Dynamic Setup Scheduling and Flow Control in Manufacturing Systems *

A. Sharifnia † M. Caramanis † S. B. Gershwin ‡

Abstract

In this paper we propose a method for flow control of parts in a manufacturing system with machines that require setups. The setup scheduling problem is investigated in the context of a multilevel hierarchy of discrete events with distinct frequencies. The higher level of the hierarchy calculates a target trajectory in the surplus/backlog space of the part types which must be tracked at the level of setups. We consider a feedback setup scheduling policy which uses *corridors* in the surplus/backlog space of the part types to determine the timing of the setup changes in order to guide the trajectory in the desired direction. An interesting case in which the target trajectory leads to a target point (e.g. a hedging point) is investigated in detail. It is shown that in this case the surplus/backlog trajectory at the setup level can lead to a limit cycle. Conditions for linear corridors which result in a stable limit cycle are determined.

1 Introduction

The setup scheduling problem arises in manufacturing systems when a significant cost and/or time is required to set up a production resource for the processing of different part types. The real-time setup scheduling problem is to decide which part type has to be processed next and when the resource has to stop its current operations and make a setup change to begin the processing of that part type.

This problem has been investigated extensively in traditional production scheduling literature. The problem is often referred to as *Economic Lot Scheduling Problem* (ELSP). A review of the models used for ELSP is given by Elmaghraby [1]. Some of the more recent works in this area are those of Goyal [2], Dobson [3], Jones [4], and Carreno [5]. These works address the production scheduling of several products on one or more identical machines. The demand rate for products is assumed constant, time horizon is infinite, and no backlog is permitted. Since everything is in steady-state, the setup scheduling problem is reduced to

*The research reported here has been supported by the National Science Foundation under grants DMC-8615560 and DDM-8914277.

†Department of Manufacturing Engineering, Boston University, Boston, MA 02215.

‡Department of Mechanical Engineering, Massachusetts Institute of Technology.

finding optimal lot sizes that minimize the average setup and inventory carrying costs per unit time.

These works focus on the setup scheduling problem in isolation and under steady-state conditions. They do not address non-steady-state situations caused by disturbances such as machine failures, scheduled maintenance, etc. in a manufacturing system. Under non-steady-state conditions the problem cannot be reduced to one of lot-sizing since the lot sizes will not remain constant over time. The objective of this paper is to investigate the setup scheduling problem in a feedback control framework, which could respond to various discrete events in a manufacturing system.

A feedback control scheduling framework developed by Kimemia and Gershwin [6] suggests that the *flow rate* of parts through a manufacturing system should be the focus of attention, rather than individual jobs or operations. This framework assumes that manufacturing systems should be operated to meet a multiple-part-type *demand rate* as closely as possible. It focuses attention on random failures and repairs of machines affecting the production capacity. Based on earlier works by Rishel [7] and Olsder and Suri [8], they determine the structure of the optimal policy for control of the flow rates of various part types into the system. It is shown that the optimal control has a special structure, described by a hedging point strategy, in which a positive surplus of part types is maintained during times of excess capacity availability to *hedge* against future capacity shortages brought about by machine failures. This approach is often referred to as *flow control* in the literature.

Since the early efforts of Kimemia and Gershwin, the flow control problem has received considerable attention. Gershwin, Akella, and Choong [9] devised approximation techniques for solving the flow control design problem. Akella and Kumar [10] and Bielecki and Kumar [11] derived the exact solution for a special case involving two machine states (up and down) and one part type. Sharifnia [12] solved the problem for the case of many machine states but still only one part type. Algoet [13] characterized the steady state probability density function for a hedging point policy, but the resulting differential equation remains intractable. Malhame and Boukas [14] formulated the one-part-type problem as a Markov renewal process which allows the transient analysis of the surplus under the hedging point strategy. Haurie [15] established a relationship between a hedging point in flow control and the turnpike property of the infinite horizon control problems. Caramanis and Sharifnia [16] proposed analytically tractable near-optimal controllers that seem promising. Caramanis and Liberopoulos [17] have used perturbation analysis to design an optimal flow controller.

Recently, a broader framework for planning and scheduling of manufacturing systems was proposed by Gershwin [18, 19, 20] in which decisions and events in a production system are grouped into various levels of a hierarchy according to their characteristic time scale. Each level of the hierarchy specializes in the detailed modeling of controllable or stochastic events which occur at roughly the same frequency. If frequencies of various levels are well separated, the modeling interactions among levels is simplified and each level can be represented as a flow control problem.

Work on flow control, cited above, assumes *perfectly flexible* machines capable to change from producing one part type to another instantaneously. Recently two excellent papers by

Perkins and Kumar [21] and Kumar and Seidman [22] addressed the issue of setup strategies, when setup times are not zero. They investigated the performance of *distributed* setup scheduling policies in which the setup decisions for each machine are made based only on that machines backlog of various parts. Since they are concerned with distributed policies, an important issue is that of stability, i.e. whether work-in-process (WIP) will remain under control. They show that some distributed setup policies are stable for acyclic manufacturing systems, but the same policies are unstable for non-acyclic systems. They propose means for stabilizing these policies. Chase and Ramadge [23] find improved lower bounds on WIP by allowing idleness for the machines. Lou, Sethi, and Sorger [24] consider the setup problem for a single machine with random breakdowns, and find conditions on machine's capacity for the expected WIP to remain bounded.

In this paper we will investigate setup scheduling policies in the hierarchical flow control paradigm. The basic idea is to generate production and setup rates at the higher (lower frequency) level of the hierarchy, and then use those rates to determine setup schedules at the level of setups. An important advantage of this approach is that it can optimize setup frequencies at the higher level of the hierarchy by taking into account considerations such as setup costs and WIP. In addition, these frequencies can change as a function of the higher levels' discrete state. Another important feature of our proposed policy is that it does not require "clearing" of a backlog in each setup as assumed in the works cited above. These are the main features that distinguish our work from the previous works on feedback setup scheduling.

In Section 2 we review the hierarchical framework with setups constituting one level of the hierarchy. The advantages of the hierarchical approach will be discussed. Also, we will explain how the production capacity sets at the setup level and the level immediately preceding it are determined. In Section 3 we propose a setup policy to track the target production rate determined at the hierarchical level immediately preceding the setup level. This policy uses *corridors* in the surplus space of part types to determine the exact time of setup changes. Section 4 describes possible limit cycles in the production surplus for certain setup corridors. In Section 5 we find conditions for corridors to yield stable limit cycles. Section 6 gives some simulation results, and we conclude in Section 7.

2 Setup Scheduling in the Hierarchical Framework

2.1 Review of Hierarchical Framework

We investigate the setup scheduling problem in the context of the hierarchy proposed by Gershwin. In this hierarchy each level specializes in the detailed modeling of controllable or stochastic events which occur at roughly the same frequency. The least frequent events are assigned to the highest level of the hierarchy while events with greater frequencies are assigned to progressively lower levels of the hierarchy. Assuming that the frequencies of the various levels are well separated, the planning and scheduling task for the system can be decomposed into smaller problems, each addressing an individual level of the hierarchy. This is done as follows: At each level, the states of all higher levels change very slowly and hence can be assumed fixed. The states of the lower levels change very often and are therefore described by the expected rates at which they change. Each level schedules its own controllable events and also determines the *rates* at which the controllable events of the lower levels must be scheduled. These rates are used as targets at the lower level for scheduling that level's controllable events.

We assume that the setup changes constitute one level of a hierarchy (Figure 1). Our analysis is general and does not assume any special structure for the hierarchy. For illustrative purposes, however, we suppose that the level above setups corresponds to the failures and repairs of machines, and these occur much less frequently than setup changes. We also assume that the level below setups schedules the actual dispatching of parts for production, and this occurs much more frequently than setup changes. An example of a system like this is Case 3A in [19].

The failure-repair level determines activity rate targets for the setup level at each failure-repair state of the system. These rates include the setup change rates as well as the production rates of various part types. In deciding the setup change rates the higher level may take into account not only its production rate targets, but also considerations such as cost of setup changes (e.g. labor) and the work-in-process (WIP) associated with a given setup change rate. This level uses an optimization model to calculate the rate targets for the setup level. These target rates depend on the discrete state of this level (failure-repair state of machines) and, in general, are different for different machine states.

The setup level must decide the exact timing of the setup changes and also must determine the production rate of the part types in each setup in order to track the target rates set by the failure-repair level. The production rates at each setup are then passed down to the lower level which has to schedule the parts dispatch times so as to track these production rates.

The generic task at each level of the hierarchy includes scheduling of the corresponding controllable events to track the higher level target rates as well as refining the targets for the lower level. Therefore, each level has to solve a flow control problem.

2.2 The Capacity Set in the Presence of Setups

One of the building blocks for flow control at each level of the hierarchy is the capacity set at that level. This is the set of all feasible points in the space of higher frequency (lower level) activity rates. Target values of those rates are determined at that level. Of course, the capacity set will depend on the current state of the discrete variables associated with that level, and a set must be found for each value of this state.

We investigate the capacity set at the setup and the higher level (the level immediately above setups). Let us consider a simple case in which we have a single machine that does operations on two part types. The machine requires a significant amount of time to switch its production form part type 1 to part type 2 and vice versa. At the level of setups we are concerned with the capacity of the machine at a given setup. In setup i, where part type i (i=1,2) is being produced, the capacity is the closed interval $[0, \hat{u}_i]$ where $\hat{u}_i$ is the maximum production rate of part type i achievable by the machine (Figure 2.a).

Next consider the same machine at the time scale of the higher level of the hierarchy i.e. the failure-repair level. At this time scale, setups change very often and therefore the relevant decisions include the *rate* of setup changes as well as the production rates, for each machine failure-repair state. When the machine is up, the capacity set will be a three-dimensional set with one dimension for the production rate of each part type and the third dimension for the rate of setup changes (Figure 2.b). The capacity set is $\mathbf{0}$ (the origin) when the machine is down.

In general the setup rate determined at the higher level may vary as a function of the state of that level. For instance, we may decide to do fewer setups when we are behind the production target for part types and tolerate wide swings in part type backlogs in order to approach the target faster. In contrast, when we are close to the target, it may be preferable to allocate more capacity to setup changes to avoid large swings in the part type backlogs.

We introduce an important simplification of the capacity sets at the higher level which makes it possible to come up with a fairly simple setup scheduling policy at the setup level. This involves fixing the setup rate at some average value. If the setup rate is fixed the capacity set (for the working state) will be reduced to a two-dimensional set as shown in Figure 2.b by the shaded triangle. Note that if the machine were perfectly flexible, i.e. if it required no time to change its setup, the capacity set would be the triangle in (u_1, u_2) plane in the same figure. Therefore, the simplified capacity set with nonzero setup times is a reduced copy of the capacity set corresponding to a perfectly flexible machine.

In a general case with many machines, we can similarly consider an average setup rate for each machine with nonzero setup times, and then aggregate over machine capacity sets to find the total capacity set. The resulting capacity set will again be a reduced copy of the set corresponding to the case where all machines are flexible.

As mentioned earlier, this simplification allows us to devise a simple scheduling policy at the setup level. This policy does allow for variation in the setup rates at the setup level. Of course, for consistency the average setup rate must be equal to the (fixed) setup rate target determined at the higher level. This policy is described in the next section.

3 The Setup Scheduling policy

The task of the setup level is to schedule the setup changes and to determine the activity rate targets for the lower level in each setup. In doing so, the setup level must follow the target rates set by the higher level as closely as possible, subject to its own capacity constraints.

We define $\mathbf{x}$ to be the vector of *surpluses* in various activities. The surplus of an activity is defined as the difference between the cumulative level of the activity performed so far and the cumulative demand for the activity. For instance, the surplus for the production of a part type represents the excess of production over demand for that part type (if this quantity is negative it represents the backlog for the part type). Therefore, the surplus dynamics are described by :

$$\dot{\mathbf{x}} = \mathbf{u} - \mathbf{d}$$

where $\mathbf{u}$ is the vector of activity rates (i.e. number of each activity performed per unit time), and $\mathbf{d}$ is the vector of demand (target) rates for the activity.

Now consider the space of $\mathbf{x}$ at the setup level. Given an initial value for $\mathbf{x}$, we can draw a target trajectory for $\mathbf{x}$ based on the activity target rates handed down from the higher level. We refer to this trajectory as the *target trajectory* at the setup level. The task of the setup level is to track this trajectory as closely as possible. As argued below, it is reasonable to assume that this trajectory is piecewise linear with a constant (but different) velocity on each piece, and may eventually terminate at a specific point in the $\mathbf{x}$-space. The surplus becomes stationary at that point(called a target or hedging point) until the next occurrence of a discrete event at the level above setups.

3.1 Piecewise Linear Target Trajectories

Under reasonable assumptions and approximations each level of the hierarchy will generate piecewise linear trajectories in the surplus space which must be tracked by the next level of the hierarchy. This follows from the dynamic programming formulation of the flow control problem and the assumption that the production capacity set at each level is a polyhedron.

To show this, let us first consider a hierarchy level in which all the discrete events are random and the only decision variables are the activity rates $\mathbf{u}(t)$. Using a stochastic dynamic programming formulation of this problem, Kimemia and Gershwin [6] derive the Hamilton-Jacobi-Bellman equation and show that the optimal production rate at each time instant is a solution of the following linear program:

$$min_{\mathbf{u}} \quad \bigtriangledown_{\mathbf{x}} J(\mathbf{x}, \alpha).\mathbf{u} \qquad \text{s.t.} \qquad \mathbf{u} \in \Omega_\alpha \tag{3}$$

where $J(\mathbf{x}, \alpha)$ is the differential cost, α is the stochastic state , and Ω_α is the corresponding production capacity set. The surplus evolves according to $\dot{\mathbf{x}} = \mathbf{u} - \mathbf{d}$, where $\mathbf{d}$ is a constant target rate.

We briefly describe the behavior of the surplus trajectory resulting from this linear program. In general, an optimum solution of the linear program is at a corner point of the production capacity set. It follows that the $\mathbf{x}$ space can be partitioned into regions of constant production rate, i.e. regions for which one of the corner points is optimum. Therefore one can define a one-to-one correspondence between these $\mathbf{x}$-space regions and corner points of the capacity set Ω_α (for details see Gershwin, Akella, and Choong [9]). The $\mathbf{x}$-space boundaries may then be associated with faces of Ω_α which connect the corresponding corner points. The $\mathbf{x}$-space boundaries go through a *hedging point* and are linear if the differential cost function $J(\mathbf{x}, \alpha)$ is approximated by a quadratic function in $\mathbf{x}$ (for details see [9]). Figure 3 shows a capacity set and the corresponding $\mathbf{x}$-space partition for this case.

Note that this policy moves the surplus trajectory along a straight line until it reaches a boundary. At this point the production rate switches to another corner point (one adjacent to the previous one) and the surplus trajectory moves into a new region. This continues until the direction of movement of the trajectory in the new region is such that it returns the trajectory back to the boundary it just crossed. Such a boundary is called *attractive*. A production rate on the corresponding face of the capacity set must be chosen to keep the surplus trajectory on the attractive boundary and avoid chattering. This latter situation is of particular interest to us and we return to it in Section 3.2. Notice that for the case under consideration the surplus trajectory is piecewise linear with constant velocity on each piece.

Now let us consider a more general case with both controllable and stochastic discrete events at a level of the hierarchy. Let us define $J(\mathbf{x}, \alpha, \sigma)$ as the differential cost function at this level, where $\mathbf{x}$ and α are as defined before, and σ is the vector of controllable discrete states corresponding to this level. In the setup scheduling problem, σ is the setup state at the setup level. More precisely, σ specifies which part can be produced by each machine at the current time. Consider a time interval (t, t') during which the optimal control keeps the controllable state σ unchanged. The Hamilton-Jacobi-Bellman equation for the system over this interval will have the same form as in the case with no controllable events, and the solution trajectory will be piecewise linear in the $\mathbf{x}$-space. At the instant t' when optimal control changes σ to σ' a new *regime* begins in which the Hamilton-Jacobi-Bellman equation has the same form as before but with σ' instead of σ. Therefore, the trajectory will be piecewise linear in the new regime as well. We conclude that the surplus trajectory determined at each level of the hierarchy will be piecewise linear.

3.2 The Proposed Setup Policy

Given the production rate targets determined in the higher level and the corresponding $\mathbf{x}$-space target trajectory, we have to find a setup scheduling policy which tracks this trajectory with minimal deviations. The target trajectory is assumed to be piecewise linear with some pieces traversing full dimension regions of $\mathbf{x}$-space, while others coincide with portions of the lower dimensional attractive boundaries that partition the $\mathbf{x}$-space as described in Section 3.1. For the former pieces, the associated production rate is a corner point of the higher level capacity set. Operating at such points is feasible at the setup level since a corner point of the capacity set at the higher level corresponds to operating all the machines (including the inflexible ones) as dedicated machines. In other words, each machine will dedicate its entire capacity to production of a particular part type.

A problem may arise, however, while tracking the trajectory pieces which coincide with attractive boundaries. As discussed earlier, in order to move on a boundary, a production rate must be chosen on a *face* of the production capacity set. This implies that some machines must split their time among production of two or more part types. If these machines are perfectly flexible (i.e. have zero setup times) we can move on the trajectory by making very frequent setup changes so that the production time of the machines is almost continuously divided among the various part types. However, if some of these machines have non-zero setup times it will be impossible to stay on the target trajectory since each setup change moves the surplus away from the target trajectory, and changing setups too frequently can cause the surplus to fall behind this trajectory.

The setup scheduling policy described below provides a method for tracking the target trajectory while observing the capacity constraints of the setup level.

Case a : Tracking the production rate target from the higher level requires the inflexible machines to work as dedicated machines (this is the case where the target production rate is either a corner point of the higher level capacity set, or it is on a face that does not correspond to inflexible machines): Set up the inflexible machines accordingly, and produce at full capacity. By producing at full capacity it is temporarily possible to exceed the part production target rates determined at the higher level. This happens since these targets are compatible with a smaller capacity which reflects a positive setup rate as described in Section 2.1. Since at the setup level all the inflexible machines work as dedicated machines (while Case a holds), the setup rate is temporarily zero and this allows for a higher production rate. We will argue later that this is a desirable policy.

Case b: Tracking the production rate target from the higher level requires some inflexible machines to produce more than one part type : Make repeated setup changes for the inflexible machines involved, in such a way that the setup level trajectory follows the general direction of the target trajectory. We propose the use of setup scheduling *corridors* as a means of achieving this.

Figure 4 shows an example of a target trajectory (solid line) and the corresponding setup level trajectory (dashed line). In regions 4 and 3 all the machines are working as dedicated machines (Case a), and the setup level allocates all capacity to production and none to setup changes. Hence the surplus trajectory will be ahead of the target trajectory in this region. The boundary between regions 2 and 3 is an attractive boundary and requires an inflexible machine to split its time between production of the two part types as well as setup changes (Case b). Here the inflexible machine has to do repeated setup changes in order to follow (in an average sense) the target trajectory. The machine is producing part type 1 from point **a** to point **b**, is being set up for part type 2 from **b** to **c** , is producing part type 2 from **c** to **d** , is changing to setup 1 from **d** to **e**, and so on.

The magnitude of the zigzags around the target trajectory should be such that the average setup rate is the same as the target rate selected at the higher level. We expect, ceteris paribus, that the target setup rate will be smaller if the machine setup times are larger. This means larger magnitude zigzags for larger setup times. With large setup times the zigzags have to be large enough to allow progress in the desired direction. When the setup times are small we can make frequent setup changes to stay closer to the target trajectory.

The following points provide justification for the proposed policy:

i) When the target production rate requires all inflexible machines to work as dedicated machines (Case a), it is feasible at the setup level to move on the target trajectory. However, it is more desirable to operate at full capacity and exceed the target rate.

The reason for this is as follows: At the higher level the setup rate is approximated by its average value. Therefore, in generating production rate targets, the higher level is constrained to a reduced capacity set reflecting this fixed setup rate (shaded region in Figure 2-b). In Case a, where no setups are needed, the desired setup frequency is zero. If at the higher level the setup frequency had been set to zero, then that level would have prescribed production at full capacity at the setup level. Therefore, it is desirable to produce at full capacity while Case a holds. In fact, we have to move faster in Case a, where no setups are needed, in anticipation of the future slowdowns when inflexible machines have to devote a portion of their time to repeated setup changes (Case b). This is analogous to a hedging point type strategy implemented at the setup level.

ii) In Case b, where the inflexible machine has to work on more than one part type, the target trajectory determined at the higher level is the ideal neighborhood to change setups. The setup level must track this trajectory, but to do so perfectly would require an infinite rate of setup changes. However, when setup times are nonzero, we have to lower the frequency of setup changes to avoid excessive loss of capacity.

iii) As setup times approach zero the setup changes can be performed very frequently, and in the limit the trajectory becomes identical to the trajectory in the absence of set up requirements.

3.3 Setup Scheduling Corridors

Motivated by the ideas presented so far, we propose a method for scheduling setup changes which uses *corridors* around a target trajectory. A corridor is constructed around each portion of the target trajectory which requires setup changes for one or more inflexible machines. A corridor serves to determine the actual timing of the setup changes. The trajectory moves inside the corridor, and a setup change is initiated once the surplus trajectory reaches each face of the corridor. The surplus trajectory zigzags under this policy inside a corridor to follow the general direction of the target trajectory. See Figure 5.

The width of a corridor depends on the target setup rates determined at the higher level. A wider corridor generates less frequent setup changes and larger swings in the surplus $\mathbf{x}$. Larger inventories and backlogs are generated with wider corridor.

An interesting situation arises in the case that the target trajectory leads to a target point. This comes up, for example, when the higher level discrete events are random failures and repairs of machines and the current machine state is one that has excess production capacity. The target point in this case is a hedging point (discussed earlier), and is the ideal level for the surplus in the given machine state.

Additional issues arise in this case, such as: how should the setup level trajectory behave when it is near a target point? Or, if the target point cannot be achieved in any fixed setup of the inflexible machines, how can we design a corridor which would make the setup level trajectory stabilize near the target point? These are the questions we investigate in the remainder of this paper. In our discussion we assume that the piecewise linear target trajectory leads to a target (hedging) point.

Parallel (Prism) Corridors:

Let us consider a portion of the target trajectory which requires setup changes for an inflexible machine. Simple corridors can be constructed from hyperplanes parallel to this line. In n-dimensional space we may consider a prism formed by n hyperplanes around a target trajectory. Each face of the prism is associated with a certain machine setup. The surplus moves inside the prism, and once a face of the prism is reached the corresponding setup change is initiated (Figure 6).

The corridor must have at least a minimum width for the trajectory to move towards the hedging point. Otherwise, the setup changes will be performed too frequently and too much capacity (time) will be devoted to setup changes, causing the trajectory to move in the opposite of the desired direction. If the corridor is chosen properly, the trajectory will move toward the target point at a uniform pace. In the absence of additional constraints the trajectory will eventually pass the target point and undesired surplus will be accumulated. Therefore, it is necessary to limit the movement of the trajectory beyond the target point by modifying the production rates or the corridor around the target point (e.g. by limiting the corridor from above by additional planes).

Hypercone Corridors:

A corridor shape resulting in a desirable surplus trajectory is a hypercone (Figure 7). Like the prism corridors, the faces of hypercone corridors are hyperplanes. In n-dimensional space we form a hypercone by n hyperplanes around the target trajectory. Each face of the hypercone is associated with a particular setup of the machine which is initiated once that face is reached. A desirable property of a hypercone corridor is that, if chosen properly, the surplus trajectory converges to a stable *limit cycle* without the need to modify the production rates or the corridor around the target point as was the case with prism corridors. In this case the speed at which the surplus trajectory approaches the target point slows down until the trajectory is trapped into a limit cycle. In the next section we will characterize possible limit cycles that may arise when a machine goes through a periodic sequence of setup changes. Conditions for convergence of the trajectory to a limit cycle are discussed in Section 5.

A Clearing Hypercone Corridor:

Consider a special case of the hypercone corridor in which the faces of the corridor are the orthogonal hyperplanes of the coordinate system (in the negative region of the $\mathbf{x}$ space) and its apex is at the origin. We assume that the setup i, $i = 1, ..n$, is initiated when the trajectory reaches the face $x_{(i-2+n)mod(n)+1} = 0$. This means that the backlog of each part type is completely cleared before the set up is changed for production of the next part type in the cyclic sequence $1, 2, ...n$. We will call this corridor a *clearing corridor*.

The CAF (Clear-A-Fraction) setup policies suggested by Perkins and Kumar [21] have an important feature in common with a clearing corridor, namely that, they also assume the clearing property. Note, however, that the general corridor policy proposed here allows for *partial clearing* of the backlogs in each setup and, in this sense, is more general than the CAF policies. An optimal corridor would depend on the cost (penalty) function used for backlogs and, in general, is not of a clearing type.

In principle, the setup policy proposed above is applicable to a manufacturing system with many flexible and inflexible machines. We limit our exposition in the rest of this paper to a system with a single inflexible machine. The results obtained in the following sections will remain essentially the same if the system consists of many flexible machines working in parallel to a single inflexible machine. In this case, we only need to determine the production rate target on the inflexible machine before we apply the proposed approach. In the case of many inflexible machine, additional issues arise regarding the sequencing of setups among the inflexible machines, and more research is needed to derive specific results.

4 Characterization of Limit Cycles

Consider a target trajectory leading to a target point in n-dimensional $\mathbf{x}$ space . Let us assume that the production of a single inflexible machine must be switched among n different part types in order to track this trajectory. Once the target trajectory reaches the target point, the production rate target from the higher level will be set equal to the demand rate and the target surplus remains at this point. However, with nonzero setup times the setup level trajectory can never stay at the target point. Instead it will keep moving about the target point. We will show that, with the proper choice of setup durations, it is possible for the trajectory to return to the same point after each completion of a given sequence of setup changes. In this way the trajectory will keep going through a *cycle* in the surplus space which we refer to as a limit cycle. In this section we characterize possible limit cycles of the surplus trajectory.

Let $\mathbf{u}_i = [0, ...0, \hat{u}_i, 0, ...0]^T$ denote the production vector of the inflexible machine in setup i ($\hat{u}_i$ is the maximum production rate of part type i when the machine is setup for that part type). We denote the net demand rate on the inflexible machine by $\mathbf{d} = [d_1, d_2, ..., d_n]^T$. This demand is the total demand rate minus the production rates of other machines. The vector $\mathbf{v_i} = \mathbf{u_i} - \mathbf{d}$ is the surplus velocity in setup i. See Figure 8-a.

In order for the trajectory to stay on a cycle, the inflexible machine's production rate averaged over one cycle must be exactly equal to the demand rate $\mathbf{d}$. Figure 8-b shows such a cycle for a two-part-type example. In this figure $\mathbf{v}_1$ is the surplus velocity in setup 1, $\mathbf{v}_2$ is the surplus velocity in setup 2, and $-\mathbf{d}$ is the surplus velocity while the machine's setup is being changed. Let T denote the duration of a limit cycle, i.e. the time required for the trajectory to complete one cycle. To find the fraction β_i of the cycle time T that the inflexible machine must spend in each setup i in order to meet this condition, we write $\mathbf{d}$ as :

$$\mathbf{d} = \sum_{i=1}^{n} \beta_i \mathbf{u}_i + \beta_0 \mathbf{0} \tag{5}$$

where $\mathbf{0}$ is the zero vector (origin). Note that β_0 is the fraction of time that the inflexible machine is not doing operations on parts. We assume that the machine is never idle and therefore this fraction of time is spent on setup changes. Since $\mathbf{u}_i = [0, ...0, \hat{u}_i, 0, ...0]^T$, (5) can be easily solved for β_i's. We find :

$$\beta_i = \frac{d_i}{\hat{u}_i} \quad i = 1, 2, ...n \tag{6}$$

Let τ_i denote the time that takes to process one unit of part type i on the inflexible machine (note that $\tau_i = \frac{1}{\hat{u}_i}$). If all the setup times were zero, the capacity set of the inflexible machine would be characterized by the following inequality :

$$\sum_{i=1}^{n} \tau_i u_i = \sum_{i=1}^{n} \frac{u_i}{\hat{u}_i} \leq 1. \tag{7}$$

If the demand vector $\mathbf{d}$ is in the interior of this set we will have :

$$\sum_{i=1}^{n} \tau_i d_i = \sum_{i=1}^{n} \frac{d_i}{\hat{u}_i} < 1, \tag{8}$$

which, in view of (6), yields :

$$\sum_{i=1}^{n} \beta_i < 1,$$

and

$$\beta_0 = 1 - \sum_{i=1}^{n} \beta_i > 0 \qquad (9).$$

We conclude that when the demand vector $\mathbf{d}$ is in the interior of the capacity set corresponding to zero setup times, β_i's and β_0 exist and are uniquely determined from (6) and (9). This implies, as we will show, the existence of a limit cycle. Since for a stable control policy $\mathbf{d}$ must be feasible at the higher level, it follows that this condition will always hold. (Recall that the capacity set at the higher level is a reduced copy of the capacity set with zero setup times.)

Next, let us consider a fixed sequence of setups S, with each setup appearing at least once in the sequence. Let τ_S denote the total time spent for setup changes in this sequence. Then the duration of the limit cycle (T) must satisfy the equation

$$\tau_S = \beta_0 T,$$

or

$$T = \frac{\tau_S}{\beta_0}.$$

The total time spent in setup i is

$$t_i = \beta_i T = \frac{\beta_i}{\beta_0} \tau_S. \tag{10}$$

It is interesting to note that the duration of the limit cycle (T) and the time spent in each setup (t_i's) depend only on the total time τ_S spent for setup changes in a sequence S. They do not depend directly on the order of setups in the sequence or even the number of setups in the sequence. This follows from the fact that β_i's and β_0 do not depend on the sequence of setups S. In particular, if the time required to change the setup from setup i to setup j depends only on j, then τ_S will be independent of the order of setups in a given sequence S, and all the limit cycles corresponding to various setup orders will have the same duration. Of course the shape of the limit cycle will still depend on the order of setups.

In general a limit cycle may consist of a sequence of setups with some setups occurring more than once in the sequence. A special case of interest is a limit cycle consisting of a *round robin* sequence of setups, i.e. a sequence of setups in which the machine will be set up to produce each part type only once in the sequence. The limit cycle corresponding to a given round robin setup sequence will be completely determined from the knowledge of the t_i's since the sequence of setups and the duration of each setup are known.

The limit cycle will be shortest for a sequence of setups for which τ_S is minimum. This gives a motivation for using round robin sequences of setups since the total setup time will be smaller than that of any other sequence. However, it can be sometimes preferable to use a different sequence. In the rest of the paper we will limit our attention to round robin setup sequences.

The shape of a limit cycle for a given sequence of setups is independent of its location in $\mathbf{x}$ space. In other words limit cycles corresponding to the same sequence S with the same t_i's and β_i's can be realized at different locations in the space.

5 Choice of Corridors and Existence of Stable Limit Cycles

5.1 Linear Convex Corridors

In this section we investigate the dynamic behavior of the surplus ($\mathbf{x}$) for a setup scheduling policy that uses a hypercone corridor. For convenience let us assume that a single inflexible machine has to switch its production among m different part types in order to track a target trajectory. All the other machine in the system are assumed to be either flexible or operating as dedicated machines. We construct a linear convex corridor with m faces around the target trajectory. Each face of the corridor is associated with one of the setups and serves to determine the time to initiate a switch into that setup.

For a given corridor, the surplus trajectory ($\mathbf{x}$) is a piecewise linear function of time. The break points of this function correspond to the intersections of the surplus trajectory with the faces of the corridor and the subsequent setup changes. Therefore, in order to determine the dynamics of $\mathbf{x}$ it is sufficient to determine the behavior of the points of intersection of the trajectory with each face of the corridor. Let $\mathbf{x}(t_k)$ denote the k-th intersection of the trajectory ($k = 1, 2, ...$) with an arbitrarily chosen face of the corridor. We prove the following proposition.

Proposition: If the faces of a linear convex corridor are such that the sequence of setups generated is a periodic round robin sequence then $\mathbf{x}(k)$ satisfies a linear time-invariant first order state-space equation :

$$\mathbf{x}(t_{k+1}) = A\mathbf{x}(t_k) + B\mathbf{s} + C\mathbf{b} \tag{11}$$

where A, B, and C are constant matrices, $\mathbf{s}$ is the vector of setup times for the given sequence of setup changes, and $\mathbf{b}$ is a constant vector associated with the hyperplanes that form the corridor.

Proof: Let us suppose that the fixed sequence of setups is $1, 2, 3, ...m$, i.e. the machine produces part type 1, then part type 2, and so on. Also suppose that the corridor face which triggers the initiation of setup i is given by:

$$\mathbf{a}_i^T\mathbf{x} = b_i$$

where $\mathbf{a}_i$ is a constant vector of dimension n and b_i is a scaler. Let $\mathbf{x}(t_k)$ be the k-th intersection of the trajectory with corridor face $\mathbf{a}_1^T\mathbf{x} = b_1$. Starting from point $\mathbf{x}(t_k)$ (for some $k \geq 1$), the inflexible machine must go through m stages of setup change and production to reach $\mathbf{x}(t_{k+1})$. In the first stage the machine is setup to produce part type 1 and then proceeds to produce this part type until the surplus trajectory encounters hyperplane $\mathbf{a}_2^T\mathbf{x} = b_2$. Let us define:

$\mathbf{x}^{(1)} =$ $\mathbf{x}(t_k)$.

$\mathbf{x}^{(i)} =$ intersection of surplus trajectory with $\mathbf{a}_i^T\mathbf{x} = b_i$.

$\hat{\mathbf{x}}^{(i)} =$ $\mathbf{x}$ at the completion of setup change to setup i

$\mathbf{d} =$ net demand rate on the inflexible machine, i.e. total demand rate minus the production rate of all the other machines.

$\mathbf{v}_0 =$ velocity of $\mathbf{x}$, $(\frac{d\mathbf{x}}{dt})$, while the inflexible machine is being setup, $\mathbf{v}_0 = -\mathbf{d}$.

$\mathbf{v}_i =$ velocity of $\mathbf{x}$ when the machine is in setup $i(\mathbf{v}_i = \mathbf{v}_0 + \mathbf{u}_i$ where $\mathbf{u}_i$ is the production rate of the inflexible machine in setup $i)$.

$s_i =$ time required to change the setup of the inflexible machine from setup $i-1$ to setup i, $i = 2, .., m$.

$s_1 =$ time required to change the setup of the inflexible machine from setup m to setup 1.

We can now write :

$$\hat{\mathbf{x}}^{(1)} = \mathbf{x}^{(1)} + s_1 \mathbf{v}_0 \tag{12}$$

and

$$\mathbf{x}^{(2)} = \hat{\mathbf{x}}^{(1)} + t_1 \mathbf{v}_1 \tag{13}$$

where t_1 is the time until the trajectory reaches the corridor face $\mathbf{a}_2\mathbf{x} = b_2$ (starting from $\hat{\mathbf{x}}^{(1)}$). Since $\mathbf{x}^{(2)}$ is on this face, from (13) we have:

$$\mathbf{a}_2^T \mathbf{x}^{(2)} = \mathbf{a}_2^T(\hat{\mathbf{x}}^{(1)} + t_1 \mathbf{v}_1) = b_2 \tag{14}$$

which yields

$$t_1 = \frac{b_2 - \mathbf{a}_2^T \hat{\mathbf{x}}^{(1)}}{\mathbf{a}_2^T \mathbf{v}_1} \tag{15}$$

substituting in (13) we find

$$\mathbf{x}^{(2)} = (I - \frac{\mathbf{v}_1 \mathbf{a}_2^T}{\mathbf{a}_2^T \mathbf{v}_1})\hat{\mathbf{x}}^{(1)} + \frac{b_2}{\mathbf{a}_2^T \mathbf{v}_1}\mathbf{v}_1.$$

Now we substitute $\hat{\mathbf{x}}^{(1)}$ from (12) to get

$$\mathbf{x}^{(2)} = (I - \frac{\mathbf{v}_1 a_2^T}{\mathbf{a}_2^T \mathbf{v}_1})\mathbf{x}^{(1)} + s_1(I - \frac{\mathbf{v}_1 \mathbf{a}_2^T}{\mathbf{a}_2^T \mathbf{v}_1})\mathbf{v}_0 + \frac{b_2}{\mathbf{a}_2^T \mathbf{v}_1}\mathbf{v}_1. \tag{16}$$

We notice that $\mathbf{x}^{(2)}$ is a linear nonhomogeneous transformation of $\mathbf{x}^{(1)}$. Transformations for the stages 2 through m of the cycle have the same form. Letting

$$A_i = I - \frac{\mathbf{v}_i \mathbf{a}_{i+1}^T}{\mathbf{a}_{i+1}^T \mathbf{v}_i}$$

we find

$$\mathbf{x}^{(i+1)} = A_i \mathbf{x}^{(i)} + s_i A_i \mathbf{v}_0 + \frac{b_{i+1}}{\mathbf{a}_{i+1}^T \mathbf{v}_i}\mathbf{v}_i \qquad i = 1, 2, ...m \tag{17}$$

Writing this equation for $i = 1, 2, ...m$, we can find $\mathbf{x}^{(m+1)}$ (which is the intersection of the surplus trajectory with the initial corridor face after a full cycle of setups) in terms of $\mathbf{x}^{(1)} = \mathbf{x}(t_k)$, in a recursive fashion. Letting $\mathbf{s} = [s_1\ s_2\ ...s_m]^T$ and $\mathbf{b} = [b_1\ b_2\ ...b_m]^T$, we get

$$\mathbf{x}^{(m+1)} = \mathbf{x}(t_{k+1}) = A\mathbf{x}(t_k) + B\mathbf{s} + C\mathbf{b}$$

where

$$\begin{aligned}
A &= A_m A_{m-1} \ \cdots A_1 \\
B &= [(A_m A_{m-1} \cdots A_1)\mathbf{v}_0 \ \ (A_m A_{m-1} \cdots A_2)\mathbf{v}_0 \ \ \cdots \ \ A_m \mathbf{v}_0] \\
C &= \left[\frac{\mathbf{v}_m}{\mathbf{a}_1^T \mathbf{v}_m} \ \ \frac{(A_m A_{m-1} \cdots A_2)\mathbf{v}_1}{\mathbf{a}_2^T \mathbf{v}_1} \ \ \frac{(A_m A_{m-1} \cdots A_3)\mathbf{v}_2}{\mathbf{a}_3^T \mathbf{v}_2} \ \ \cdots \ \ \frac{A_m \mathbf{v}_{m-1}}{\mathbf{a}_m^T \mathbf{v}_{m-1}^T} \right].
\end{aligned}$$

Since by assumption the same sequence of setups are repeated in the next cycle, the matrices A, B, and C will remain the same i.e. they will be independent of k. This completes the proof of the proposition.

Note: The same line of proof as in above can be used to show that (11) holds for any periodic sequence of setups (not just round robin sequences).

Corollary: If the conditions of the proposition hold, the nonhomogeneous system (11) is stable if the associated homogeneous system

$$\mathbf{x}(t_{k+1}) = A\mathbf{x}(t_k)$$

is stable. That is, starting from an arbitrary $\mathbf{x}$ the surplus trajectory generated by a corridor policy will converge to a limit cycle if A is a stable matrix. Note that the stability is independent of $\mathbf{s}$, the setup times. For a similar result regarding independence of stability from setup times, see Perkins and Kumar [21].

5.2 Existence and Stability of Limit Cycles for Hypercone Corridors

Let us consider a hypercone corridor with m faces. We assume, for convenience, that the apex of the cone is at the origin. This condition can be achieved without loss of generality by translating the coordinate system. We may represent such a hypercone by

$$\mathbf{a}_i^T\mathbf{x} \leq 0 \quad i = 1, 2, ...m \tag{18}$$

According to the proposed setup policy, once the trajectory reaches the face $\mathbf{a}_i^T\mathbf{x} = 0$ from inside the hypercone, the setup change to production of part type i is initiated. We find conditions on vectors $\mathbf{a}_i$ so that for the proposed setup policy the sequence of setups will be a periodic round robin sequence.

Theorem 1: A hypercone corridor generates a periodic round robin sequence of setups 1,2, ... m, if the following conditions hold:

i) for all i and j, $(i = 1, 2, ...m \;\; j = 1, 2, ...m) \;\; i \neq j$,

$$\mathbf{a}_j^T\mathbf{v}_i > 0 \qquad \text{when } j = i(mod\ m) + 1$$

$$\mathbf{a}_j^T\mathbf{v}_i \leq 0 \qquad \text{when } j \neq i(mod\ m) + 1 \tag{19}$$

ii) for all i

$$\mathbf{a}_i^T\mathbf{v}_0 \leq 0, \tag{20}$$

where $\mathbf{v}_i$ and $\mathbf{v}_j$ are velocities of the surplus trajectory in setups i and j, respectively, and $\mathbf{v}_0$ is the velocity while the machine is being set up.

Proof: Let us suppose that the trajectory has just reached the face $\mathbf{a}_i^T\mathbf{x} = 0$ of the corridor. According to the corridor policy, we have to change the setup to setup i at this instant. Let $\mathbf{x}_0$ denote the surplus at the completion of this setup change. From condition (ii), $\mathbf{x}_0$ will be inside the corridor, or possibly on the face $\mathbf{a}_i^T\mathbf{x} = 0$. The trajectory moves from $\mathbf{x}_0$ with velocity $\mathbf{v}_i$. If the trajectory can reach a face $\mathbf{a}_j^T\mathbf{x} = 0$ of the corridor (while moving with velocity $\mathbf{v}_i$), then the time t_j to reach this face can be found from :

$$\mathbf{a}_j^T(\mathbf{x}_0 + t_j\mathbf{v}_i) = 0,$$

If t_j found from this equation is negative, it means that the trajectory will never reach this face. Therefore :

$$t_j = \begin{cases} -\frac{\mathbf{a}_j^T\mathbf{x}_0}{\mathbf{a}_j^T\mathbf{v}_i} & \text{if this quantity is greater than or equal zero} \\ \infty & \text{otherwise} \end{cases}$$

When $\mathbf{x}_0$ is strictly inside the corridor, it follows from (18) that the numerator of the above expression is strictly negative, and in view of condition (i) t_j exists (i.e. it is positive and finite) only for $j = i(mod\ m) + 1$. For all other values of j, $t_j = \infty$. Therefore, we conclude that the trajectory will always reach the face corresponding to the next setup in the sequence and no other face of the corridor. Consequently, the sequence of setups generated by the corridor will be a periodic round robin sequence in this case.

For the case in which $\mathbf{x}_0$ is on the face $\mathbf{a}_i^T\mathbf{x} = 0$ of the corridor, t_i is trivially zero. For $j = i$, from condition (ii) the denominator is either strictly negative or it is zero. In the former case, the trajectory will move away from the face $\mathbf{a}_i^T\mathbf{x} = 0$, and reaches the face corresponding to the setup $i(mod\ m) + 1$ in the sequence as before. In the latter case, the trajectory moves on the face $\mathbf{a}_i^T\mathbf{x} = 0$ until it reaches the next face corresponding to setup $i(mod\ m) + 1$ in the sequence. Therefore, the theorem remains valid for this case as well.

One can always find vectors $\mathbf{a}_i$ such that the conditions of Theorem 1 hold. To see this recall that the velocity of $\mathbf{x}$ in setup i is $\mathbf{v}_i = \mathbf{v}_0 + [0\ 0 ... \hat{u}_i\ 0 ... 0]^T$, where the i-th entry $\hat{u}_i$ is the production rate of part type i by the inflexible machine. Since $\mathbf{v}_0 \leq 0$ by assumption, all the entries of the vector $\mathbf{v}_i$ are negative except the i-th one which is positive. Now consider, as an example, the clearing corridor discussed in Section 3.3. The vectors $\mathbf{a}_i$ in this case are:

$$\begin{aligned} &\mathbf{a}_1 = [0\ 0\ 0 ... 1]^T \\ &\mathbf{a}_2 = [1\ 0\ 0 ... 0]^T \\ &\mathbf{a}_3 = [0\ 1\ 0 ... 0]^T \\ &\vdots \\ &\mathbf{a}_m = [0\ 0\ ... 1\ 0]^T. \end{aligned}$$

These vectors satisfy conditions (i) and (ii)of the theorem since for all $i = 1, 2, ..., m$, and $j = 1, 2, ..., m$:

$$\mathbf{a}_j^T\mathbf{v}_i = \begin{cases} v_{0i} + \hat{u}_i > 0 & j = i(mod\ m) + 1 \\ v_{0,(j+m-2)(mod\ m)+1} \leq 0 & j \neq i(mod\ m) + 1 \end{cases}$$

where v_{0i} is the i-th component of the vector $\mathbf{v}_0$. Condition (ii) is also satisfied since $\mathbf{v}_0 \leq 0$.

Finally, we prove the following theorem regarding the existence and stability of a limit cycle for hypercone corridors.

Theorem 2: For the proposed setup policy with a hypercone corridor satisfying the conditions of Theorem 1, the surplus trajectory converges, after sufficiently long time, to a unique stable limit cycle, provided that the target demand rate on the inflexible machine $\mathbf{d}$ (as defined in Section 5.1) satisfy

$$\sum_i \tau_i d_i < 1, \tag{21}$$

where τ_i is the processing time of part type i on the inflexible machine.[1]

Proof: First notice that (21) means that the target demand on the inflexible machine must be in the interior of the capacity set of the machine corresponding to zero setup times (see (7)). By Theorem 1 the sequence of setups will be periodic round robin. It follows, by Proposition 1 and its corollary, that it is sufficient to investigate the existence and stability of the homogeneous system equilibrium, i.e. when the setup times are zero. Recalling that $\mathbf{x}(t_k)$ is the intersection of the surplus trajectory with a face of the corridor, we show that the origin ($\mathbf{x} = \mathbf{0}$) is the unique stable equilibrium for

$$\mathbf{x}(t_{k+1}) = A\mathbf{x}(t_k). \tag{22}$$

This proves that in the presence of setup times, the intersection of the surplus trajectory with each hypercone face converges to a unique point, and therefore the theorem holds.

To prove that $\mathbf{x}(t_k)$ in (22) converges to the origin, we use a Liapunov function $L(\mathbf{x})$. Let τ be the vector of processing times τ_i of part types on the inflexible machine. Consider the production rate vector $\mathbf{u_i} = [0, 0, ..., \hat{u}_i, 0, , 0]^T$ of the inflexible machine in setup i (Figure 8-a). Since $\tau^T \mathbf{u}_i = 1$, it follows from (21) that :

$$\tau^T \mathbf{d} < \tau^T \mathbf{u}_i \qquad \text{for all } i.$$

Since the surplus velocity $\mathbf{v}_i$ in setup i is $\mathbf{u}_i - \mathbf{d}$, it follows from the inequality above that :

$$\tau^T \mathbf{v}_i > 0 \qquad \text{for all } i. \tag{23}$$

Now let us consider the function

$$L(\mathbf{x}) = -\tau^T \mathbf{x}.$$

We will show that when there is no setup time this function will always decrease during each round robin cycle of setups. Denoting by t_i the duration of setup i in a given round robin cycle of setups, we have (with zero setup times) :

[1] It has been brought to our attention by one of the reviewers that in Perkin's thesis (see reference [21]) there is a proof for the asymptotic periodicity of the round robin CAF setup policies. As was mentioned in Section 3.3, the corridor policy is not, in general, a clearing policy.

$$\mathbf{x}(t_{k+1}) = \mathbf{x}(t_k) + \sum_i t_i \mathbf{v}_i.$$

Therefore :

$$L\left(\mathbf{x}(t_{k+1})\right) = -\tau^T \mathbf{x}(t_{k+1}) =$$
$$= -\tau^T \mathbf{x}(t_k) - \tau^T \sum_i t_i \mathbf{v}_i = L\left(\mathbf{x}(k)\right) - \sum_i t_i (\tau^T \mathbf{v}_i)$$

Since $t_i \geq 0$ for all i and $t_i > 0$ for at least one i, (23) implies

$$L\left(\mathbf{x}(t_{k+1})\right) < L\left(\mathbf{x}(t_k)\right) \tag{24}$$

Since $L(0) = 0$, if we show that $L(\mathbf{x})$ is strictly positive for every point inside and on the faces of the corridor except $\mathbf{x} = 0$, then $\mathbf{x} = 0$ is the unique global minimum of this function over the domain of $\mathbf{x}$ (we know that the trajectory does not leave the corridor). This together with (24) imply that $\mathbf{x} = 0$ is the unique and stable equilibrium of (22).

To show that $L(\mathbf{x}) = -\tau^T \mathbf{x} > 0$ for all $\mathbf{x}$ inside and on the boundary of the corridor (except $\mathbf{x} = 0$), we have to show that the entire corridor (except for its apex) is strictly under the hyperplane $\tau^T \mathbf{x} = 0$, i.e. where $\tau^T \mathbf{x} < 0$ (see Figure 9). In order to prove this, we first show that the corridor must be strictly on one side of the hyperplane $\tau^T \mathbf{x} = 0$. Let us suppose, on the contrary, that either the hyperplane $\tau^T \mathbf{x} = 0$ goes through the interior of the corridor (case 1), or it coincides with a face of the corridor (case 2). We will show that either case results in a contradiction.

In case 1 we can choose an arbitrary point $\mathbf{x}_0$ on a face of the corridor which lies above the hyperplane, thus $\tau^T \mathbf{x}_0 > 0$. Starting at this point the setups of the machine change, as determined by the corridor, until the surplus trajectory intersects the hyperplane $\tau^T \mathbf{x} = 0$ at some point $\mathbf{x}_1$. The trajectory will always intersect the hyperplane since it has to reach all faces of the corridor, some of which are on the other side of the hyperplane. We can write

$$\tau^T \mathbf{x}_1 = \tau^T [\mathbf{x}_0 + \sum_i t'_i \mathbf{v}_i] = \tau^T \mathbf{x}_0 + \sum_i t'_i \tau^T \mathbf{v}_i$$

where t'_i is the time spent by the machine in setup i before reaching the hyperplane. Since $t'_i \geq 0$ for all i and at least one $t'_i > 0$, it follows (in view of (23)) that the rightmost side of the above equality is strictly positive, contradicting the fact that the leftmost side is zero.

We next consider case 2 which assumes that $\tau^T \mathbf{x} = 0$ is a face of the corridor. From (19) we have $\tau^T \mathbf{v}_i \leq 0$ for some i which again contradicts (23). We conclude that the corridor is located strictly on one side of the hyperplane.

It remains to show that the corridor lies under the hyperplane. By assumption, the corridor must include its associated attractive boundary. We show that this boundary is always under the hyperplane, and this completes the proof. The target velocity on the attractive boundary can be expressed as : $\mathbf{v}_b = \sum_i \gamma_i \mathbf{v}_i$, where the γ_i's are positive constants. It then follows from (23) that $\tau \mathbf{v}_b > 0$, which implies that the attractive boundary is under the hyperplane. □

The main advantages of reaching a limit cycle are that the demand is tracked automatically when the system reaches the limit cycle (since the surplus returns to the same point after each cycle) and, secondly, the setup changes become periodic with a fixed production cycle. The latter property implies more regularity and perhaps ease of implementation. Note, however, that it is not necessary for the surplus to always reach its limit cycle since the occurrence of a discrete event at the higher levels of the hierarchy may change the targets for the setup level at any instant.

6 Simulation Experience with Hypercone Corridors

We have simulated a large number of hypercone corridor setup scheduling policies. The behavior of many policies with corridors satisfying the conditions of Theorem 2 was investigated. In all cases, a stable limit cycle was observed. We also simulated policies that did not satisfy condition (ii) of Theorem 1. Interestingly, the surplus trajectory converged to a limit cycle in all cases (Figure 10-a). When we relaxed condition (i) of Theorem 1, the trajectory did not converge to a limit cycle but rather took an apparently chaotic path (Figure 10-b). Even though limit cycles generally were found for these cases they were locally unstable. In some cases we found stable complex cycles in which some setups were present more than once in one cycle. These are typical behaviors of chaotic dynamic systems.

7 Conclusion

This paper proposes a framework for flow control of parts in a manufacturing system with machines that require setups. It has been shown that corridors can be constructed in the space of part type production surplus to trigger setup changes. The resulting setup schedule is consistent with production rate targets determined at a higher level of the hierarchy. The conditions which must be satisfied by the corridors for a stable setup schedule have been investigated. When these conditions are met, part type production surplus levels converge to a unique limit cycle. When certain conditions are violated simulation experience has shown that unstable complex cycles may exist and that chaotic behavior of the production surplus levels is likely.

Although the findings of this paper as well as research reported in Perkins and Kumar [21], Kumar and Siedman [22], and Chase and Ramadge [23] have improved our understanding of dynamic setup scheduling, future research focusing on the construction of optimal corridors as well as further investigation of chaotic behavior determinants can be very fruitful.

Acknowledgment : The authors would like to thank the anonymous reviewers for their constructive comments and suggestions.

References

[1] S. E. Elmaghraby,"The Economic Lot Scheduling Problem(ELSP) : Review and Extensions," *Management Science*, Vol. 24, No. 6, February 1978, pp. 587-598.

[2] S. K. Goyal, "Determination of Economic Production Quantities for a Two-Product Single Machine System," *International Journal of Production Research*, Vol. 22, No. 1, 1984, pp. 121-126.

[3] G. Dobson,"The Economic Lot Scheduling Problem : Achieving Feasibility Using Time-Varying Lot Sizes," *Operations Research*, Vol. 35, No 5, 1987, pp. 764-771.

[4] P. C. Jones, and R. R. Inman, " When Is The Economic Lot Scheduling Problem Easy ?," *IIE Transactions*, Vol. 21, No. 1, March 1989, pp. 11-20.

[5] J. J. Carreno," Economic Lot Scheduling For Multiple Products on Parallel Identical Processors," *Management Science*, Vol 36, No. 3, March 1990, pp. 348-358.

[6] J.G. Kimemia and S.B. Gershwin, "An Algorithm for the Computer Control of Production in Flexible Manufacturing Systems, " *IIE Transactions*, Vol. 15, p.353-362, December, 1983.

[7] R. Rishel," Dynamic programming and minimum principle for systems with jump Markov disturbances,"*SIAM Journal of Control*, vol. 13, pp.338-371, Feb. 1975.

[8] G. J. Olsder and R. Suri, "Time-Optimal Control of Parts Routing in a Manufacturing System with Failure-Prone Machines,"*Proceedings of the 19-th IEEE Conference on Decision and Control*, Dec. 1980.

[9] S.B. Gershwin, R. Akella, and Y.F. Choong, "Short-Term Production Scheduling of an Automated Manufacturing Facility," *IBM Journal of Research and Development*, Vol.29, pp.392-400, July 1985.

[10] R. Akella and P.R. Kumar, "Optical Control of Production Rate in a Failure Prone Manufacturing System," *IEEE Transactions on Automatic Control*, Vol. AC-31, No.2, pp.116-126, February 1986.

[11] T. Bielecki and P.R. Kumar,"Optimality of zero-inventory policies for unreliable manufacturing systems," *Operations Research*, Vol. 36, No.4, July-August 1988, PP. 532-541.

[12] A. Sharifnia, "Optimal Production Control of a Manufacturing System with Machine Failures," *IEEE Transactions on Automatic Control*, Vol. 33, No.7, pp.620-625, July 1988.

[13] P.H. Algoet, "Flow Balance for the Steady-State Distribution of a Flexible Manufacturing System," *IEEE Transactions on Automatic Control*,Vol. 34, No. 8, pp.917-921, August 1989.

[14] R. Malhame, and E. Boukas, "Transient and Steady-States of the Statistical Flow Balance Equations in Manufacturing Systems," *Proceedings of the 3rd ORSA/TIMS Conference on FMS*, Cambridge, MA, August 1989, K.E. Stecke and R. Suri eds. Elsevier Science Publishers B.V., Amsterdam, pp 339-345.

[15] A. Haurie, and Ch Van Delft, "Turnpike Properties for a Class of Piecewise Deterministic Systems Arising in Manufacturing Flow Control," *Proceedings of the 3rd ORSA/TIMS Conference on FMS*, Cambridge, MA, August 1989, K.E. Stecke and R. Suri eds. Elsevier Science Publishers B.V., Amsterdam, pp 333-338.

[16] M. Caramanis and A. Sharifnia, "Design of Near Optimal low Controllers for Flexible Manufacturing Systems," *Proceedings of the 3rd ORSA/TIMS Special Interest Conference on FMS*, Cambridge, MA, August 1989, K.E. Stecke and R. Suri eds. Elsevier Science Publishers B.V., Amsterdam.

[17] M. Caramanis and G. Liberopoulos, "Perturbation Analysis for the Design of Flexible Manufacturing Systems Flow Controllers," Laboratory for Manufacturing Systems and Productivity, Manufacturing Engineering Department, Boston University Technical Report BU-LMSP-88-007, September 1988.

[18] S.B. Gershwin, "A Hierarchical Framework for Discrete Event Scheduling in Manufacturing Systems," *Discrete Event Systems: Models and Applications*, IIASA Conference, Sopron, Hungary, August 1987, Varaya and Kurzhanski eds. No.103 in the series Lecture Notes in Control and Information Sciences, Springer-Verlag.

[19] S.B. Gershwin, "A Hierarchical Framework for Manufacturing Systems Scheduling: A Two Machine Example," Proceedings of the 26th Conference on Decision and Control, Los Angeles, December 1987.

[20] S.B. Gershwin, "Hierarchical Flow Control: A Framework for Scheduling and Planning Discrete Events in Manufacturing Systems, "*Proceedings of IEEE*, Vol. 77, No.1, January 1989, pp. 195-209.

[21] J. Perkins and P.R. Kumar, "Stable, Distributed, Real-Time Scheduling of Flexible Manufacturing/Assembly/Disassembly Systems," *IEEE Transactions on Automatic Control*, Vol. 34, No.2,, pp.139-148, February 1989.

[22] P.R. Kumar and T.I. Seidman "Dynamic Instabilities and Stabilization Methods in Distributed Real-time Scheduling of Manufacturing Systems," Coordinated Science Laboratory, University of Illinois, Urbana-Champaign.

[23] C. Chase and P.J. Ramadge, " On Real Time Scheduling Policies For Flexible Manufacturing Systems," preprint, Department of Electrical Engineering, Princeton University, 1989.

[24] S. Lou, S. Sethi, and G. Sorger,"*Stability of Real-Time Lot Scheduling Policies for an Unreliable Machine,"* preprint, Faculty of Management, University of Toronto, June 1989.