International Journal of Production Research
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713696255

Design of predictable production scheduling model using control theoretic approach

Sohyung Cho [a]; Murat Erkoc [a]
[a] Department of Industrial Engineering, University of Miami, Coral Gables, FL 33146, USA

First Published on: 06 February 2008
To cite this Article: Cho, Sohyung and Erkoc, Murat (2008) 'Design of predictable production scheduling model using control theoretic approach', International Journal of Production Research, 1 - 19
To link to this article: DOI: 10.1080/00207540701749281
URL: http://dx.doi.org/10.1080/00207540701749281

International Journal of Production Research
2008, 1–19, iFirst

Design of predictable production scheduling model using control theoretic approach

Sohyung Cho* and Murat Erkoc

Department of Industrial Engineering, University of Miami, 268 McArthur Building, 1251 Memorial Drive, Coral Gables, FL 33146, USA

(*Received 23 February 2007; final version received 24 September 2007*)

As one of the most important planning and operational issues in manufacturing systems, production scheduling generally deals with allocating a set of resources over time to perform a set of tasks. Recently, control theoretic approaches based on nonlinear dynamics of continuous variables have been proposed to solve production scheduling problems as an alternative to traditional production scheduling methods that deal with decision-making components in discrete nature. The major goal of this paper is to improve predictability and performance of an existing scheduling model that employs the control theoretic approach, called distributed arrival time controller (DATC), to manage arrival times of parts using an integral controller. In this paper, we first review and investigate unique dynamic characteristics of the DATC in regards to convergence and chattering of arrival times. We then propose a new arrival time controller for the DATC that can improve predictability and performance in production scheduling. We call the new mechanism the double integral arrival-time controller (DIAC). We analyse unique characteristics of the DIAC such as oscillatory trajectory of arrival times, their oscillation frequency, and sequence visiting mechanism. In addition, we compare scheduling performance of the DIAC to the existing DATC model through computational experiments. The results show that the proposed system can be used as a mathematical and simulation model for designing adaptable manufacturing systems in the future.

Keywords: production scheduling; control theory; nonlinear dynamics; double integral control

1. Introduction

Manufacturing systems organise equipment, processes, and information to produce the desired products for customers via planning and control. In fact, enhanced co-ordination to increase productivity with minimised operating costs can be achieved by effective planning and control of production activities. As one of the most important planning and operational issues in the manufacturing systems, scheduling generally deals with allocating a set of resources over time to perform a set of tasks. Efficient scheduling helps managers and supervisors co-ordinate activities to increase productivity and reduce operating costs.

*Corresponding author. Email: scho@miami.edu

ISSN 0020–7543 print/ISSN 1366–588X online

DOI: 10.1080/00207540701749281
http://www.informaworld.com

For example, a major steel company in Japan has saved over a million dollars a year using a scheduling system developed by IBM (Luh and Hoitomt 1993). Enhanced scheduling (by even a small percentage of performance improvement) of space shuttle ground processing at NASA, where more than 1500 tasks need to be scheduled over 120 days has resulted in millions of dollars savings (Zweben *et al.* 1993). Therefore, a vast amount of research about scheduling has been conducted in more than 20,000 articles published over the past several decades (McKay and Wiers 1999).

Although the majority of the articles have focused on scheduling optimisation, academia and industry acknowledge that conventional scheduling optimisation is only one aspect of scheduling. In fact, scheduling systems should not only be able to generate high-quality schedules but also respond quickly to changes in the manufacturing systems and environment (Sabuncuoglu and Karabuk 1999). For instance, one of the key issues in job shop scheduling is to provide decidability (predictability) because it is extremely difficult to predict the system behaviour due to frequent changes in the system and environment (Parunak 1991). The notion of predictability can be described in terms of anticipation of the changes in job completion times on the occurrence of disruptions. Hence, it is a challenge to develop a predictable scheduling system that can ensure improved visibility of manufacturing systems in terms of dynamic and performance behaviour that eventually leads to improved productivity and increased customer satisfaction. Such a predictable scheduling system with improved performance can mainly be applied to production scheduling, where traditional dispatching policies are preferred in spite of their poor performance, because their behaviour is more predictable than that of the black-box approach such as optimal scheduling algorithms.

In the past, control theoretic techniques were used for predictable production scheduling of manufacturing systems based on hierarchical control architecture that deals with high-volume low-variety production (Caramanis and Sharifnia 1991). Recently, auction-based scheduling schemes, where heuristics select various dispatching policies depending on system status, have been developed to provide enhanced predictability in manufacturing systems based on heterarchical control architecture (Veeramani and Wang 1997). More recently, a highly distributed feedback control approach using a continuous arrival time controller, that replaces decision-making heuristics in traditional scheduling algorithms, has been proposed (Prabhu and Duffie 1999, Cho and Prabhu 2002a, b). The latter approach is known as the distributed arrival time controller (DATC). The DATC approach transforms the combinatorial scheduling problem into a dynamic control problem of continuous variables in the vector space. Although extensive analysis has been conducted for understanding dynamics and performance characteristics of the DATC, a more rigorous analysis is still required to satisfy diverse needs in different industrial sectors and space/military missions where a swift and precise decision-making is critical.

In this paper we provide a review of working principles of the DATC using a simple example. Specifically, in Section 2, we explain and illustrate with examples how an embedded integral controller adjusts part arrival times to minimise the deviation of completion time from due date. In Section 3, we discuss the unique dynamic characteristics of the DATC such as convergence and chattering of arrival times. We then identify limitations of the DATC on predictability and performance in scheduling. In Section 4, we propose the design of a new controller for the existing DATC to improve predictability and performance in scheduling. We call this new controller double integral arrival-time controller (DIAC). Next, we analyse unique characteristics of the DIAC such as oscillatory trajectory of arrival times, their oscillation frequency, and sequence visiting mechanism. In addition, the scheduling performance of the DIAC is compared with the existing DATC model through computational experiment.

2. Overview of DATC

Exact solution methods such as branch and bound (Bagchi *et al.* 1987), dynamic programming (Abdul-Razaq and Potts 1988), and Lagrangian relaxation (Li 1997) generate optimal schedules, but they are computationally so inefficient in general that they are not recommended for scheduling of adaptable manufacturing systems under dynamically changing conditions in practice. On the other hand, scheduling methods using heuristics or meta-heuristics techniques such as simulated annealing (Aarts *et al.* 1986), tabu search (Glover 1986), and genetic algorithm (Goldberg 1989) generate acceptable (near-optimal) schedules with less computational effort than the optimisation approach, but it is difficult to rigorously predict their behaviour and use them dynamically. For wider acceptance in real practice, scheduling tools must incorporate heuristics that deal with prevailing uncertainty or changes such as machine failures and demand fluctuations in a cost-effective manner.

Recognising a need for a computationally fast and dynamically predictable scheduling approach for adaptable manufacturing systems, Prabhu and Duffie (1999) introduced the distributed arrival time controller (DATC). The DATC utilises a feedback control structure and adjusts arrival times of parts iteratively using feedback information from shop floor. More specifically, the DATC starts its first iteration with the release of parts into the shop floor. In the DATC, the timing of parts release into the shop floor is defined as arrival time. Its shop floor simulation module computes expected completion times of parts using the arrival times and processing times information. Note that the expected completion time is simulated by the part processing sequence based on first-come first-served (FCFS) dispatching policy, which is determined by the arrival times. Then, the integral controller, which is embedded in the part, computes the deviations of the expected completion time from the due date and adjusts the arrival time based on the accumulated deviations. The arrival times adjusted by the integral controllers are then released again into the shop floor simulation for next iteration. At the end of the simulation within a given number of iterations, various schedules based on different arrival times and their relative orders are generated. The DATC provides the best schedule as an output. In this paper, we limit the best schedule to the one that gives the minimum of mean squared deviations of completion times from due dates (MSD).

Working principle of the DATC can be efficiently explained using an example illustrated in Figure 1, where two parts are needed to be scheduled in a single machine. Even though only two parts are shown in this example, there is no limit to the number of parts to be scheduled. Also, note that the same working principle is applied for more general type of shop floors such as flow shops and job shops.

In the figure, discrete time (iteration) is denoted by t. Arrival time, processing time, complete time, due date and deviation of completion time about due date of ith part at time t are represented by $a_i(t)$, $p_i(t)$, $c_i(t)$, $d_i(t)$ and $z_i(t)$, respectively. In the DATC where an integral controller is embedded in each part, arrival time of ith part in discrete time

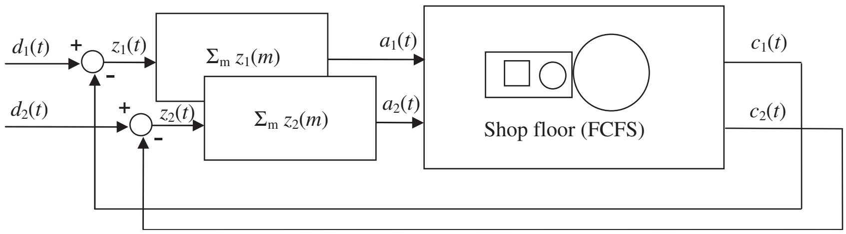


Figure 1. Feedback control structure of DATC.

domain can be expressed as:

$$a_i(t) = k_i \sum_{m=0}^{t-1} [d_i(m) - c_i(m)] + a_i(0) = k_i \sum_{m=0}^{t-1} z_i(m) + a_i(0) \tag{1}$$

where k_i is control gain for ith part. The trajectory of arrival time in continuous time domain is expressed as:

$$a_i(t) = k_i \int_0^t [d_i(\tau) - c_i(\tau)]d\tau + a_i(0) = k_i \int_0^t z_i(\tau)d\tau + a_i(0) \tag{2}$$

In Equations (1) and (2), an integral controller adjusts arrival time to make completion time as close to due date as possible and thus the scheduling objective in this paper is to minimise the MSD that can be expressed as:

$$\text{MSD} = \frac{\sum_i (d_i - c_i)^2}{n} \tag{3}$$

where both earliness and tardiness of processing completion are penalised. Minimising MSD is important in just-in-time (JIT) production where the scheduling objective is to reduce work-in-process (WIP) and increase customer satisfaction, simultaneously. It must be pointed out here that the integral controller in the DATC functions as a search engine replacing the heuristics used in the traditional approach. In the implementation of the DATC, the integral controllers are distributed on part entities and computation of deviations and adjustment of arrival times in part entities takes place with limited global information.

As a numerical example, suppose two parts are ready for processing at time unit 0 and 0.1 so that initial arrival times are set as $\mathbf{a}(0) = [0.0 \quad 0.1]$. Part processing times are given as $\mathbf{p} = [1 \quad 3]$ and due dates are given as common as $\mathbf{d} = [10 \quad 10]$. In the first iteration, processing sequence is determined as $\boldsymbol{\pi}(0) = [1 \quad 2]$ by FCFS dispatching policy employed in the resource at shop floor. With this processing sequence, the shop floor simulation computes the expected completion times as $\mathbf{c}(0) = [1 \quad 4]$. The expected completion times are then compared with the due dates resulting in due date deviations as $\mathbf{z}(0) = \mathbf{d}(0) - \mathbf{c}(0) = [9 \quad 6]$. The embedded integral controller adjusts the arrival times for next iteration resulting in

$$\mathbf{a}(1) = k \cdot \mathbf{z}(0) + \mathbf{a}(0) = [0.9 \quad 0.6] + [0.0 \quad 0.1] = [0.9 \quad 0.7].$$

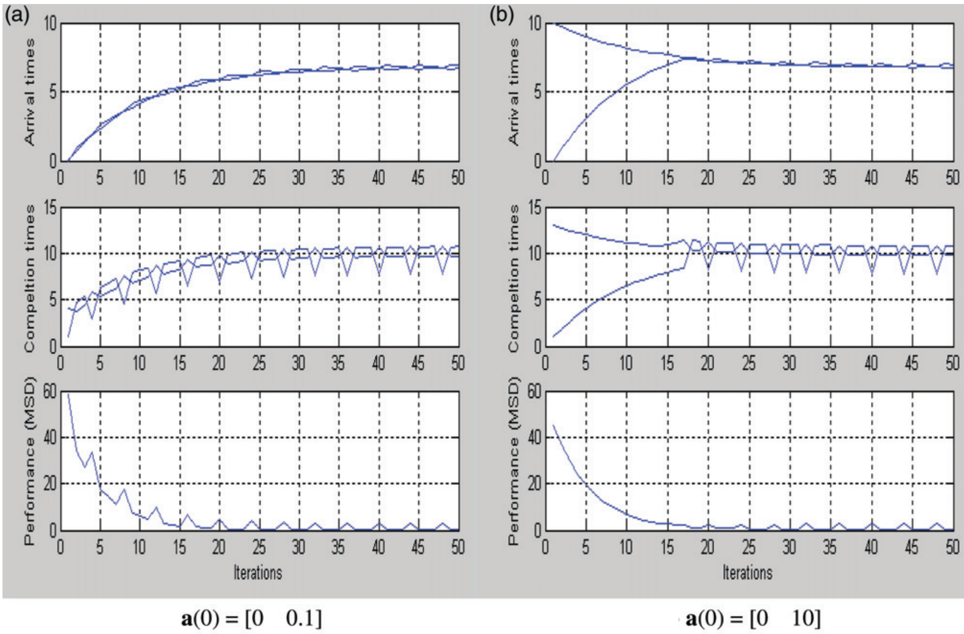
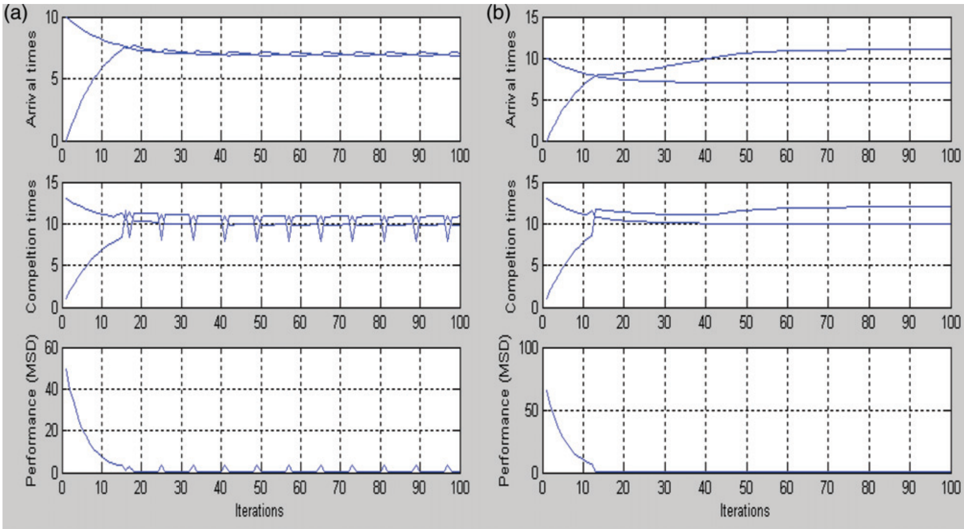


Figure 2. Arrival time, completion time and scheduling performance trajectory of DATC for two-part in a single machine problem (common due date).

Note that identical control gain of $k = 0.1$ is assumed here. After the adjusted arrival times are released into shop floor simulation, the same steps repeat iteratively resulting in the arrival time, completion time and scheduling performance (MSD) trajectory as shown in Figure 2(a). Figure 2(b) shows the arrival time trajectory for the same problem with different condition of initial arrival times as $\mathbf{a}(0) = [0 \quad 10]$. Figure 2(b) shows that regardless of different initial conditions, the arrival time trajectory converges to the identical steady-state value and this has been proved in Prabhu and Duffie (1999). It should be pointed out that the scheduling performance (MSD) in DATC is dependent on the quality of steady-state value and relative order of the arrival times (processing sequence). Therefore, it is important to design the controller of the DATC in such a way that it can drive the arrival time trajectory to a region where high quality of scheduling performance is ensured.

Whereas a common due date was considered in the example of Figure 2, Figure 3 shows different arrival time trajectory from Figure 2 if due dates are distinct. Note that due dates and processing times are constant in this example and thus we omit time variable t. With distinct due dates, arrival time trajectory can converge to either an identical steady-state value (Figure 3(a)) or distinct values (Figure 3(b)) depending on the relationship between processing times and due dates. If due dates of two parts are given close to each other, then it is impossible to meet those due dates simultaneously due to insufficient resource capacity. We call due dates that cannot be met simultaneously 'infeasible' due dates, otherwise they are referred to as 'feasible' due dates. It is clear and straightforward to show that arrival time trajectory converges to distinct values of ($\mathbf{d}$–$\mathbf{p}$) if due dates are feasible. On the other hand, infeasible due dates make arrival time trajectory to converge to a steady-state value; which is more difficult yet interesting to analyse.



d = [10.5 10] **d** = [12 10]

Figure 3. Arrival time trajectory in DATC of two-part in a single machine together with completion time and performance trajectory (distinct due dates).

Therefore, the focus of this research is on the analysis of dynamic characteristics of arrival time trajectory when due dates are infeasible.

3. Dynamic characteristics of DATC

Two key determinants of scheduling performance in the DATC approach are the steady-state values and the relative order of arrival times which determine starting times of part processing. Clearly, processing sequence is determined by the relative order of arrival times. To design an enhanced controller for the DATC in terms of performance and dynamics, we first investigate the unique dynamic characteristics of the DATC in terms of its convergence and chattering of arrival times.

3.1 *Steady-state arrival time*

Dynamics of arrival times in the DATC is modelled by the following differential equation:

$$\dot{a}_i(t) = k_i z_i(t) = k_i[d_i(t) - c_i(t)] = k_i[d_i(t) - a_i(t) - q_i(t) - p_i(t)] \tag{4}$$

where $q_i(t)$ denotes queuing time. It should be pointed out that the right hand side of Equation (4) has discontinuity because queuing times are changed discontinuously depending on different part processing sequences, which are determined by the relative orders of arrival times. Filippov (1960) proposed a solution method for differential equation that has discontinuity in the right hand side by transforming it into differential inclusion. Dynamics of arrival times in the DATC has been solved using the Filippov's method (Prabhu and Duffie 1999). Figure 4 illustrates the dynamics including the

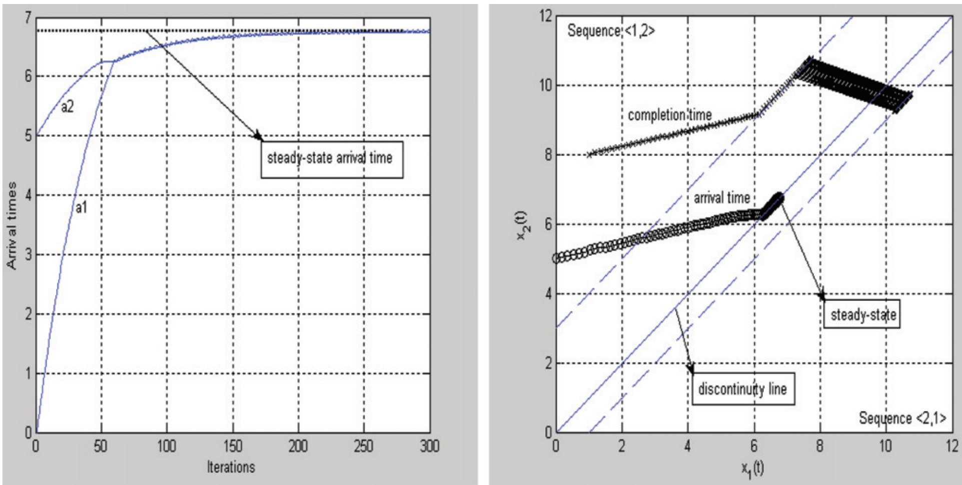
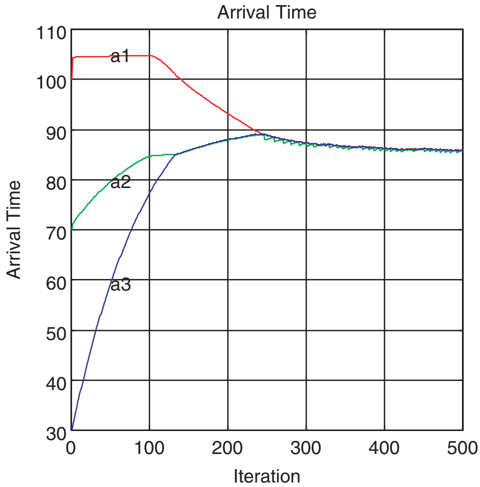


Figure 4. Arrival time trajectory in DATC of two-part in a single machine in time domain (left) and state-space domain (right).

trajectory of arrival times and completion time for the same two-part single machine example with $\mathbf{a}(0) = [0 \quad 5]$. As shown in Figure 4 (right), discontinuity region is defined as $a_1(t) = a_2(t) = \cdots = a_n(t)$ when scheduling parameters of the DATC are represented in the vector space R^n. Here, the discontinuity region represents a space where arbitrary tie breaking of identical arrival times can lead to discontinuity in the completion times. When the arrival time trajectory passes through the discontinuity region, the completion time vector changes its location discontinuously from one sequence region to another sequence region, say from sequence ⟨1, 2⟩ to ⟨2, 1⟩ or vice versa. Note that in the figure, the upper triangle with respect to the discontinuity region represents processing sequence ⟨1, 2⟩ and the lower triangle represents processing sequence ⟨2, 1⟩. In the figure, a region enclosed by dashed lines is called 'queuing region' because queuing times incur in this region. This region is also called 'infeasible region' because any due date vector in this region is infeasible to be met simultaneously.

For infeasible due dates which are given in the infeasible region, arrival time trajectory moves towards the discontinuity region regardless of different initial conditions of arrival times because moving velocity is governed by the due date deviation vector (see Equation (4)). In Figure 4, arrival time trajectory initiated by $\mathbf{a}(0) = [0 \quad 5]$ remains in sequence ⟨1, 2⟩ region moving towards to the discontinuity region. Until the trajectory passes through the discontinuity region, the dynamics of arrival times is modelled by ordinary differential equation. However, as the trajectory moves across the discontinuity region, processing sequence changes discontinuously resulting in discontinuous changes in the completion time vector and thus directional changes in due date deviation vector. In the example of Figure 4, arrival time trajectory moves into sequence ⟨2, 1⟩ region from sequence region ⟨1, 2⟩ and the directional changes in due date deviation vector makes the trajectory moving back to the other side of the discontinuity region (sequence ⟨1, 2⟩). Likewise, the trajectory remains near the discontinuity regions and converges to a steady-state region. The volume of steady-state region is determined by the control system gain (Cho and Prabhu 2000a, b). In addition, control system gain determines moving



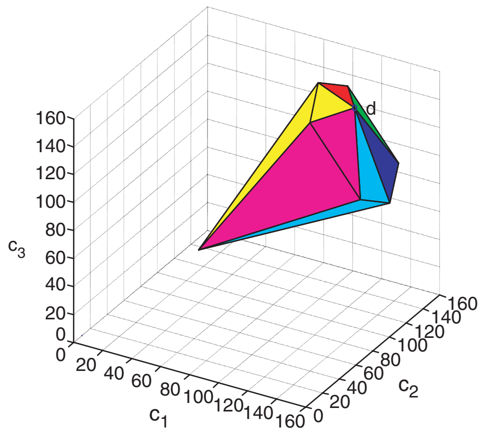


Figure 5. Arrival time trajectory (left) and convex hull geometry (right) in a three-part example.

speed or convergence rate to the steady-state region. The dynamics in the steady-state region is represented by differential inclusion due to the directional changes in arrival time trajectory. It has been analysed in previous research that the due date deviation vector is orthogonal to the processing time vector and using this property the steady-state value of arrival times (a_{ss}) has been predicted as follows (Prabhu and Duffie 1999):

$$a_{ss} = \frac{(\mathbf{p} \cdot \mathbf{d}) - (\mathbf{p} \cdot \mathbf{r})}{\sum_{i=1}^{n} p_i} \tag{5}$$

where $(\cdot)$ represents dot product and $\mathbf{r}(t) = \mathbf{c}(t) - \mathbf{a}(t)$ is a flow time vector for any processing sequence.

Figure 5 (left) illustrates the arrival time trajectory converges to a steady-state value (a_{ss}) in a three parts example, where infeasible due dates are given. Under the steady-state condition, small changes in arrival time trajectory across the discontinuity region cause changes in processing sequence resulting in discontinuous changes in completion times. The discontinuous changes in completion times implicate instantaneous changes in due date deviation vector that make the arrival time trajectory moving back to the steady-state region, more specifically, another sequence region across the discontinuity region. Note that in this three parts example, six possible completion time vectors corresponding to permutation schedules at the steady-state condition form vertices on hexagonal base (convex hull on a hyper-plane) of the pyramid as shown in Figure 5 (right). This dynamic behaviour makes the arrival time trajectory to visit different part processing sequences and to be used as a search engine for optimal or near-optimal production schedules replacing conventional heuristics.

3.2 *Sequence visiting mechanism of DATC*

As explained above, the DATC has a unique mechanism to visit different processing sequences and it can be efficiently illustrated using Figure 6 for two-parts in a single

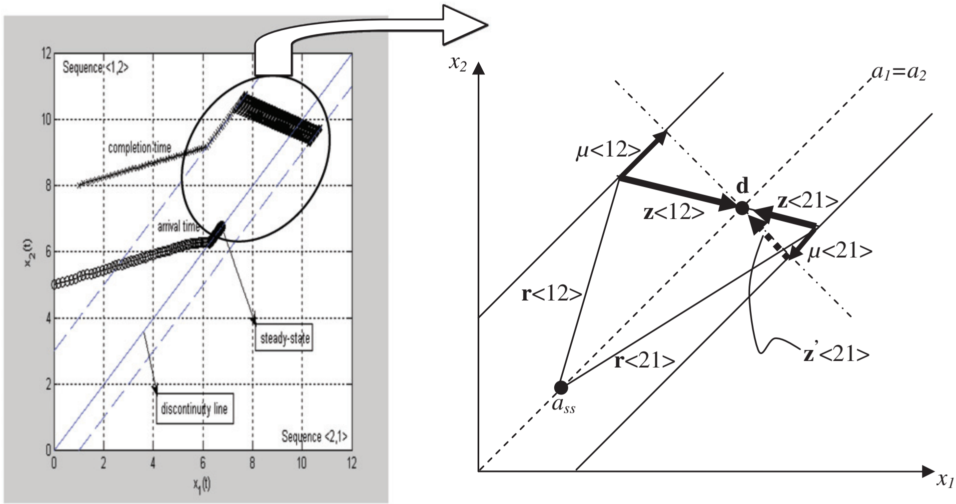


Figure 6. Velocity component of arrival time along discontinuity line.

machine scheduling again. The simulation of the trajectory of the arrival times and completion times in the steady-state region can be modelled using a unique steady-state value (a_{ss}) of Equation (5) and convex hull geometry consisting of flow time vectors ($\mathbf{r}\langle 12\rangle, \mathbf{r}\langle 21\rangle$) as shown in Figure 6 (right) where it is clear that two possible completion time vectors for sequence $\langle 12$ and $\langle 21$ with respect to identical are expressed as follows:

$$\mathbf{c}\langle 12\rangle = a_{ss}[1 \quad 1]^T + \mathbf{r}\langle 12\rangle = a_{ss}[1 \quad 1]^T + [p_1 \quad p_1 + p_2]^T$$
$$\mathbf{c}\langle 21\rangle = a_{ss}[1 \quad 1]^T + \mathbf{r}\langle 21\rangle = a_{ss}[1 \quad 1]^T + [p_1 + p_2 \quad p_2]^T$$

In addition, it is straightforward to show that the base of the convex hull is orthogonal to the processing time vector since $(\mathbf{c}\langle 12\rangle - \mathbf{c}\langle 21\rangle \cdot \mathbf{p}) = 0$. This property holds for n parts in a single machine without loss of generality.

The due date deviation vectors ($\mathbf{z}\langle 12\rangle$, $\mathbf{z}\langle 21\rangle$) are, of course, on the base of the convex hull as shown. Interestingly, we can consider a line (dash dot) that is orthogonal to the discontinuity line ($a_1 = a_2$). This line can be a hyper-plane in general and we call this hyper-plane 'optimal plane' in this paper. The optimal plane can be obtained by shifting completion time vectors by $\mu\langle\cdots\rangle$, where $\langle\cdots\rangle$represents arbitrary processing sequence. It is not difficult to show that

$$\mu\langle\cdots\rangle = \frac{\sum d_i - \sum c\langle\cdots\rangle_i}{n} \tag{6}$$

Suppose processing times and due dates are given as $\mathbf{p} = [1 \quad 2]$ and $\mathbf{d} = [10 \quad 10]$. For this problem, we have $a_{ss} = 7.67$ and $\mu\langle 12\rangle = 0.333$, $\mu\langle 21\rangle = -0.167$ from Equations (5) and (6). Since in the steady state the arrival time trajectory stays in a certain region along the discontinuity line, the summation of velocity component (μ) must be zero. To meet this steady-state condition, arrival time trajectory visits sequence $\langle 21\rangle$ twice as frequently than sequence $\langle 12\rangle$ in this specific example. From the scheduling performance (MSD) point of view, it is clear that sequence $\langle 21\rangle$ is the optimal sequence in this example, because

$||\mathbf{z}\langle 21\rangle|| < ||\mathbf{z}\langle 12\rangle||$. However, we can observe that we can improve the scheduling performance more by projecting $\mathbf{z}\langle 21\rangle$ onto the optimal plane so that the due date deviation vector becomes $\mathbf{z}'\langle 21\rangle$. The projection can be achieved by shifting the identical start time (a_{ss}) to $a_{\mathrm{opt}} = a_{ss} + \mu\langle 21\rangle$. The need of such shifting can be considered as limitation of the existing DATC approach from the scheduling performance point of view.

The sequence visiting mechanism of DATC approach explained above becomes more complicated as the number of parts to be scheduled increases. In general, the net velocity of arrival time trajectory (equivalently convex combination of $\mathbf{z}(t)$ vectors) at the steady-state condition is expressed as

$$\dot{\mathbf{a}}(t) = \alpha_1\dot{\mathbf{a}}_1(t) + \cdots + \alpha_i\dot{\mathbf{a}}_i(t) + \cdots + \alpha_N\dot{\mathbf{a}}_N(t) = k\sum_{i=1}^{N}\alpha_i\mathbf{z}_i(t) = \mathbf{0} \tag{7}$$

where $k = k_i$ is integral control gain, and $N = n!$ (total number of possible sequences). For $\forall i = 1, \ldots, N$ we have

$$\sum_{i=1}^{N}\alpha_i = 1 \tag{8}$$

Note that α_i can be considered as the weight of the contribution or visiting frequency of sequence i. Using Equations (7) and (8), we can have

$$\mathbf{d} - \mathbf{a}_{ss} - \mathbf{p} = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_N\end{bmatrix}\alpha = \mathbf{Q}\alpha \tag{9}$$

where $\mathbf{Q}$ is N by n queuing matrix,

$$\alpha_{ss} = a_{ss}[\underbrace{1 \quad 1 \quad \cdots \quad 1}_{n}]$$

and $\boldsymbol{\alpha} = \begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_N\end{bmatrix}$. Equation (9) can then be found using pseudo inverse (+) as follows:

$$\boldsymbol{\alpha} = \mathbf{Q}^{+}[\mathbf{d} - \mathbf{a}_{ss} - \mathbf{p}] \tag{10}$$

$\boldsymbol{\alpha}$ allows us to predict distribution of visiting frequencies for all the possible processing sequences and Figure 7 shows the comparison of sequence visiting frequency obtained from DATC simulation and from Equation (10) for three parts and four parts in a single machine scheduling problem. In the simulation, total number of iterations is 5000. From the comparison, it is observed that model can approximate the distribution of sequence visiting frequency of the DATC simulation with a need of further improvement. The discrepancy between the model and simulation is caused by the fact that Equation (10) involves deterministic parameters and pseudo inverse manipulation is used to solve it.

Figure 8 depicts the sequence visiting mechanism of the DATC in more detail for a scheduling problem with three parts and a single machine. In the figure, different control system gains are represented by k_s (small), k_m (medium) and k_h (high). In addition, arrival time and completion time space are divided into six possible sequence regions of $\langle 123\rangle$, $\langle 132\rangle$, $\langle 213\rangle$, $\langle 231\rangle$, $\langle 312\rangle$, and $\langle 321\rangle$. In Figure 8 (left), the inside of the sphere represents the steady-state region, where a_i illustrates arrival time vector at the ith iteration. Depending on the value of control system gain, the arrival time vector at the $(i+1)$th iteration can be either $a_{i+1,1}$, $a_{i+1,2}$, or $a_{i+1,3}$ in different sequence regions. This implies that the arrival time trajectory of DATC is dependent on the control system gains and makes

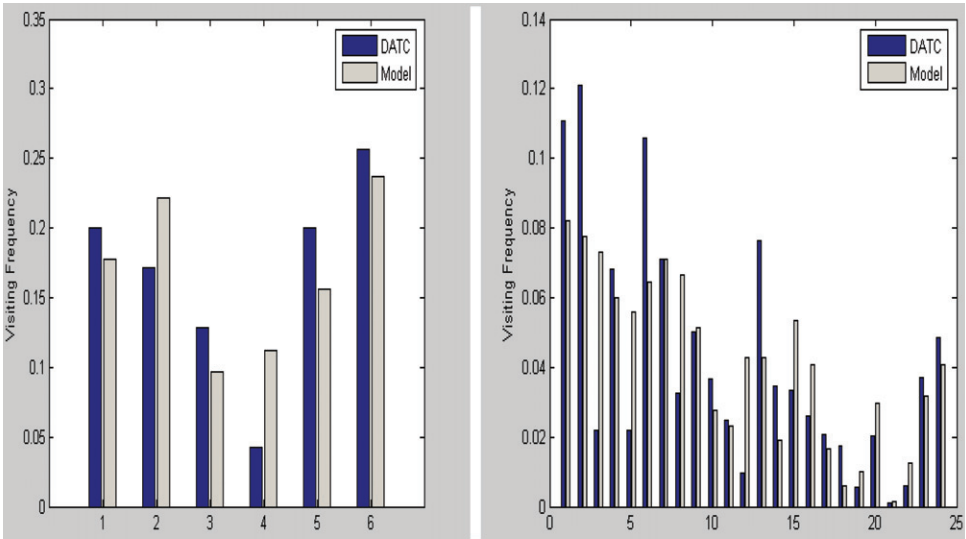


Figure 7. Distribution of sequence visiting frequency from DATC simulation (left) and from model using Equation (10) (right) with control gain $k = 0.05$.

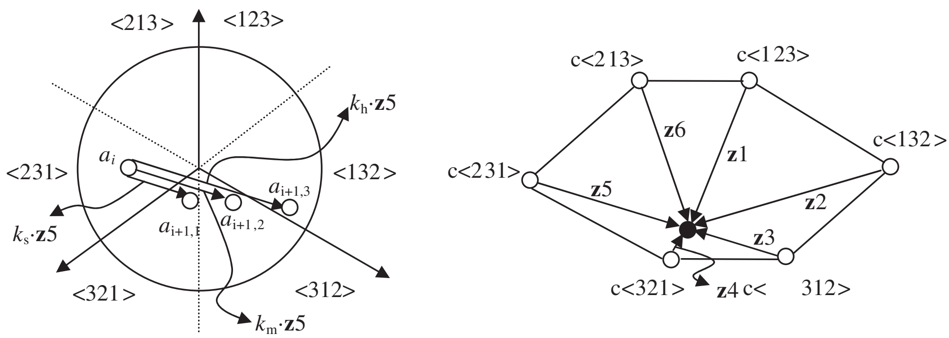


Figure 8. Arrival time trajectory in arrival time space (left) and due date deviation vectors in completion time space (right).

the DATC a dynamically unpredictable system from the sequence visiting point of view. Hence, primary focus of this research is to design a new controller for the existing DATC approach that can improve the predictability of the sequence visiting frequency.

4. Double integral arrival time control system (DIAC)

It has been observed in previous research (Cho and Prabhu 2002a) that scheduling performance measure (MSD) for a specific processing sequence ($\langle \ldots \rangle$) is a convex function of steady state arrival times. This observation becomes clearer when we consider the same example problem as in Figure 6. More specifically, for a processing sequences $\langle 12 \rangle$ and $\langle 21 \rangle$, MSD is a convex function of the steady state arrival time and minimised when the

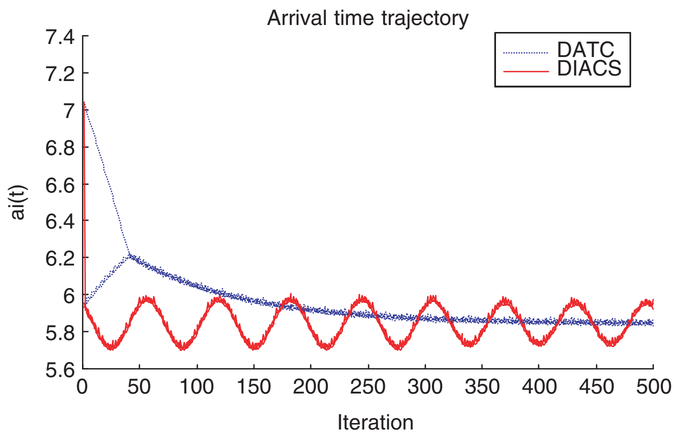


Figure 9. Arrival time trajectory of DIAC and DATC.

steady state arrival time is changed from a_{ss} to $a_{ss} - \mu\langle 21\rangle$ and $a_{ss} + \mu\langle 12\rangle$, respectively. From scheduling performance perspective, this implies that local optima corresponding to different processing sequences are obtained by shifting the steady state arrival times. It should be emphasised here that shifting the steady state arrival times requires global information (for example, due dates and processing times of other parts), which may not be desirable for distributed control system. From the control perspective, shifting steady arrival times can be achieved by designing a new controller that makes arrival time trajectory not to converge to a unique $a_{\rm ss}$ but to sweep a certain range periodically. For this need and improved predictability described in the end of Section 3, we design a double integral controller in this paper, which is called DIAC (double integral arrival time controller).

4.1 *Dynamic analysis of DIAC*

A distinguishable property of DIAC is its double integral control law employed in the existing DATC. With this double integral controller, arrival time of the ith part in DIAC can be expressed as follows:

$$a_i(t) = k_i \iint_t z_i(\tau)d\tau + a_i(0) \tag{11}$$

The second order differentiation of Equation (11) gives us the following dynamics of DIAC:

$$\ddot{a}_i(t) + k_i a_i(t) = k_i(d_i - p_i) - k_i q_i(t) \tag{12}$$

Figure 9 compares arrival time trajectories of DIAC and DATC for three parts in a single machine example. In this example, due dates and processing times are given as $\mathbf{d} = [10, 10, 10]$, $\mathbf{p} = [1, 2, 3]$. Due to the double integral controller of Equation (11) that leads to the second order differential equation in Equation (12), and the arrival time trajectory of DIAC oscillates within a certain range including the steady state arrival time, a_{ss}. Details of this dynamic property of DIAC including investigation of the initial moving

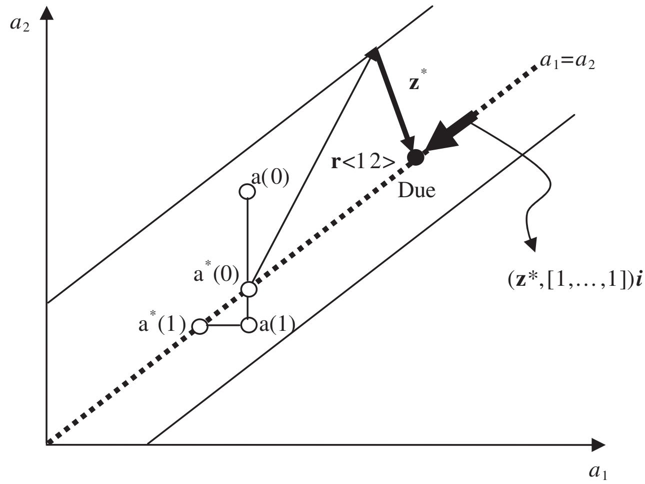


Figure 10. Dynamic behaviour of DIAC.

direction, amplitude, and frequency of the oscillation of the arrival time trajectory is analysed in the following.

Initial moving direction of arrival time trajectory in DIAC is determined by location of the initial arrival times, which is denoted as $a(0)$ in Figure 10 for a two parts in a single machine example, because the direction of initial velocity vector is proportional to ($\mathbf{z}^*$, [11])$\mathbf{i}$ by Equation (11) or equivalently Equation (12). Here, $\mathbf{z}^*$ is the initial due date deviation vector, which is $\mathbf{z}^* = \mathbf{d} - \mathbf{c}^*$, and $\mathbf{i}$ represents an identity vector along the discontinuity line $a_1 = a_2$. Due to this property, the trajectory moves toward a_{ss} and oscillates around a_{ss}. This is explained below.

Amplitude of oscillation is determined by the initial condition of arrival times. The distance between initial arrival times and a_{ss} along discontinuity line is presented by $|a_{ss} - \min(\mathbf{a}(0))|$. Starting from the initial arrival time vector $\mathbf{a}(0)$, the moving direction of arrival times along the discontinuity line is not changed until it reach a_{ss}. After the trajectory passes through a_{ss}, the direction of due date deviation vector is reversed. However, the accumulation of due date deviation vector due to double integral control law is not zero in spite of the directional change of the due date deviation vector. Hence, the trajectory continues to move along the same direction until the accumulation becomes zero. The accumulated amount of due date deviation vector during the movement of arrival times to a_{ss} becomes zero when arrival times move to $|a_{ss} + \min(\mathbf{a}(0))|$. From this observation, the amplitude (A) of the oscillation of arrival times can be modelled as follows:

$$A = |\min(\mathbf{a}(0)) - a_{ss}| \tag{13}$$

Similar to other systems that can be modelled by the second order of differential equation, natural frequency (Ω) and period (τ) of arrival time oscillation are obtained by the following relationship:

$$\Omega_i = \sqrt{\frac{k_i}{1}}, \quad \tau_i = \frac{2\pi}{\Omega_i} \tag{14}$$

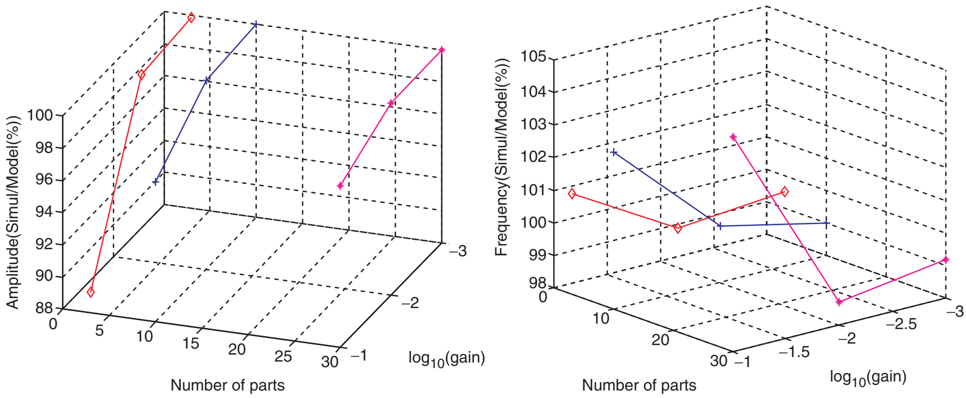


Figure 11. Prediction of amplitude and frequency of arrival time oscillation in DIAC.

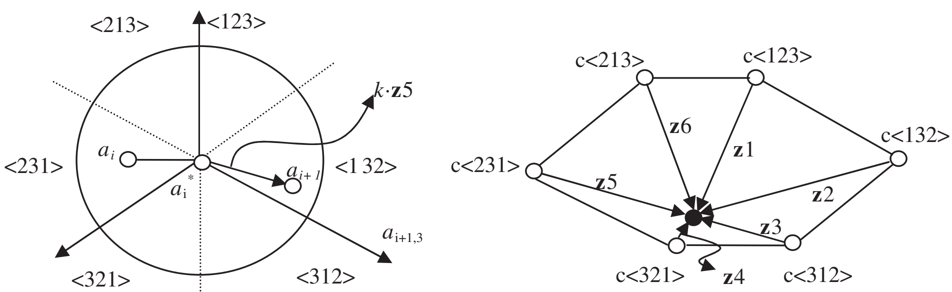


Figure 12. Antithetical sequence visiting mechanism of DIAC.

We have tested Equations (13) and (14) for various problem instances with random conditions of the initial arrival times. Numbers of parts used for the test are 3, 10 and 30. In the test, three levels of control system gain ($k = k_i = 0.1, 0.01, 0.001$) were considered. The test results show that Equations (13) and (14) can predict frequency and amplitude within less than 10% error compared to simulation results as depicted in Figure 11.

Figure 12 illustrates the sequence visiting mechanism of DIAC. Whereas DATC can visit a neighbouring sequence or antithetical sequence depending on the control gains, DIAC visits exactly an antithetical sequence. This property is achieved by translating current arrival times ($\mathbf{a}_i$) to new arrival times ($\mathbf{a}_0^*$) on the discontinuity line as shown in Figure 12. Algebraically, this is equivalent to taking minimum value of current arrival times as the common arrival time for all other parts. Thus, the antithetical sequence ($\mathbf{a}_{i+1}$) is uniquely determined along the direction of the due date deviation vector regardless of different values of the control system gains and this results in improved predictability of the arrival time trajectory and chattering in DIAC.

4.2 *Performance analysis of DIAC*

Scheduling performance of DIAC is evaluated with several problem instances for which global optimum is known. In addition to the evaluation for the problem with known

Table 1. Problem sets with known optima.

Problem	n	Processing time (**p**)	Optimal sequence
1	6	2,6,9,12,19,21	6-4-3-1-2-5
2	7	2,3,6,9,21,65,82	7-5-4-3-2-1-6
3	8	2,4,6,7,8,9,10,16	8-6-5-2-1-3-4-7
4	8	1,2,8,9,10,12,13,16	8-6-4-3-1-2-5-7
5	10	1,2,5,8,9,10,13,16,18,19	10-8-7-4-2-1-3-5-6-9
6	10	2,3,6,9,21,23,34,65,82,92	10-8-7-4-3-2-1-5-6-9
7	10	5,7,8,9,10,13,21,25,41,100	10-8-7-5-2-1-3-4-6-9

Table 2. Percent deviations of MSD from optima.

		DATC			DIAC		
Problem	Optimum	Best (% dev.)	Average (% dev.)	Worst (% dev.)	Best (% dev.)	Average (% dev.)	Worst (% dev.)
1	218.47	0.01	2.10	2.35	0.00	0.11	1.07
2	918.29	1.18	1.44	3.91	0.00	0.00	0.00
3	187.23	0.26	1.65	1.72	0.80	1.49	2.87
4	254.23	0.72	0.86	3.89	0.25	0.77	1.97
5	486.40	3.10	3.32	8.11	0.01	1.61	3.33
6	3584.00	4.63	5.02	5.04	0.11	1.29	4.48
7	1336.24	0.64	2.32	2.45	0.62	3.79	4.96

optima, relatively large size problems with 100 parts are tested. In the latter case, performance is compared with the DATC.

4.2.1 *Performance evaluation with known optima*

Several CDD (common due-date) problem instances from Kim and Foote (1996), shown in Table 1, are used to evaluate the scheduling performance of DIAC. For these problem instances, the optimal MSD and corresponding optimal sequence are known. In Table 1 common due date is assigned to each part as summation of all processing times.

Table 2 shows a computational result of the performance comparison between DIAC and DATC. Considering the random effect of initial arrival times, the scheduling performance has been evaluated through 20 replications for each problem instance. Duration of numerical simulation has been set to 500 iterations to ensure the steady-state condition of arrival times. Using Equation (15), the worst-case performance of DIAC is 4.96% and the best is 0% deviation from the global optimum.

$$\frac{\mathrm{MSD} - \mathrm{MSD}_{\mathrm{opt}}}{\mathrm{MSD}_{\mathrm{opt}}} \times 100\% \tag{15}$$

4.2.2 *Performance evaluation with DATC*

For large size problems, it is hard to compare the scheduling performance of DIAC in an absolute sense, because these problems belong to NP-complete. Therefore, scheduling performance improvement has been examined with respect to the DATC. Figure 13 shows

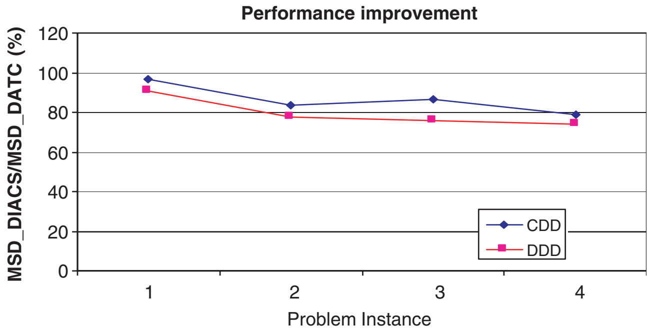


Figure 13. Performance comparison of DIAC and DATC.

performance improvement for four problem instances with $n = 10$, 30, 50, 100 parts for both CDD and DDD (distinct due-date). Twenty trials with random initial conditions are replicated for each problem instance. Processing times are selected randomly in the range of $U(1100)$ and due dates are assigned by $\sum_i p_i + U(1{,}100)$ summation $(p) + U(1100)$. The result in Figure 13 shows that the scheduling performance of DIAC is improved by 26.2% with respect to DATC in case of a 100 parts problem with DDD while it is improved by 3.5% in case of a 10 parts problem with CDD.

4.3 *DIAC for dynamic job shop scheduling*

While the performance analysis of DIAC given in previous sections is focused on static single machine scheduling problems, most of the scheduling problems in real circumstances are dynamic and complicated. To make the proposed DIAC a viable scheduling tool for more realistic circumstances, we have considered the application of DIAC for dynamic job shop problems as follows. Figure 14 illustrates arrival time trajectories of DIAC and DATC for a job-shop scheduling, which has the following problem configuration:

Part number	Part type	Routings	Processing time
1	1	M1-M2-M3	1-3-7
2	1	M1-M2-M3	1-3-7
3	1	M1-M2-M3	1-3-7
4	1	M1-M2-M3	1-3-7
5	1	M1-M2-M3	1-3-7
6	2	M3-M2-M1	12-4-6
7	2	M3-M2-M1	12-4-6
8	2	M3-M2-M1	12-4-6
9	2	M3-M2-M1	12-4-6
10	2	M3-M2-M1	12-4-6

Figure 14 shows that arrival times of DIAC oscillate as in single machine problems while the arrival times of DATC converge to a steady state. It has been observed that DIAC outperforms DATC for this problem instance such that MSD_DIAC = 5657.837

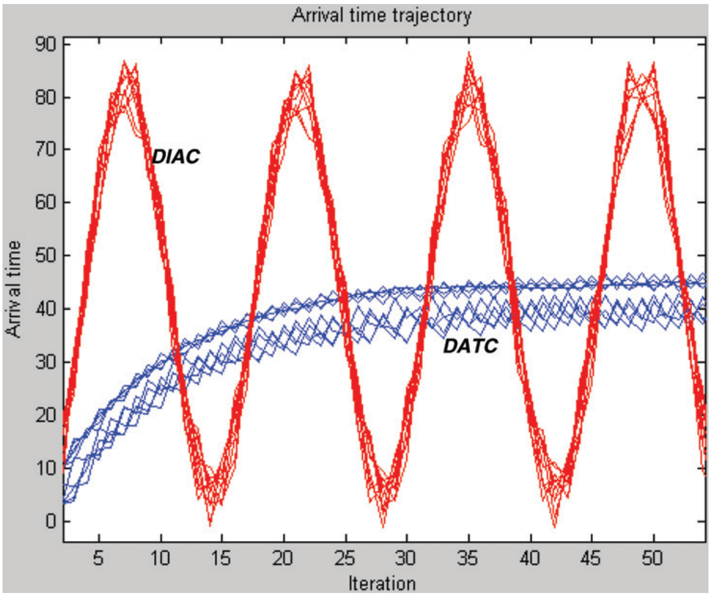


Figure 14. Arrival time trajectory in DIAC for a job shop problem instance (oscillatory: DIAC; converging: DATC).

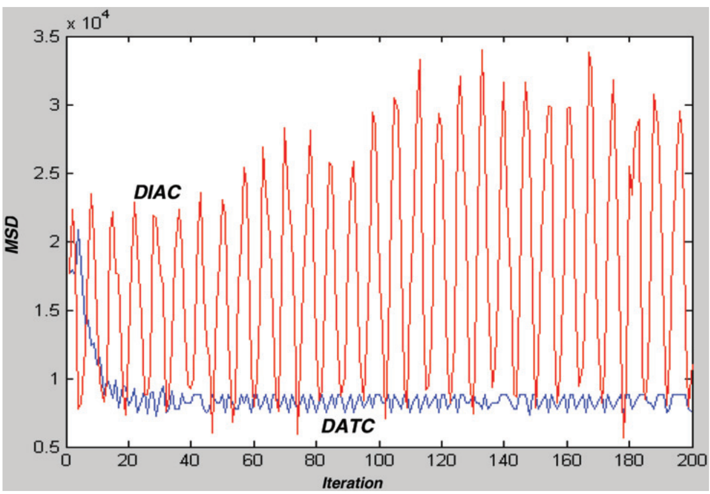


Figure 15. Performance comparison of DIAC and DATC (oscillatory: DIAC; converging: DATC).

and MSD_DATC = 7246.721 (see Figure 15). In other words, 22% performance improvement has been observed. In addition, DIAC has been applied to a dynamic job shop problem and the result is shown in Figure 16. DIAC controls arrival times in marginally stable manner when parts arrive dynamically such that it does scheduling for nine parts at the beginning and adjusts arrival times to take account for arrival of another part (10th part) in the system. Note that a similar job shop problem to Figure 14 has been

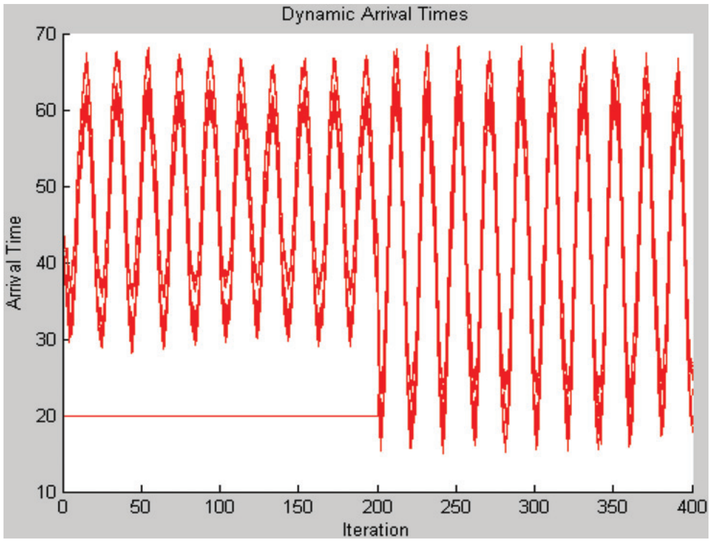


Figure 16. Arrival time trajectory of DIAC for a dynamic job shop problem.

used and the change in amplitude of oscillatory trajectory is because the initial arrival time of the newly arrived (or 10th) part is different from those of other parts that are in the system previously (refer to Equation (13)).

5. Conclusions

In this paper, unique dynamic characteristics of the DATC such as convergence and chattering of arrival times are analysed. The analysis has resulted in a closed form model to predict the sequence visiting frequency and identified limitations of the existing DATC approach. The identification of the limitation has led to the design of a double integral arrival-time controller (DIAC). Due to an embedded double integral controller, DIAC exhibits an oscillating dynamic property of arrival time trajectory with improved predictability in the sequence visiting mechanism. This paper proposes analytical models that can predict the amplitude and frequency of such oscillation in the arrival times. In addition, scheduling performance of DIAC is evaluated for various problem classes in this paper. For small size problems with less than 10 parts, MSD scheduling performance of DIAC has been observed to deviate from the optimum 4% in average and 5% in worst-case. Using DIAC we observe more than 30% performance improvement over DATC for large size problems with 100 parts. The proposed system can be used as a mathematical tool and a simulation model for designing adaptable manufacturing systems. In the future, research is required to analyse more rigorously the sequence visiting mechanism of DIAC in more general configurations such as flow shops and job shops.

References

Aarts, E.H.L., Frans, M.J. and Habers, E.H.A., 1986. Parallel implementations of the statistical cooling algorithm. *VLSI Journal*, 4 (3), 209–238.

Abdul-Razaq, T.S. and Potts, C.N., 1988. Dynamic programming state-space relaxation for single-machine scheduling. *Journal of Operational Research Society*, 39 (2), 141–152.

Bagchi, U., Sullivan, R.S. and Chang, Y.L., 1987. Minimising mean squared deviation of completion times about a common due date. *Management Science*, 33, 894–906.

Caramanis, M. and Sharifnia, A., 1991. Near optimal manufacturing flow controller design. *International Journal of Flexible Manufacturing Systems*, 3, 321–336.

Cho, S. and Prabhu, V.V., 2002a. A vector space model for variance reduction in single machine scheduling. *IIE Transactions*, 34, 933–952.

Cho, S. and Prabhu, V.V., 2002b. Sliding mode dynamics in continuous feedback control for distributed discrete-event scheduling. *Automatica*, 38 (9), 1499–1515.

Dessouky, M.I., Moray, N. and Kijowski, B., 1995. Strategic behaviour and scheduling theory. *Human Factors*, 37 (3), 443–472.

Eilon, S. and Chowdhury, I.G., 1977. Minimising waiting time variance in the single machine problem. *Management Science*, 23, 567–575.

Filippov, A.F., 1960. Differential equations with discontinuous right-hand sides. *American Mathematical Society Translations*, 42, 199–231.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13 (5), 533–549.

Goldberg, D.E., 1989. *Genetic algorithms in search, optimisation, and machine learning*. England: Addison-Wesley.

Kim, C. and Foote, B.L., 1996. Minimising the mean squared deviation from a common due date: unconstrained and constrained cases. *Production Planning & Control*, 7 (5), 492–502.

Li, G., 1997. Single machine earliness and tardiness scheduling. *European Journal of Operational Research*, 96, 546–558.

Luh, P. and Hoitomt, D.J., 1993. Scheduling of manufacturing systems using the Lagrangian relaxation technique. *IEEE Transactions on Automatic Control*, 38 (7), 1066–1079.

McKay, K.N. and Wiers, V.C.S., 1999. Unifying the theory and practice of production scheduling. *Journal of Manufacturing Systems*, 18 (4), 241–255.

NRC, 1998. *Visionary manufacturing challenges for 2020*. National Research Council Report.

Parunak, H.D., 1991. Characterising the manufacturing scheduling problem. *Journal of Manufacturing Systems*, 10 (3), 241–259.

Prabhu, V.V. and Duffie, N.A., 1999. Nonlinear dynamics in distributed arrival time control of heterarchical manufacturing systems. *IEEE Transactions on Control Systems Technology*, 1 (6), 724–730.

Sabuncuoglu, I. and Karabuk, S., 1999. Rescheduling frequency in an FMS with uncertain processing times and unreliable machines. *Journal of Manufacturing Systems*, 18 (4), 268–283.

Veeramani, D. and Wang, K.-J., 1997. Performance analysis of auction-based distributed shop-floor control schemes from the perspective of the communication system. *The International Journal of Flexible Manufacturing Systems*, 9, 121–143.

Zweben, M., Davis, E., Daun, B. and Deale, M.J., 1993. Scheduling and rescheduling with iterative repair. *IEEE Transactions on Systems, Man, and Cybernetics*, 23 (6), 1588–1596.