

Brief Papers

Nonlinear Dynamics in Distributed Arrival Time Control of Heterarchical Manufacturing Systems

Vittaladas V. Prabhu and Neil A. Duffie

Abstract—Heterarchical control architectures with fully distributed control have been developed in order to improve responsiveness and effectiveness of manufacturing shop-floor control systems. The dynamics of these highly distributed systems have been difficult to predict particularly when control is based on heuristics. In this paper a dynamical model is developed for a single machine processing an arbitrary number of parts. The structure of the system, which requires queuing of parts when they arrive at a machine, leads to nonlinearities such as dead-zone and discontinuities. A continuous arrival time controller of the integrating type is used that results in a system that can be modeled using nonlinear differential equations that can be solved using a method due to Filippov. This enables prediction of trajectories of part arrival times and derivation of closed form expressions for steady-state values. The analytical model for the dynamics is validated and the dynamic response of the system is illustrated using numerical simulation.

Index Terms—Discontinuities, discontinuous differential equations, distributed control, distributed information systems, heterarchical manufacturing scheduling, manufacturing systems.

I. INTRODUCTION

ONE of the main objectives of manufacturing shop-floor control systems is to maintain good performance by rescheduling in real-time to compensate for disturbances such as rush orders, machine failures, tool breakages, etc. Scheduling can be defined as determining the timing of controllable events on the manufacturing shop-floor. Past work in scheduling has predominantly approached it as a combinatorial problem for finding optimal schedules, a computationally intractable problem [1]. Moreover, the information required for making scheduling decisions is usually distributed throughout the shop-floor, and in practice it can be difficult to obtain in real-time. This lack of information feedback and computational burden has limited the responsiveness and effectiveness of manufacturing shop-floor control systems.

A range of manufacturing control architectures have been developed in order to improve responsiveness and effectiveness. At one extreme are centralized architectures in which one supervisory controller controls the entire system based on rigid master-slave relationships and large amounts of global information; and at the other extreme are heterarchical architectures in which there are no supervisory controllers [2]. In heterarchical architectures control is fully distributed among loosely coupled, cooperating, autonomous controllers that retain a minimum amount of global information [3]. These characteristics tend to endow heterarchical systems with reduced complexity, increased fault-tolerance, and increased flexibility compared to centralized and hierarchical systems as demonstrated in several experimental systems [2]–[6]. Furthermore, highly distributed control based on local information reduces the computational burden and facilitates real-time information feedback thereby enhancing responsiveness and effectiveness of heterarchical shop-floor control systems [4].

The potential for applying control theoretic approaches to manufacturing systems has been widely recognized. Specifically, control theory has been successfully applied for controlling the flow of materials through different routes on the shop floor with the objective of maintaining certain production rates [7]–[8]. These approaches deal exclusively with rates rather than constituent events which allows the discrete event scheduling problem to be approximated as a continuous system in which the frequency of control decisions is at least an order of magnitude lower than frequency of events on the shop-floor. Therefore, flow control approaches are suitable for make-to-stock, high-volume manufacturing. For low-volume/high-variety manufacturing systems, the challenge is to develop dynamical models amenable to control theoretic analysis and synthesis techniques.

In the remainder of this paper a dynamical model is developed for a single machine system with arbitrary number of parts. The structure of the system, which requires queuing of parts when they arrive at a machine, leads to nonlinearities such as dead-zone and discontinuities. The choice of a continuous arrival time controller of the integrating type results in a system that can be modeled using nonlinear differential equations. It is shown that these can be solved using a method due to Filippov, which enables prediction of trajectories of part arrival times and derivation of closed form expressions

Manuscript received March 23, 1997; revised December 4, 1998. Recommended by Associate Editor, W. Gruver. This work was supported in part by the National Science Foundation under Grants DMC-8912361 and DMI-9612003.

V. V. Prabhu is with the Department of Industrial and Manufacturing Engineering, Penn State University, University Park, PA 16802 USA.

N. A. Duffie is with the Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706 USA.

Publisher Item Identifier S 1063-6536(99)09043-0.

for steady-state values. An example is then given to illustrate both the dynamics of the system and geometric relationships in the solution.

II. CONCEPT OF DISTRIBUTED ARRIVAL TIME CONTROL

Heterarchical manufacturing systems can achieve a high level of control distribution by allowing each part entity (workpiece) to reserve required resources such as machine tools, robots, automatic guided vehicles, cutting tools, etc., using well-defined communication protocols [3]. The time at which a part initiates the reservation protocol for a given resource can be called the arrival time of the part at that resource. When first-come-first-served (FCFS) queuing policy is used at each resource, scheduling heterarchical manufacturing systems becomes a problem of determining various arrival times. Arrival time affects the performance of a part and performance of other parts with which it shares resources, as well as utilization and idle times of resources in the system.

Arrival time is a continuous variable that can be increased or decreased allowing a closed-loop arrival time controller to be constructed for each event that needs to be scheduled. Each controller continually adjusts arrival times and the resulting completion times, local performance, and global performance are predicted using distributed simulation [6], [9]–[11]. In this approach, the arrival times and the order of part arrivals can change significantly from simulation to simulation and hence with time. The best arrival time vector generated in these simulations, which results in the best global performance, can be used to schedule events on the shop floor. Specifically, the time at which a machine actually starts processing a part can be used as the scheduled arrival time on the shop floor thereby resolving in simulations all contentions for using resources. Here simulation serves as the feedback mechanism for closed-loop arrival time control [9].

III. DYNAMICAL MODEL FOR DISTRIBUTED ARRIVAL TIME CONTROL

Arrival time control of n events gives rise to the n -dimensional multivariable control system [10] where due-date $\mathbf{d}(t)$, completion time $\mathbf{c}(t)$, due-date deviation $\mathbf{z}(t)$, and arrival time $\mathbf{a}(t)$ are n -dimensional vectors which act as the command, output, error, and manipulated vectors, respectively. Parts are processed in the order in which they arrive; i.e., the queuing discipline is first-come-first-served (FCFS). If a part arrives when another part is being processed at the machine, it incurs queuing time. Therefore, if arrival times are close to each other, then parts interact via queuing.

The time when a part entity arrives is crucial because it will determine the processing sequence; i.e., it will determine how a part interacts with other parts because of its relative position in the queue. It should be noted that the first part in the queue does not incur queuing time thus defining the beginning of a queue. There can be more than one part with zero queuing time leading to “fragmented queues.” The machine idles between queue fragments because there are no parts to process. Because parts in different queue fragments do not interact, this analysis

focuses on a single queue fragment, which are assumed to be stationary for the sake of analysis.

The i th part in a given queue fragment incurs a queuing time of q_{qi} after which the machine processes it for a duration of p_{qi} and then completes it at c_{qi} . (The subscript q will be used henceforth to denote that analysis is being performed on a queue fragment.) The completion time for the i th part in the queue can be expressed as

$$c_{qi}(t) = a_{qi}(t) + q_{qi}(t) + p_{qi}(t) \quad (1)$$

and because of the FCFS queuing discipline

$$q_{qi}(t) = a_{q1}(t) + p_{q1}(t) + p_{q2}(t) + \cdots + p_{q(i-1)}(t) - a_{qi}(t). \quad (2)$$

Equation (2) is valid for any given queue fragment. From (1) and (2), completion time can be expressed as

$$c_{qi}(t) = a_{q1}(t) + p_{q1}(t) + p_{q2}(t) + \cdots + p_{q(i-1)}(t) - p_{qi}(t). \quad (3)$$

It is important to note that, except for the first part, completion time of a part is not affected by its own arrival time as long as the order of arrival times of parts in the queue is not changed. However, when the order of arrival times change, the part processing sequence changes and the corresponding changes in completion time are discontinuous. Therefore, unless arrival times are so far apart that each part is in its own queue fragment and incurs zero queuing time, completion time is a highly nonlinear function of arrival time, with dead-zones and discontinuities.

Depending on the arrival times, the dynamics of a system of n parts can be divided into three regions: the decoupled region, the dead-zone region, and the discontinuity region. In the decoupled region, M_c , there is no interaction among controllers, all queuing times are zero, and the dynamic behavior of each part can be analyzed independently. In the dead-zone region, M_z , two or more parts with unequal arrival times interact and at least one queuing time is greater than zero. This region is called the dead-zone region because the partial derivative of the completion time with respect to the arrival time is zero for any part which incurs queuing time, this is evident from (3).

Controlling arrival time for multiple parts gives rise to a multivariable control system with highly nonlinear coupling between arrival and completion times that can include dead-zone and discontinuity-type nonlinearities. Explicit linearization or decoupling is ruled out because it would require global information and rigid relationships. Instead, these nonlinearities can be ignored, aggregated as a disturbance, and compensated for using feedback in a local arrival time controller for each part as shown in Fig. 1 where d_{qi} is the due date and z_{qi} is the due date deviation. The disturbance aggregates

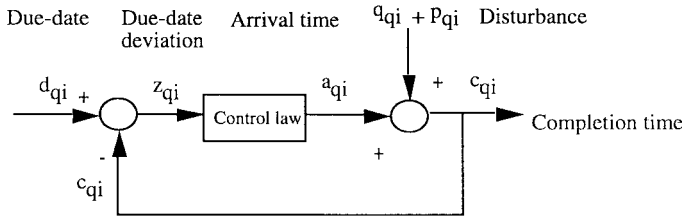


Fig. 1. A generic local arrival-time controller.

all of the relevant global information into one variable and couples the controllers. As arrival times change dynamically, the disturbance and the coupling between controllers will also change dynamically.

The arrival time controller in each part uses a local control law and local feedback for adjusting its arrival time. No assumptions are made by arrival time controllers about the number of parts, processing sequence, processing times, due dates of other parts, machine(s) used, or queuing discipline. This results in a high level of local autonomy and control distribution, and a system retaining minimal global information; these are design principles of heterarchical systems.

The choice of the local control law used in each part will have a significant impact on system stability, dynamics, and analyzability. The fact that there is no "plant" in the traditional sense can be used advantageously in selecting a controller because it is well known that for a constant disturbance input, error will asymptotically become zero when integral control is used for a plant with no dynamics. Therefore the following integral control law was chosen:

$$a_j(t) = k_j \int_0^t (d_j(\tau) - c_j(\tau)) d\tau + a_j(0) \quad (4)$$

where k_j is the controller gain for the j th part in the system, and $a_j(0)$ is an arbitrary initial condition. It should be emphasized that each of the N parts in the systems has its own controllers, and that all of the controllers in a system compute their arrival times in parallel. All controllers are considered equal. This simple control algorithm only requires local feedback for computing the arrival time. When there is sufficient capacity to meet processing demands and when there is no queuing, disturbances due to queuing will be zero, and error between due-date and completion time (the due-date deviation of the part) will asymptotically become zero. When there is insufficient capacity, this control law adjusts arrival time to make the time average of the part due-date deviation zero with equal penalty for earliness and tardiness.

The dynamics of a system of integral arrival-time controllers will depend on the region in which its arrival time vector lies. Assuming that due-dates and processing times stay constant, when an arrival time vector for n parts is in the decoupled region, all queuing times are zero and the dynamical model can be expressed as

$$\dot{\mathbf{a}} = f_c(\mathbf{a}), \quad \mathbf{a} \in M_c \quad (5)$$

where $\mathbf{a} = [a_1 \ a_2 \ \dots \ a_n]^T$ and $f_c(\mathbf{a})$ is given by

$$f_c(\mathbf{a}) = - \begin{bmatrix} k_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & k_2 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & k_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} - \begin{bmatrix} k_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & k_2 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & k_n \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} + \begin{bmatrix} k_1 & 0 & 0 & 0 & \dots & 0 \\ 0 & k_2 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & k_n \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix}.$$

As long as the arrival times stay in the decoupled region, the dynamic response of each part can be analyzed independently. Solving (5) gives

$$a_1(t) = (d_1 - p_1)(1 - e^{-k_1 t}) + a_1(0)e^{-k_1 t} \quad (6)$$

$$a_2(t) = (d_2 - p_2)(1 - e^{-k_2 t}) + a_2(0)e^{-k_2 t} \quad (7)$$

$$\dots \quad (8)$$

When the arrival time vector for n parts is in the dead-zone region, the dynamical model can be expressed as

$$\dot{\mathbf{a}} = f_z(\mathbf{a}), \quad \mathbf{a} \in M_z \quad (9)$$

where $\mathbf{a} = [a_{q1} \ a_{q2} \ \dots \ a_{qn}]^T$ and $f_z(\mathbf{a})$ is given by

$$f_z(\mathbf{a}) = \begin{bmatrix} k_{q1} & 0 & 0 & 0 & \dots & 0 \\ 0 & k_{q2} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{qn} & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} a_{q1} \\ a_{q2} \\ \vdots \\ a_{qn} \end{bmatrix} - \begin{bmatrix} k_{q1} & 0 & 0 & 0 & \dots & 0 \\ k_{q2} & k_{q2} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{qn} & k_{qn} & k_{qn} & k_{qn} & \dots & k_{qn} \end{bmatrix} \begin{bmatrix} p_{q1} \\ p_{q2} \\ \vdots \\ p_{qn} \end{bmatrix} + \begin{bmatrix} k_{q1} & 0 & 0 & 0 & \dots & 0 \\ 0 & k_{q2} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & k_{qn} \end{bmatrix} \begin{bmatrix} d_{q1} \\ d_{q2} \\ \vdots \\ d_{qn} \end{bmatrix}.$$

Solving (9) gives

$$a_{q1}(t) = (d_{q1} - p_{q1})(1 - e^{-k_{q1} t}) + a_{q1}(0)e^{-k_{q1} t} \quad (10)$$

$$a_{q2}(t) = \frac{k_{q2}}{k_{q1}} (d_{q1} - p_{q1})(1 - e^{-k_{q1} t}) - \frac{k_{q2}}{k_{q1}} a_{q1}(0) \cdot (1 - e^{-k_{q1} t}) + k_{q2}(d_{q2} - d_{q1} - p_{q2})t + a_{q2}(0) \quad (11)$$

$$\begin{aligned}
a_{qn}(t) = & \frac{k_{qn}}{k_{q1}} (d_{q1} - p_{q1}) (1 - e^{-k_{q1}t}) \\
& - \frac{k_{qn}}{k_{q1}} a_{q1}(0) (1 - e^{-k_{q1}t}) \\
& + k_{qn} (d_{q2} - d_{q1} - p_{q2} - \dots - p_{qn})t \\
& + a_{qn}(0). \tag{12}
\end{aligned}$$

As the arrival time vector changes dynamically the processing sequence can change. The function $f_z(a)$ is continuous in each of the $n!$ regions corresponding to the $n!$ possible processing sequences. In the next section, Filippov's approach is used to solve the discontinuous differential equations that describe the dynamics in the discontinuous region.

IV. DYNAMICS IN THE DISCONTINUOUS REGION

In 1960 Filippov generalized the concept of solution of differential equations such as

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)), \quad \text{where } \mathbf{x}(t) \in \mathbf{R}^n \tag{13}$$

where the function f is possibly discontinuous in \mathbf{x} and t and the solution encounters a line (or surface) of discontinuity infinitely often in a finite time [12]. Many of the results in the classical theory of differential equations have been shown to be valid for these discontinuous differential equations [13]. Filippov's approach can be used here to solve the discontinuous differential equations that describe the dynamics in the discontinuous region with an arbitrary number of parts and a single machine. Briefly, the solution of such an equation is an absolutely continuous function whose derivative vector is a weighted average of all its possible values in the neighborhood. The solution is found by taking the intersection of the discontinuity surface with the closure of the convex hull of all possible derivative vectors in the neighborhood. The key issue in the dynamics of this discontinuous system is that the velocity of the trajectory may not be uniquely determined by the state of the system on the discontinuity surface but belongs to a set of feasible values that depends "loosely" on the state. Therefore the main issue in the solution of the discontinuous differential equations is to find the velocity $\dot{\mathbf{x}}(t)$ for solutions that remain on a discontinuity surface or on an intersection of discontinuity surfaces.

Consider (13) where f is a piecewise continuous function in the domain $G \in \mathbf{R}^n$ and M is a set of points, with measure zero, where f is discontinuous. The solution specified by Filippov [13] is an absolutely continuous function $\mathbf{x}(t)$ defined on an interval $I: [0, T]$ for which

$$\dot{\mathbf{x}}(t) \in F(\mathbf{x}(t)) \tag{14}$$

almost everywhere on I . If the point $\mathbf{x}(t) \in M$ then $F(\mathbf{x}(t)) = f(\mathbf{x}(t))$ which is a single point in \mathbf{R}^n as in classical solutions to continuous differential equations. More importantly, if the point $\mathbf{x}(t) \in M$ then $F(\mathbf{x}(t))$ is given by

$$F(\mathbf{x}(t)) = \bigcap_{\delta > 0} \bigcap_M \overline{\text{Conv}} f(\beta(\mathbf{x}(t), \delta) - M). \tag{15}$$

The set $F(\mathbf{x}(t))$ depends on the behavior of the system in the δ -neighborhood of the discontinuity and is the smallest

convex closed set containing all the limit values of the function $f(\mathbf{x}^*(t))$ as $\mathbf{x}^*(t) \rightarrow \mathbf{x}(t)$, with t constant and $\mathbf{x}^*(t) \notin M$. This is accomplished in (15) by taking the intersection over all sets with $\delta > 0$. In (15), "minus the set M of measure zero" allows possible "misbehavior" of f on M to be ignored in forming the set $F(\mathbf{x}(t))$. The intersection with the set M constrains the solution to lie in M . If the point $\mathbf{x}(t) \in M$ lies on the intersection of two or more domains G_i ($i = 1, \dots, k$) then all points $f_i(\mathbf{x}(t))$ will be contained in $F(\mathbf{x}(t))$ which could be a line segment, convex polygon, or a convex polyhedron with vertices $f_i(\mathbf{x}(t))$, where

$$f_i(\mathbf{x}(t)) = \lim_{\substack{\mathbf{x}^*(t) \in G_i \\ \mathbf{x}^*(t) \rightarrow \mathbf{x}(t)}} f_i(\mathbf{x}^*(t)) \tag{16}$$

with t constant.

Applying Filippov's approach to distributed arrival control, the solution of the following differential equation:

$$\dot{\mathbf{a}}(t) = f_s(\mathbf{a}(t)), \quad \text{where } \mathbf{a}(t) \in M_s \tag{17}$$

is specified as a solution to

$$\dot{\mathbf{a}}(t) \in F_s(\mathbf{a}(t)). \tag{18}$$

In (17), $f_s(\mathbf{a}(t))$ is a piece-wise continuous function in \mathbf{R}_n and f_s is discontinuous in M_s which is a union of a finite number of hyperplanes, and has a measure zero in \mathbf{R}^n . In the discontinuity region the set $F_s(\mathbf{a}(t))$ will depend on the behavior of the system in the δ -neighborhood of $\mathbf{a}(t)$. If n arrival times are equal at $\mathbf{a}(t)$ then $n!$ sequences are possible in its δ -neighborhood, the point $\mathbf{a}(t)$ lies at the boundary of $n!$ deadzone domains, and the convex hull of $f_z(\mathbf{a}^*(t))$ in the corresponding deadzones needs to be considered as prescribed by Filippov's approach in (15). From (4) it is evident that $f_z(\mathbf{a}^*(t)) = \mathbf{Kz}(t)$ where \mathbf{K} is the diagonal gain matrix with diagonal $(\mathbf{K}) = [k_1 \ k_2 \ k_3 \ \dots \ k_n]$ and $\mathbf{z}(t)$ is the due date deviation vector for the corresponding processing sequence.

The dynamics of the system in the discontinuity region will be determined by the geometry of the convex hull of $\mathbf{Kz}(t)$ and the manner in which the convex hull intersects M_s . When n arrival times are equal, M_s is an intersection of hyperplanes which is a line in n -dimensional space. It can be shown that the convex hull of $\mathbf{Kz}(t)$ lies in an $(n-1)$ -dimensional plane with normal in the direction of the vector $\mathbf{K}^{-1}\mathbf{p}$ and the equation for this plane can be expressed as [9]

$$(\mathbf{K}^{-1}\mathbf{p})^T \dot{\mathbf{a}}(t) = \mathbf{p}^T \mathbf{z}(t). \tag{19}$$

Therefore $\dot{\mathbf{a}}(t)$ vector starts at $\mathbf{a}(t)$ and is constrained to end at the intersection of M_s and the convex hull of $\mathbf{Kz}(t)$ which is a unique point if it exists. If the intersection point does not exist then the arrival time trajectory will leave the discontinuity region instantaneously [13]. Hence in the discontinuous region $\dot{\mathbf{a}}(t)$ is unique, $\mathbf{a}(t)$ is unique, implying that the solution to the differential equation in (17) is unique whenever it exists. Since the end point of $\dot{\mathbf{a}}(t)$ lies in the discontinuity region M_s , all n components of $\dot{\mathbf{a}}(t)$ will be equal. Using this and (19), each of the n components of $\dot{\mathbf{a}}(t)$, which are equal, can be expressed as a scalar

$$\ddot{a}(t) = \frac{\mathbf{p}_z^T(t)}{(\mathbf{K}^{-1}\mathbf{p})^T [1 \ \dots \ 1]}. \tag{20}$$

Substituting $z(t) = \mathbf{d} - \mathbf{a}(t) - \mathbf{r}$ in (20) gives

$$\dot{a}(t) = \tilde{k}(\alpha - a(t)) \quad (21)$$

where

$$\tilde{k} = \frac{p_{q1} - q_{q2} + \dots + p_{qn}}{\frac{p_{q1}}{k_{q1}} + \frac{p_{q2}}{k_{q2}} + \dots + \frac{p_{qn}}{k_{qn}}} \quad (21a)$$

and

$$\alpha = \frac{\mathbf{p}^T(\mathbf{d} - \mathbf{r})}{p_1 + p_2 + \dots + p_n} \quad (21b)$$

\mathbf{r} is the disturbance vector $[(q_{q1} + p_{q1})(q_{q2} + p_{q2}) \dots (q_{qn} + p_{qn})]$ and α can be rewritten as

$$\alpha = \frac{(p_{q1}(d_{q1} - p_{q1}) + p_{q2}(d_{q2} - (p_{q1} + p_{q2})) + \dots + p_{qn}(d_{qn} - (p_{q1} + \dots + p_{qn})))}{p_{q1} + p_{q2} + \dots + p_{qn}}.$$

Solving (21) gives

$$a(t) = \alpha(1 - e^{-\tilde{k}t}) \quad (22)$$

for all components of $\mathbf{a}(t)$. This solution holds for all n parts as long as their arrival times are equal and in the discontinuity region. Interestingly, dynamics in the discontinuity region can be modeled as a scalar regardless of n . The exponential convergence rate \tilde{k} depends on processing times \mathbf{p} and gains \mathbf{K} . Arrival time trajectories are unique in the discontinuity region.

V. QUEUE FRAGMENTS WITH MULTIPLE REGIONS

A queue fragment can consist of parts in the dead-zone region as well as parts in the discontinuity region. The solution in (10)–(12), for all parts in the dead-zone region, and the solution in (22), for all parts in the discontinuity region, needs to be modified for this case. Consider a given queue fragment consisting of two subfragments. One subfragment consists of h parts in the discontinuity region with $a_s(t) = a_{s1}(t) = a_{s2}(t) = \dots = a_{sh}(t)$. The other subfragment consists of i parts in the dead-zone region with $a_{q1}(t) < a_{q2}(t) < \dots < a_{qi}(t)$. Further, suppose as the first possibility that $a_s(t) < a_{q1}(t)$ and the discontinuity subfragment is processed before the dead-zone fragment. The first part in the dead-zone subfragment becomes the $(h+1)$ th part in the queue fragment and incurs a queuing time of

$$q_{q1}(t) = a_s(t) + p_s - a_{q1}(t) \quad (23)$$

where

$$p_s = \sum_{x=1}^h p_{sx}.$$

Equations (10)–(12) then become

$$a_{q1}(t) = \frac{k_{q1}}{\tilde{k}} (\alpha) (1 - e^{-\tilde{k}t}) - \frac{k_{q1}}{\tilde{k}} a_s(0) (1 - e^{-\tilde{k}t}) + k_{q1}(d_{q1} - \alpha - p_s - p_{q1})t + a_{q1}(0) \quad (24)$$

$$a_{q2}(t) = \frac{k_{q2}}{\tilde{k}} (\alpha) (1 - e^{-\tilde{k}t}) - \frac{k_{q2}}{\tilde{k}} a_s(0) (1 - e^{-\tilde{k}t}) + k_{q2}(d_{q2} - \alpha - p_s - p_{q2})t + a_{q2}(0) \quad (25)$$

$$a_{qi}(t) = \frac{k_{qi}}{\tilde{k}} (\alpha) (1 - e^{-\tilde{k}t}) - \frac{k_{qi}}{\tilde{k}} a_s(0) (1 - e^{-\tilde{k}t}) + k_{qi}(d_{qi} - \alpha - p_s - p_{q1} - p_{q2} - \dots - p_{qi})t + a_{qi}(0) \quad (26)$$

where α and are the same as in (21). In this case, (22) holds for the discontinuity subfragment because it is unaffected by the dead-zone subfragment.

Consider the other possibility where $a_s > a_{qi}$ and the discontinuity subfragment is processed after the dead-zone fragment. All parts in the discontinuity subfragment incur a queuing time and the due date deviation vector in (20) is given by

$$z(t) - \mathbf{d} - \mathbf{a}_q(t) - \mathbf{p}_z - \mathbf{r} \quad (27)$$

where \mathbf{p}_z is a $1 \times h$ vector with each element is equal to

$$\sum_{x=1}^i p_{qx}$$

and $\mathbf{a}_q(t)$ is a $1 \times h$ vector with each element equal to $a_{q1}(t)$. From (20) and (27)

$$\dot{a}_{qi}(t) = \tilde{k}(\alpha' - a_{qi}(t)) \quad (28)$$

where

$$\alpha' = \frac{\mathbf{p}^T(\mathbf{d} - \mathbf{p}_z - \mathbf{r})}{p_1 + p_2 + \dots + p_n}.$$

From (10) and (28), the solution for each of the arrival times in the discontinuity region is

$$a_q(t) = -\frac{\tilde{k}}{k_{q1}} (d_{q1} - p_{q1}) (1 - e^{-k_{q1}t}) - \frac{\tilde{k}}{k_{q1}} a_{q1}(0) \cdot (1 - e^{-k_{q1}t}) - \tilde{k}(d_{q1} - p_{q1} + \alpha')t + a_s(0). \quad (29)$$

In this case (10)–(12) hold for the dead-zone subfragment because it is unaffected by the discontinuity subfragment. This analysis of queue fragment with multiple regions can be extended to more than two subfragments by modifying the queuing time and due-date deviations appropriately and unique arrival time trajectories can be derived for single machine systems with arbitrary number of parts.

VI. STEADY-STATE ARRIVAL TIMES

When $\dot{\mathbf{a}}(t)$ becomes zero, the system converges to a steady-state value \mathbf{a}_{ss} . In the decoupled region this occurs when $z(t) = 0$, i.e., when all due-date deviations are zero. Under this condition the due dates and completion times are equal and this can happen only when due dates are feasible, and the equilibrium point a_{ssi} for a given part can be expressed as

$$a_{ssi} = d_i - p_i. \quad (30)$$

For a given d_i and p_i , a_{ssi} is unique and independent of \mathbf{K} provided each $k_i > 0$. The steady-state sequence is also unique and is determined by \mathbf{d} . Furthermore, from (6)–(8), in the decoupled region $a(t)$ exponentially converges and the decoupling allows the dynamic equations to be treated as scalar equations in the vicinity of the equilibrium point.

TABLE I
DATA FOR TWO-PART ONE-MACHINE EXAMPLE

	Due-Date	Processing Time	Gain	Initial Arrival Time
Part 1	8	2	1	0
Part 2	7.5	1	1	9

Due-dates are said to be infeasible when all due-dates cannot be met simultaneously. Under such a condition all arrival time trajectories are directed toward the discontinuity region and once a trajectory reaches the discontinuity region it stays in the region [13]. From (20), in the discontinuous region $\dot{\mathbf{a}}(t)$ becomes zero when

$$\mathbf{p}^T \mathbf{z}(t) = 0. \quad (31)$$

In other words $\dot{\mathbf{a}}(t)$ becomes zero when all possible $\mathbf{z}(t)$ vectors lie in a plane and $\mathbf{a}(t)$ has converged to its steady-state value \mathbf{a}_{ss} in the discontinuity region. Since $\mathbf{z}(t) = \mathbf{d} - \mathbf{a}(t) - \mathbf{r}$, the steady-state condition in the discontinuity region can be expressed as

$$\mathbf{p}^T \mathbf{a}_{ss} = \mathbf{p}^T (\mathbf{d} - \mathbf{r}). \quad (32)$$

Equation (32) implies that $\mathbf{a}(t)$ will converge to its steady-state value \mathbf{a}_{ss} when the projections of the \mathbf{a}_{ss} vector and the $(\mathbf{d} - \mathbf{r})$ vector on a line in the direction of \mathbf{p} are equal. Making use of the fact that all arrival times are equal in the discontinuity region, \mathbf{a}_{ss} can be expressed as a scalar equation where each of its n components is given by

$$a_{ss} = \alpha - \frac{\mathbf{p}^T \mathbf{d} - \mathbf{p}^T \mathbf{r}}{\mathbf{p}^T [1 \dots 1]} \quad \mathbf{a}_{ss} \in M_s, \quad \mathbf{d} \in \text{Infeasible}. \quad (33)$$

For a given set of due dates and processing times \mathbf{a}_{ss} will be unique and independent of \mathbf{K} provided each $k_i > 0$. It should be pointed out that even though the queue fragment converges to a unique arrival time in the steady state, the sequence is not unique because the arrival time trajectory can be in any of the $n!$ regions in its δ -neighborhood corresponding to $n!$ possible sequences.

VII. EXAMPLE

Neither the fundamental system dynamics nor the method of analysis are affected by the number of parts in the system. Therefore, it is sufficient to consider a case with one machine and two parts having the processing times, infeasible due dates, and initial arrival times shown in Table I. Initially, the system dynamics are in the decoupled region. The arrival times are far apart, the part dynamics are decoupled, and the arrival times exponentially converge toward their steady-state values as shown in Fig. 2. These trajectories are predicted by the decoupled model given by (6) and (7). During this time the completion times also converge exponentially as shown in Fig. 3. After time t_1 , the systems dynamics are in the dead-zone region and the response in c_2 in Fig. 3 illustrates the dead-zone effect. Trajectories in this region are predicted by the dead-zone model given by (10) and (11). After time t_2 ,

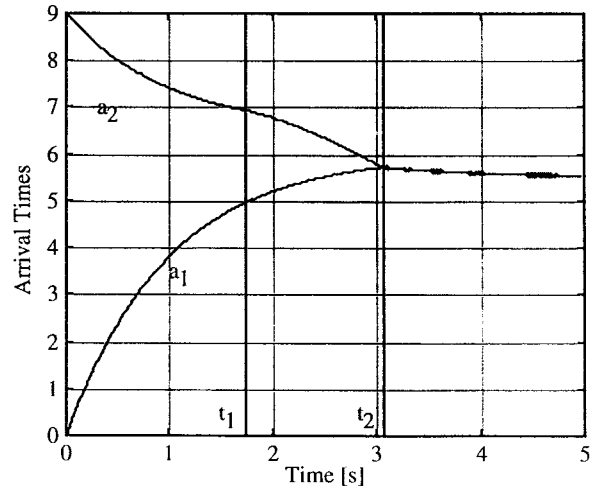


Fig. 2. Arrival time trajectories.

the systems dynamics are in the discontinuous region, a_1 and a_2 are equal causing the corresponding completion times to oscillate discontinuously. Trajectories in this region are predicted by the discontinuity model given by (22), with the steady-state arrival time of 5.5 as predicted using (21b) and an exponential convergence rate of one as predicted using (21a).

Even though the system converges to a unique point in arrival time space in the steady-state, the arrival time trajectory can be in either of the two directions corresponding to the possible sequences in its infinitesimal neighborhood of this point. This is seen as high-frequency “chatter” in Fig. 3, which is similar to “chatter” in controller output in sliding mode control. In this case the “chatter” can be considered advantageous because it allows the arrival time trajectory to visit many possible sequences within its infinitesimal neighborhood. In this example, there are two possible sequences. Which sequence represents the better schedule depends up on the function used to evaluate merit.

VIII. CONCLUSIONS

Increasing global competition, increasing variety of products, and decreasing production volumes are increasing the need for intelligent manufacturing shop-floor control systems with reduced complexity and high flexibility [5]. The arrival time control approach is not only highly distributed because it uses local variables and local control laws, but control decisions are made at least an order of magnitude faster than events on the shop-floor making it suitable for real-time event-level control in low-volume/high-variety heterarchical manufacturing systems with batch sizes as low as unity. Furthermore, the approach allows shop-floor control problem

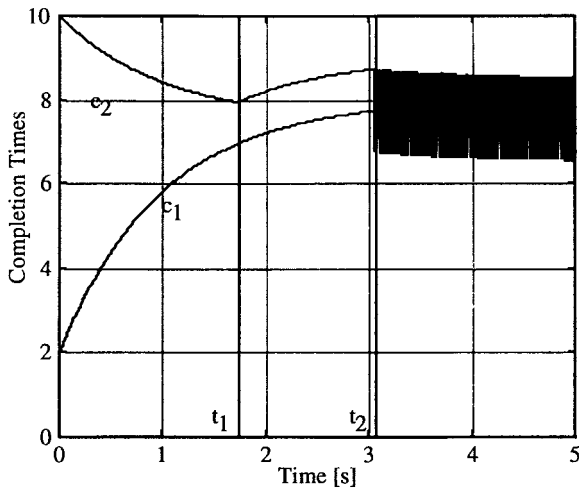


Fig. 3. Completion time trajectories.

to be modeled using differential equations and offers the opportunity to apply to shop-floor control the rich theory of analysis, control, and optimization that have been so successfully applied to dynamical systems governed by differential equations. The analysis in this paper helps to explain the complex dynamics of highly distributed, real-time scheduling systems such as the heterarchical systems described in [3].

The scope of this work was restricted to a single machine with arbitrary number of parts. However, in many industrial scheduling applications this may not be adequate. Future research should focus on more general configurations with multiple machines and multiple manufacturing cells. Future generalizations should also include modeling and analysis of dynamics such as varying processing times and tool changeover times. Simulation studies have indicated that the

arrival time approach can simultaneously reduce mean and variance of due-date deviation which makes it an attractive approach for real-time control of Just-In-Time manufacturing systems [9], [11]. There is a need to rigorously model the performance the arrival time control approach in order to gain a deeper understanding of such highly distributed systems.

REFERENCES

- [1] S. C. Graves, "Review of production scheduling," *Operations Res.*, vol. 29, 1982, pp. 646-675.
- [2] D. M. Diltz *et al.*, "The evolution of control architectures for automated manufacturing systems," *J. Manufacturing Syst.*, vol. 10, no. 1, 1991, pp. 79-93.
- [3] N. A. Duffie and V. V. Prabhu, "Real-time distributed scheduling of heterarchical manufacturing systems," *J. Manufacturing Syst.*, vol. 13, no. 2, 1994, pp. 94-107.
- [4] ———, "Heterarchical control of highly distributed manufacturing systems," *Int. J. Comput. Integrated Manufacturing*, vol. 9, no. 4, 1996, pp. 270-281.
- [5] J. J. Solberg and R. L. Kashyap, "ERC research in intelligent manufacturing systems," *Proc. IEEE*, vol. 81, Jan. 1993, pp. 25-41.
- [6] V. V. Prabhu, "Distributed real-time manufacturing systems scheduling," M.S. thesis, Univ. Wisconsin, Madison, 1991.
- [7] M. Caraminis and A. Sharifnia, "Near optimal manufacturing flow controller design," *Int. J. Flexible Manufacturing Syst.*, vol. 3, 1991, pp. 321-336.
- [8] S. G. Gershwin, *Manufacturing Systems Engineering*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [9] V. V. Prabhu, "Real-time distributed arrival time control of heterarchical manufacturing systems," Ph.D. dissertation, Univ. Wisconsin, Madison, 1995.
- [10] V. V. Prabhu and Duffie, "Modeling and analysis of nonlinear dynamics in autonomous heterarchical manufacturing systems control," *Ann. CIRP*, vol. 44, no. 1, 1995, pp. 425-428.
- [11] ———, "Modeling and analysis of heterarchical manufacturing systems using discontinuous differential equations," *Ann. CIRP*, vol. 45, no. 1, 1996, pp. 445-448.
- [12] A. F. Filippov, "Differential equations with discontinuous righthand sides," *Amer. Math. Soc. Translations*, vol. 42, 1960, pp. 199-231.
- [13] ———, *Differential Equations with Discontinuous Right-hand Sides*. Boston, MA: Kluwer, 1988.