible impact of complex dynamics on control limits, two sets of experiments are conducted on a three-layer feed-forward neural network trained by a back propagation algorithm. In both experiments, the arrival pattern is changed from exponential distribution to uniform distribution and the mean arrival rates are kept identical for comparing the results. Besides, service time is assumed to be exponentially distributed with the mean time of 10 and the mean arrival rates are changed from 10 to 40. However, different standard deviations σ of the uniform functions are used in the two experiments as shown in table 2.

We developed a C++ program to simulate the production system. The results of the two experiments are illustrated in figure 12 where the optimal control limits are obtained using the perturbation procedure described in section IV. The base $Q^*-\lambda$ curve is based on the exponential arrival distribution. The $Q^*-\lambda$ curves generated from the two experiments are both significantly different from the base curve. Note that the $Q^*-\lambda$ curve obtained from the second experiment is no longer monotonic but illustrates non-linearity. It is then evident that dynamics stems from changes

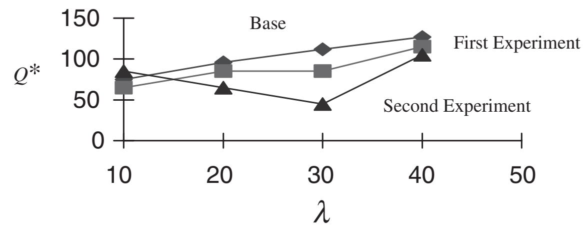


Figure 12. Effect of distribution dynamics on the optimal control limit.

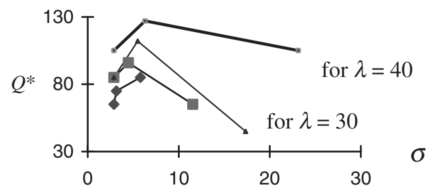


Figure 13. Non-linear impact of standard deviation on the optimal control limit ($Q^* - \sigma$).

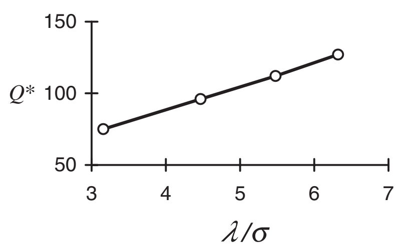


Figure 14. Q^* versus λ/σ for an exponential distribution.

in distribution function and its standard deviation affects our optimal control limit optimal policy.

To define input variables for our ANN that can capture this non-linearity, we report the results obtained from simulation for Q^* versus σ (figure 13). Even for the same arrival distribution function and fixed mean arrival rate, the optimal control limit (Q^*) will vary when the standard deviation is different. In this illustration, $Q^* - \sigma$ are obtained for different exponential mean arrival rates (i.e. for $\lambda = 10$, 20, 30 and 40).

Though results of the experiments undertaken so far implied that both mean arrival rate and standard deviation have direct impact on the optimal value of control limit in a non-linear way, it is quite possible to define a new input variable obtained by dividing mean value to standard deviation and achieve nearly linear relationship between Q^* and λ/σ (figure 14). This is, however, limited to only exponential arrivals. The $Q^* - \lambda/\sigma$ curve becomes non-linear for uniform arrivals (figure 15). For the combination of exponential and uniform arrivals, Q^* is a strong non-linear function of λ/σ (figure 16). It is clear from these experimental results that the optimal control limit Q^* is strongly dependent upon the variable λ/σ.

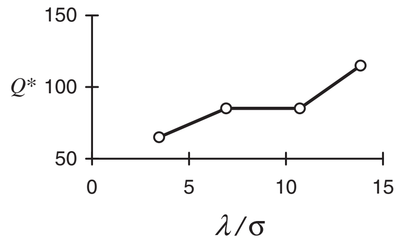


Figure 15. Q^* versus λ/σ for a uniform distribution.

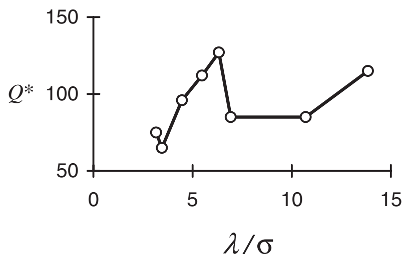


Figure 16. Q^* versus λ/σ for a combination of exponential and uniform distributions.

Table 3. Introducing new input variables.

λ	Distribution	λ/σ	σ	$\sigma/\sigma_{\mathrm{Exp}}$	N	N/λ	Q^*
10	Exp(10)	3.16	3.16	1	124	12.4	75
	Uniform(5, 15)	3.46	2.89	0.91	119	11.9	65
	Uniform(0, 20)	1.73	5.77	1.83	128	12.8	85
20	Exp(20)	4.47	4.47	1	230	11.5	96
	Uniform(15, 25)	6.92	2.89	0.65	206	10.3	85
	Uniform(0, 40)	1.73	11.55	2.58	200	10.0	$\leq$65
30	Exp(30)	5.48	5.48	1	319	10.6	112
	Uniform(25, 35)	10.71	2.89	0.53	294	9.8	85
	Uniform(0, 60)	1.73	17.34	3.16	326	10.9	45
40	Exp(40)	6.32	6.32	1	423	10.6	127
	Uniform(35, 45)	13.84	2.89	0.46	395	9.9	115
	Uniform(0, 80)	1.73	23.12	3.66	431	11.0	105

Experimental results illustrate that the optimal control limit Q^* does not necessarily reflect upon the number of parts served, which is accumulated before a service is started. This is because a service is activated when at least Q^* parts are accumulated. Hence, Q^* is only the lower limit of the number of parts served. N may vary depending upon the system state variables. In table 3, for a given mean arrival rate, the average number of parts served varies under different distribution functions. The average number of parts per service is therefore selected as another input variable. It is normalized by λ to compensate for the effect of mean arrival rate. Figure 17 shows the $N/\lambda - \lambda$ curves for different arrival distribution patterns.

It can also be seen from table 3 that for a given λ, the standard deviation normalized by an exponential standard deviation (or $\sigma/\sigma_{\mathrm{Exp}}$) significantly changes with different distribution patterns. This normalized standard deviation is thus used as

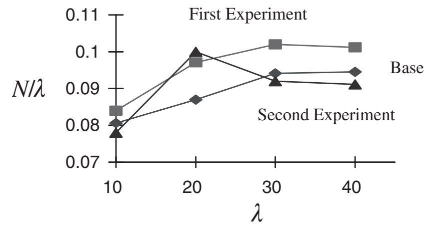


Figure 17. Illustration of N/λ versus λ.

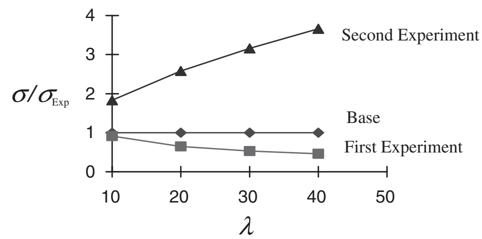


Figure 18. Illustration of $\sigma/\sigma_{\mathrm{Exp}}$ versus λ.

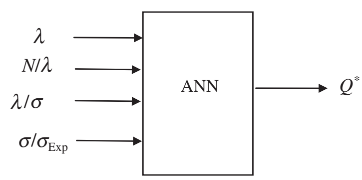


Figure 19. Illustration of the artificial neural network module.

another input variable. The $\lambda-\sigma/\sigma_{\mathrm{Exp}}$ for different distribution patterns are shown in figure 18.

Using above analysis, an ANN model is constructed (figure 19). Input variables are selected to include: mean arrival rate (λ), the average number of parts served denoted by N normalized by λ (N/λ), mean arrival rate λ normalized by the standard deviation $\sigma(\lambda/\sigma)$, and the standard deviation σ normalized by the exponential standard deviation σ_{Exp} (or $\sigma/\sigma_{\mathrm{Exp}}$). The relationships between these input variables and Q^* are investigated using the perturbation procedure as described above by using our simulation package.

5.2 *Testing the ANN module*

The results obtained are listed in table 4. Note that the range of input data is extended to all three distributions including an exponential distribution (i.e. Exp( )) and two uniform distributions (i.e. Unif_1 ( ) and Unif_2 ( )). One can see from table 4 that the difference between the simulation results and the optimal results is very small. This indicates that the candidate input variables can satisfactorily capture the dynamics of the existing system.

Two tests were conducted. In the first, new input data within the range of the training data (i.e. for λ between 10 and 40) are considered and in the second new data outside the range (i.e. for $\lambda > 40$) are considered. The experiment results as given

Table 4. Results of an ANN model.

Distribution	Optimal limit, Q^*	Network output, Q_{ANN}	Error, $Q^* - Q_{\mathrm{ANN}}$
Exp(10)	75	72	3
Uniform(5, 15)	65	70	−5
Uniform(0, 20)	85	88	−3
Exp(20)	96	92	4
Uniform(15, 25)	85	80	5
Uniform(0, 40)	≤65	65	0
Exp(30)	112	109	3
Uniform(25, 35)	85	93	−8
Uniform(0, 60)	45	45	0
Exp(40)	127	129	−2
Uniform(35, 45)	115	111	4
Uniform(0, 80)	105	105	0
	Total error, $\Sigma\lvert Q^* - Q_{\mathrm{ANN}}\rvert$		37

Table 5. Validating the ANN with new data.

Test number	Distribution	λ	λ/σ	$\sigma/\sigma_{\mathrm{Exp}}$	N/λ	Q^*	Q_{ANN}
1	Exp(15)	15	3.87	1.00	13.00	86	82
	Normal(15, 5)	15	3.00	1.29	16.67	60	48
	Fixed(15)	15	∞	0.00	12.73	115	73
2	Exp(45)	45	6.71	1	11.13	133	141
	Exp(50)	50	7.07	1	11.20	139	151
	Exp(55)	55	7.41	1	11.42	145	156

in table 5 illustrate that ANN models can work reasonably well in respect to both interpolation (i.e. test number one) and extrapolation (i.e. test number two).

5.3 *Further integration for improving the system intelligence*

Through a step-wise modelling procedure, as proceeded above, we have shown that ANN can capture and represent the complex dynamics of the system, when appropriate input variables are selected. This provides the possibility of integrating the ANN technology with the perturbation procedure in order to increase the intelligence of the system. This integration is mutually beneficial to both ANN module and the Perturbation module.

This is because the ANN can produce a good initial point for the perturbation procedure, and hence can accelerate its convergence. On the other hand, the perturbation produces new data, which can then be used to retrain the network. As time goes by, new data using a perturbation module will be generated and can be fed into the ANN.

Hence, the ANN will become capable of representing the real system dynamics. This in turn will improve the generalization capability of the ANN, and will become more intelligent in facing with new situations. When enough time has passed, the ANN can rightly cope with more (even unseen) dynamics. In this way, the need for applying an expensive Perturbation module will be reduced, and eventually the Perturbation module may be practically eliminated.

6. Integration methodology

The design of intelligent manufacturing scheduling and control system proposed in the present paper integrates on-line adaptation and optimization in a non-stationary stochastic environment. It provides scheduling and control actions in an integrated hierarchical approach. This section explains the proposed integration methodology.

The system consisted of different components in a multilevel architecture (figure 20). The levels in the hierarchy are chosen to supervise appropriate scheduling and control actions. In this sense, the lowest level illustrates the classical control system with no parameter dynamics, i.e. a PLC controller.

The second level treats Case 1 parameter dynamics using a novel fuzzy predictive system, as illustrated in the third level of figure 20. This system senses the changes in mean arrival rate and acts on it accordingly. This module is normally active unless the command is taken over by the upper level.

A perturbation mechanism is embedded in the third level to implement on-line optimization for Case 2 structural changes. When it is active, it takes over the command line and sets the $Q(t)$ directly on the PLC controller. The activation and deactivation of the Perturbation module depend on error signals coming from multiple fuzzy predictive mechanisms (figure 20).

The possibility of sensing and measuring structural dynamics, using similar mechanisms as in the case of mean arrival rate, is based on the fact that a probability

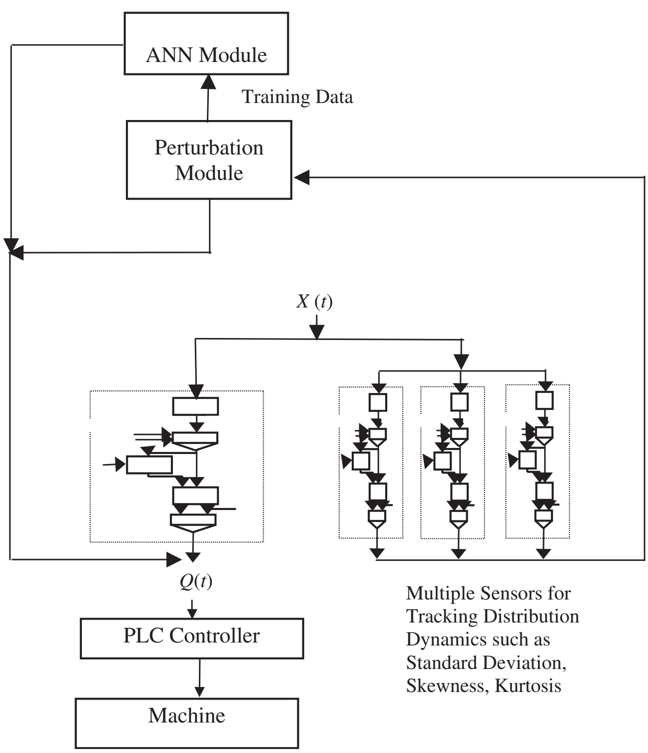


Figure 20. Integrated multilevel manufacturing scheduling and control system.

distribution function can be detected by its statistical moments (e.g. standard deviation, skewness, kurtosis). All information required is then only contained in the stream of $X(t)$ data. For example, if the distribution is changing from exponential to normal, the error signals become non-zero and this leads to activation of the Perturbation module. The full investigation of this system, however, is beyond the scope of this paper.

The final level of the approach is composed of ANN that can learn from experiences provided by the perturbation mechanism. The training phase of ANN will be activated when new dynamics has been detected and implemented by the Perturbation module. The working phase of ANN is activated when enough learning takes place. In this case, the control value of $Q(t)$ will be set only by ANN. It should be added that the ANN module can be pretrained using simulation experiments, as discussed in section 6. In this case, the need to activate the Perturbation module will be minimized, which is desirable.

7. Concluding remarks

In a conventional approach, systems are designed with complete and precise a priori knowledge for which the chance for on-line adaptation and optimization is almost insignificant (or is left to be maintained through human interventions). A unique feature of biological intelligent systems is that they build and improve over their communication, decision-making, and control structures in real time and autonomously. Research and development for integrating such interesting features into engineering systems are of recent interest. This paper proposed a new approach for the design of manufacturing scheduling and control systems that can illustrate both on-line adaptation and optimization properties.

This approach is of multidisciplinary nature where concepts and techniques from operations research, control theory, fuzzy logic and neural networks were integrated. Using this approach, a flow shop manufacturing scheduling and control system has been successfully modelled in order to work autonomously and intelligently. The system is autonomous because it does not require human intervention in performing scheduling and control tasks. The system is intelligent because it can successfully encounter various forms of dynamics (or uncertainty).

The key element in this approach is the integration of new methodologies such as fuzzy logic and ANNs into the classical methodologies of operations research and control theory. This provides capacity for adding on-line adaptation and optimization to scheduling and control systems, which currently suffer from the lack of intelligence and autonomy. Hence, the integrated approach is worth exploiting for other manufacturing systems (e.g. job shop production lines). It can be expected that through such special case analyses and modelling efforts, a new theory can eventually emerge to address the systematic design of intelligent manufacturing scheduling and control systems.

The integrated approach developed herein can also be applied to other non-stationary stochastical systems such as mailing systems, advanced library systems, traffic control systems where intelligent and autonomous real-time scheduling and control is of immediate concern (see Vasquez-Marquez 1991, Aubin 1992, Nogami *et al.* 1996).

A.1. Appendix 1: Notation

- c variable cost of producing a single unit,
- h holding cost of a single unit per time,
- i batch number,
- j counter to be used in the fuzzification procedure,
- $e(t)$ uncertainty band (or the error signal), which is an integer,
- t current time,
- t_i period number in batch i,
- $C(t)$ total operating cost per production batch,
- $\bar{C}(t)$ average cost so far,
- $H(t)$ total holding cost,
- $M(t)$ membership function, or the fuzzification of $\lambda l(t)$ with respect to the uncertainty band of $e(t)$,
- $P(t)$ total production cost,
- Q^* optimal control limit for a stationary system,
- $Q(t)$ optimal control limit at time t for a non-stationary system,
- R fixed cost of producing a single batch,
- V estimated average cost for adopting control limit (Q) when the system is stable,
- X total number of items waiting for processing (i.e. $\Sigma X(t)$),
- $X(t)$ number of orders (parts, jobs or passengers) that arrive at time t,
- mean arrival rate for any distribution (i.e. denoted as λ in the case of exponential distribution),
- $\Delta C(t)$ average cost difference,
- $\Delta \bar{C}(t)$ second-order difference on average cost,
- α discount rate of incurring cost,
- λ assumed mean arrival rate of a Poisson distribution,
- $\lambda(t)$ actual estimate of the mean arrival rate at time t,
- $\lambda l(t)$ long-term estimate of the mean arrival rate, i.e. to be considered as an approximation for the actual parameter λ,
- μ mean processing time of a negative exponential distribution (i.e. $B(t) = 1 - e^{-\mu t}$),
- γ constant defining the learning rate of the actual parameter (i.e. λ) of the distribution,
- η adjustment multiplier, which is an integer,
- ζ control signal, which is 0 or 1 (i.e. do nothing or start a new operation),
- τ windowing time, i.e. time over which the moving average is computed,
- ψ machine status, which is 0 or 1 (i.e. off or on),
- σ standard deviation of the arrivals.

References

Abdelhameed, M.M. and Tolbah, F.A., Design and implementation of a flexible manufacturing control system using neural network. *Int. J. Flex. Manuf. Sys.*, 2002, **14**(3), 263–279.

Akturk, M.S. and Ozkan, S., Integrated scheduling and tool management in flexible manufacturing systems. *Int. J. Production Research*, 2001, **39**(12), 2697–2722.

Antsaklis, P., Defining intelligent control. *IEEE Contr. Sys.*, 1994, **4–5**, 58–67.
Askin, R.G. and Standridge, C.R., *Modelling and Analysis of Manufacturing Systems*, 1993 (Wiley: USA).
Aubin, J., Scheduling ambulances. *Interfaces*, 1992, **22**(2), 1–10.
Bodington, E., *Planning, Scheduling and Control integration in the Process Industries*, 1995 (McGraw-Hill: New York).
Buzacott, J.A. and Shanthikumar, J.G., *Stochastic Models of Manufacturing Systems*, 1993 (Prentice-Hall: USA).
Cervin, A., Eker, J., Bernhardsson, B. and Crzén, K., Feedback–feedforward scheduling of control tasks. *Real-Time Sys.*, 2002, **23**(1), 25–53.
Chen, F.F., Wu, D. and Henry, R.R., Intelligent scheduling and control of flexible manufacturing systems, In Proceedings of the 1996 24th NAMRC Conference, Ann Arbor, MI, USA, 21–23 May 1996.
Deb, R.K. and Serfozo, R.F., Optimal control of batch service queues. *Adv. Appl. Prob.*, 1973, **5**, 340–361.
Diaz, A., Sancho, L., Garcia, R. and Larraneta, J., A dynamic scheduling and control system in an ENSIDESA Steel Plant. *Interfaces*, 1991, **21**(5), 53–62.
Gershwin, S.B., *Manufacturing Systems Engineering*, 1994 (PTR Prentice-Hall: USA).
Ho, Y.C., Dynamics of discrete event systems. *Proc. IEEE*, 1989, **77**(1), 3–6.
Kogan, K. and Khmelnitsky, E., *Scheduling, Control-based Theory and Polynomial-Time Algorithms*, 2000 (Kluwer, Dordrecht).
Kumar, R.A. and Jaikumar, R., An interactive decision support system for on-line process control. *Eur. J. Oper. Res.*, 2002, **138**(3), 554–568.
Lu, C., Feedback control real-time scheduling. PhD thesis, University of Virginia, 2001.
Monfared, M.A. S., A study of control algorithms for some classes of intelligent manufacturing systems. PhD dissertation, School of Manufacturing and Mechanical Engineering, University of Birmingham, 1997.
Monfared, M.A. S. and Steiner, S.J., Fuzzy adaptive scheduling and control systems. *Fuzzy Sets Sys.*, 2000, **115**, 231–246.
Nogami, J., Nakasuka, S. and Tanabe, T., Real-time decision support for air traffic management utilising machine learning. *Contr. Eng. Pract.*, 1996, **4**(8), 1129–1141.
Prabhu, V.V. and Duffie, N.A., Nonlinear dynamics in distributed arrival time control of heterarchical manufacturing systems. *IEEE Trans. Contr.*, 1999, **T7**(6), 724–730.
Ramadge, P.J. and Wonham, W.M., Supervising control of a class of discrete event processes. *SIAM J. Contr. Optim.*, 1987, **25**, 206–223.
Suri, R., Perturbation analysis: the state of the art on research issues explored via the G/G/1 queue. *Proc. IEEE*, 1989, 114–137.
Trentesaux, D., Pesin, P. and Tahan, C., Distributed artificial intelligence for FMS scheduling, control and design support. *J. Intell. Manuf.*, 2000, **11**(6), 573–589.
Vasquez-Marquez, A., American airline Arrival Slot Allocation System (ASAS). *Interfaces*, 1991, **21**(1), 42–61.
Werbos, P., On the analytic formulation of intelligent controls. *IEEE Contr. Sys.*, 1994, 65–66.
Yang, J.B., GA-based discrete dynamic programming approach for scheduling in FMS environments. *IEEE Trans. Sys. Man Cyber. — Part B: Cybernet*, 2001, **31**(5), 824–835.