

## PERIODIC ORBITS IN A CLASS OF RE-ENTRANT MANUFACTURING SYSTEMS

IVONNE DIAZ-RIVERA, DIETER ARMBRUSTER, AND TOM TAYLOR

Queue changes associated with each step of a manufacturing system are modeled by constant vector fields (fluid model of a queueing network). Observing these vector fields at fixed events reduces them to a set of piecewise linear maps. It is proved that these maps show only periodic or eventually periodic orbits. An algorithm to determine the period of the orbits is presented. The dependence of the period on the processing rates is shown for a 3(4)-step, 2-machine problem.

### 1. Introduction.

**1.1. Re-entrant manufacturing systems.** A *re-entrant manufacturing system* (Kumar 1993) is a queueing network modeling a manufacturing process in which parts at different stages of completion await processing by the same machine. Different choices of scheduling policies at such a machine lead to different production efficiencies.

In manufacturing processes, such as the production of semiconductor products, there is also a requirement for release policies. These policies determine how much raw material needs to be added into the process and when it should be done. An example of a release policy is one in which a new job is added into the system at the time a job is completely finished. Such a policy will convert the system from an open loop system to a closed system.

The objective of studying various scheduling policies is to determine which one provides the best performance of the manufacturing process (see, e.g., Dai and Weiss 1996, Lu and Kumar 1991, Kumar and Meyn 1995, and Wolff 1989 for *stable* policies). The literature describes two primary goals: to reduce the mean cycle time, i.e., how long the part takes in the system; and to reduce the variance of the cycle time (Kumar 1993). The mean cycle time is directly related to the throughput of the system. *Throughput* is identified as the rate at which jobs get done (units of throughput used in this research are jobs/min).

**1.2. Motivation.** This work is motivated by the research done by Beaumariage and Kempf (1995) and Chase et al. (1993). Beaumariage and Kempf studied a four-step, four-machine system motivated by an Intel wafer fab. The focus of their investigation was a sensitivity analysis to determine if a re-entrant manufacturing system can be proved to be chaotic in the sense of a chaotic dynamical system (Alligood et al. 1996). The parameters that were varied to investigate the sensitivity of the system included: batching size, scheduling policies used, the number of jobs in the system, and the amount of raw material released into the system at various shifts. Two metrics were used to measure performance and characterize the dynamics of the system. One metric in Beaumariage and Kempf (1995) was the throughput time distributions over all jobs, and the second metric

Received June 16, 1998; revised September 1, 1999, April 17, 2000, and June 25, 2000.  
*MSC 2000 subject classification.* Primary: 37N40; secondary: 68M20.  
*OR/MS subject classification.* Primary: Queues/networks.  
*Key words.* Re-entrant manufacturing, periodic orbit, chaos.

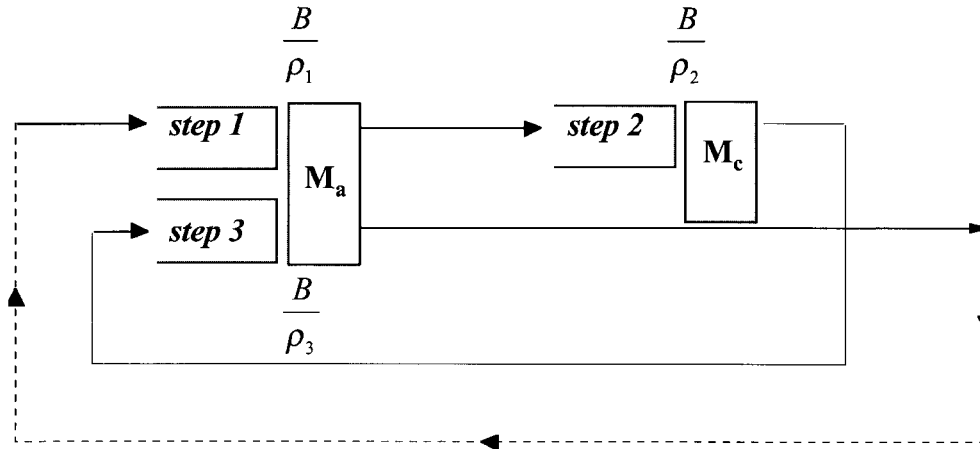


FIGURE 1. Three-step, two-machine system.

was the interdeparture times, i.e., the difference between the times at which jobs leave the last step. The second metric was used to identify the resulting patterns as the parameters were changed. The conclusion of their experiments was “that small changes in policies or initial state of the queues produce large effects on throughput time distributions and output interdeparture patterns, therefore chaotic behavior is strongly indicated” (Beaumariage and Kempf 1995).

Related work in which a fluid model was used to study a manufacturing model was presented in a paper by Chase et al. (1993). They studied two discretely controlled continuous variable systems based on sampled deterministic flow models. The *switched server system* was shown to be periodic in contrast to the *switched arrival system*, which was shown to be chaotic.

In their analysis of the switched arrival system, a discrete map from the unit interval to itself is determined using the Poincaré map method. This map is then studied and shown to have three unstable fixed points. The map exhibits characteristics of a chaotic system, such as sensitive dependence on initial conditions. It is also topologically transitive and its periodic orbits are dense in state space.

Walsh et al. (1996) studied the dynamics of maps and flows that characterize a special type of closed queueing network used in computer science applications. Their system consists of  $n=4$  servers, one of which is a central server,  $P$ , with a queue capacity of  $n-1=3$ ; and 3 other servers. They use FIFO and the last-in, first-out (LIFO) scheduling policies and study the dynamics of the maps generated via the Poincaré map method. Their results include the proof of the existence of a global attractor which consists of the set of periodic orbits which involve no waiting in the queue for the FIFO protocol. In the case of the LIFO policy, they showed that no attractors were found although the return map was shown to be area preserving; therefore, theorems from ergodic theory are used in this case to compute average behavior of the queue dynamics. Their work shows that dynamical systems methods are indeed useful in the study of a special type of a closed queueing network and therefore reinforces the motivation of this work.

The goal of this paper is to show that methods from dynamical systems theory can be used to determine whether simple models like the one used by Beaumariage et al. are indeed chaotic or not.

**1.3. Setup.** This paper uses the three-step, two-machine system shown in Figure 1 as the main example, although we will show how our methodology extends to more complicated systems. We will use a fluid model of a queueing network. The system is

re-entrant since the machine  $M_a$  serves Step 1 and Step 3, while machine  $M_c$  only serves Step 2. For the systems studied in this investigation the following assumptions are used:

- *Scheduling policy.* Consider a priority-like policy in which  $M_a$  prefers jobs that are coming in for step-3 service over step-1 service. This policy is called the *pull policy* by Beaumariage and Kempf (1995), last buffer first serve (LBFS) by Dai et al. (1997) and Kumar (1993) and shortest expected remaining processing time (SERPT) (Kleinrock 1976) in the other queueing theory literature. In general, a *pull policy* is such that later process steps are preferred over earlier process steps.

- *Release policy.* In all the cases that are studied in this research the release policy has been fixed to a closed queueing network.

- *Batch size.* For most of this analysis the batching size is set to two although some results are presented for arbitrary batch size  $B$  (and large enough load; see below).

Under these assumptions, the modeling is based on the use of ordinary differential equations that describe the queue size  $x_i$  of jobs that are waiting for step- $i$  service. For simplicity,  $x_i$  will be referred to as queue  $i$ . Therefore the state of a three-step, two-machine system is characterized by  $(x_1(t), x_2(t), x_3(t))$ , which represents the number of jobs waiting in each of the queues at time  $t \geq 0$ . The values  $(x_1(0), x_2(0), x_3(0))$  represent the *initial conditions* of the system. Under the assumptions presented so far, the set of all initial conditions lies in the simplex determined by

$$\left\{ \sum_{i=1}^3 x_i = L, \quad \forall i \ x_i \geq 0 \right\}$$

where  $L$  is the total number of parts in the system or just the total load.

After the model has been constructed and scheduling and release policies have been implemented, a discrete system is obtained by sampling the flow at a sequence of interevent times (Alligood et al. 1996). This map represents how the system evolves as a function of the size of the queues. One of the main results in this paper is that *maps generated by event-driven sampling are explicitly obtained to model a scheduling policy.* These maps enable the study of the trajectories for each set of initial conditions and in particular the resulting steady state behavior. We are interested in determining if the behavior is eventually periodic or chaotic and, if applicable, to predict its eventual period. Some of the tools used here to characterize the dynamics are:

- transition diagrams,
- integer programming models, and
- results from the iterated map via computer evaluations.

The second main result of this work is that for all initial conditions the dynamics is eventually periodic and *the length of the periodic orbits can be explicitly calculated.* As a consequence, for the models of re-entrant manufacturing systems studied here *no chaos exists but orbits can exhibit extremely long periods.* In fact, processing rates can be chosen to produce periodic orbits of arbitrarily long period.

Subsequent papers will show how dynamical systems methods allow us to determine the structure of the phase space, the existence of neutrally stable periodic orbits and the characterization of transients. In addition we will analyze bifurcations as processing rates or batch sizes are varied. It turns out that resonances between processing rates play a key role (Diaz-Rivera et al. 2000). Once phase space and parameter dependence is understood more thoroughly, control issues can be addressed.

The rest of this paper is organized as follows. Section 2 presents the modeling. It is shown how the policies are implemented and how event-driven sampling reduces the system to a simple map. Section 3 presents the application of the dynamical systems methodology to the models developed in §2. We eventually find periodic orbits for all possible initial conditions. The periods of the orbits can be calculated from a priori considerations;

their throughput is discussed. In §4, results from the application of this technique to a three-step, two-machine system are used as an example. We conclude with a discussion about generalizations of the methods and some ideas for future studies.

## 2. Modeling of re-entrant manufacturing systems.

**2.1. The flow model.** We use a three-step, two-machine system (see Figure 1) to illustrate the modeling approach. The size of the queue for each step in the manufacturing system is described as a continuous flow using a set of three differential equations. Let  $x_i(t)$  be the queue size at time  $t$  for step  $i$  service. Let  $\rho_i$  be the processing time required for step- $i$  service and let  $B$  be the size of the batch required by machine  $M_a$ . A necessary condition for machine  $M_a$  to start serving Queue 1 or Queue 3 is that there are at least  $B$  jobs in this queue. A machine will continue serving its queue until the next observation point is reached and a decision has been made to reallocate the machine to do a different step. In that way, a machine may continue to work on a queue even though the queue size drops below  $B$ . However, this decision process will have to make sure that the system will never produce negative queue sizes. Since the number of jobs is a conserved quantity we get:

$$(1) \quad \begin{aligned} \frac{dx_1}{dt} &= -u_a \frac{B}{\rho_1} + \tilde{u}_a \frac{B}{\rho_3}, \\ \frac{dx_2}{dt} &= -u_c \frac{B}{\rho_2} + u_a \frac{B}{\rho_1}, \\ \frac{dx_3}{dt} &= -\tilde{u}_a \frac{B}{\rho_3} + u_c \frac{B}{\rho_2}, \end{aligned}$$

with

$$(2) \quad u_a(t) = \begin{cases} 0 & \text{if } M_a \text{ is not serving } x_1, \\ 1 & \text{if } M_a \text{ is serving } x_1, \end{cases}$$

and

$$(3) \quad \tilde{u}_a(t) = \begin{cases} 0 & \text{if } M_a \text{ is not serving } x_3, \\ 1 & \text{if } M_a \text{ is serving } x_3. \end{cases}$$

Therefore, at the observation time, which we arbitrarily set to  $t=0$  when the allocation of machines (until the next observation time  $\tau$ ) can be rearranged, we may have four cases:

$$(4) \quad \begin{aligned} &\text{if } x_1(0) < B \text{ and } x_3(0) \geq B \Rightarrow u_a(t) = 0 \text{ and } \tilde{u}_a(t) = 1 \text{ for } 0 \leq t \leq \tau, \\ &\text{if } x_1(0) < B \text{ and } x_3(0) < B \Rightarrow u_a(t) = 0 \text{ and } \tilde{u}_a(t) = 0 \text{ for } 0 \leq t \leq \tau, \\ &\text{if } x_1(0) \geq B \text{ and } x_3(0) < B \Rightarrow u_a(t) = 1 \text{ and } \tilde{u}_a(t) = 0 \text{ for } 0 \leq t \leq \tau, \\ &\text{if } x_1(0) \geq B \text{ and } x_3(0) \geq B \Rightarrow \text{need to choose a policy.} \end{aligned}$$

In addition,  $u_c(t)$  is zero if  $M_c$  is idle, i.e.,  $x_2(0) < B$  and one if  $M_c$  serves Queue 2, i.e.,  $x_2(0) \geq B$ . This implies that if  $x_1(0) \geq B$  and  $x_3(0) < B$  machine  $M_a$  will work on Step 1 and hence  $x_1$  decreases at a rate  $B/\rho_1$  until the system is reallocated. Similarly, if  $x_1(0) < B$  and  $x_3(0) \geq B$  then the queue  $x_1$  increases at a rate  $B/\rho_3$  since machine  $M_a$  is working on queue 3 which feeds into step 1. Notice that if the queues for Steps 1 and 3 both have enough work  $\geq B$  at  $t=0$  then a scheduling policy needs to be implemented.

TABLE 1. Classification of states for the three-step, two-machine system

CLASS ID	REPRESENTATION	CONDITION
1	$(x_1^*, \bar{x}_2, \bar{x}_3)$	$x_1 \geq B, x_2 < B, x_3 < B$
2	$(x_1^*, \bar{x}_2, x_3^*)$ or $(\bar{x}_1, \bar{x}_2, x_3^*)$	$x_1 \geq B, x_2 < B, x_3 \geq B$ or $x_1 < B, x_2 < B, x_3 \geq B$
3	$(\bar{x}_1, x_2^*, \bar{x}_3)$	$x_1 < B, x_2 \geq B, x_3 < B$
4	$(x_1^*, x_2^*, \bar{x}_3)$	$x_1 \geq B, x_2 \geq B, x_3 < B$
5	$(x_1^*, x_2^*, x_3^*)$ or $(\bar{x}_1, x_2^*, x_3^*)$	$x_1 \geq B, x_2 \geq B, x_3 \geq B$ or $x_1 < B, x_2 \geq B, x_3 \geq B$

Consider a *pull policy*: If  $x_1(0) \geq B$  and  $x_3(0) \geq B$  then  $u_a = 0$  and  $\tilde{u}_a = 1$  (both implying that  $M_a$  is serving Queue 3). Then the phase space for the three differential equations (1) can be divided into five regions. Each region will be referred to as a class.

For example, Class 1, identified as  $(x_1^*, \bar{x}_2, \bar{x}_3)$ , represents the condition (at the observation point) in which Queue 1 has at least  $B$  jobs waiting for step-1 service and Queue 2 ( $x_2$ ) and Queue 3 ( $x_3$ ) both have strictly less than  $B$  jobs waiting for service. We denote any variable that is bigger than  $B$  by an asterisk and any variable that is less than  $B$  by an overbar. In summary,  $(x_1(0), x_2(0), x_3(0)) \in \text{Class 1}$  if

$$(5) \quad x_1(0) \geq B \quad \text{and} \quad x_2(0) < B \quad \text{and} \quad x_3(0) < B.$$

All other classes are represented similarly. Using this type of representation the state space can be divided into five regions depending on  $(x_1, x_2, x_3)$  at the observation time. From now on the identification numbers (1–5) will be used to represent each of these classes (see Table 1). In addition, there is the trivial Class 0 in which  $x_i < B$  for all  $i$ . We do not consider this class further since it is clearly not interesting.

For each class of the system the flow (Equation (1)) is described by a different set of fixed differential equations with constant flow. We can therefore say that the system is piecewise constant. Table 2 contains the system of ordinary differential equations that describes the changes in the size of the queues depending on which of the five classes the system is in at time  $t = 0$ .

**2.2. The discrete model.** Observing the system at certain events will discretize the system's dynamics. The main task is to determine a time at which to record the state of the system. At this time, which we will call the observation time or the snapshot time (Chase et al. 1993, call it the clearing time or the interevent time), it will be decided what each of the machines should do next. There are different criteria that can be used to determine these times. Different observation times will initiate different manufacturing processes and hence, different maps are created. This event-driven sampling is similar to observing a recurrent dynamical system using Poincaré cross-sections leading to Poincaré maps (Alligood et al. 1996).

Let *Pull Policy 1* be such that the observation time is determined by the following event: Either a queue has become zero or a queue has decreased by  $B$ , the batching size.

Let *Pull Policy 2* be such that the observation time is determined by the event: Either a queue has become zero or a machine has finished a batch of  $B$  jobs.

Notice that Pull Policy 1 focuses on the change in the queues, while Pull Policy 2 focuses on the production output. Either policy, together with the scheduling rules for pull policies, makes Equation (1) well defined and allows us to determine unique maps through event-driven sampling.

TABLE 2. Model for the three-step, two-machine system

CLASS ID	REPRESENTATION	EQUATIONS
1	$(x_1^*, \bar{x}_2, \bar{x}_3)$	$\dot{x}_1 = \frac{-B}{\rho_1}$ $\dot{x}_2 = \frac{B}{\rho_1}$ $\dot{x}_3 = 0$
2	$(x_1^*, \bar{x}_2, x_3^*)$ or $(\bar{x}_1, \bar{x}_2, x_3^*)$	$\dot{x}_1 = \frac{B}{\rho_3}$ $\dot{x}_2 = 0$ $\dot{x}_3 = \frac{-B}{\rho_3}$ $\dot{x}_1 = 0$
3	$(\bar{x}_1, x_2^*, \bar{x}_3)$	$\dot{x}_2 = \frac{-B}{\rho_2}$ $\dot{x}_3 = \frac{B}{\rho_2}$
4	$(x_1^*, x_2^*, \bar{x}_3)$	$\dot{x}_1 = \frac{-B}{\rho_1}$ $\dot{x}_2 = \frac{B}{\rho_1} - \frac{B}{\rho_2}$ $\dot{x}_3 = \frac{B}{\rho_2}$
5	$(x_1^*, x_2^*, x_3^*)$ or $(\bar{x}_1, x_2^*, x_3^*)$	$\dot{x}_1 = \frac{B}{\rho_3}$ $\dot{x}_2 = \frac{-B}{\rho_2}$ $\dot{x}_3 = \frac{B}{\rho_2} - \frac{B}{\rho_3}$

**2.2.1. Pull Policy 1.** Consider the general case in which all the processing times are different. Let  $B=2$  throughout the next example. For each of the five classes in Table 2 the observation time from one class to another needs to be determined. Let  $\tau_i$  be this time while the system is in class  $i$ . Consider Class 1, in which  $x_1$  decreases by 2 at  $t = \rho_1$  and  $x_2$  increases, therefore  $\tau_1 = \rho_1$ . Similarly it can be determined that in class 2,  $\tau_2 = \rho_3$  and, in class 3,  $\tau_3 = \rho_2$ . Therefore the maps are very easily determined by solving the ode's and then substituting  $t = \tau_i$  for  $i = 1, 2, 3$ , and letting  $\tilde{x}_i$  be the new value of the map. For Class 1 then,

$$(6) \quad \begin{aligned} \tilde{x}_1 &= x_1 - 2, \\ \tilde{x}_2 &= x_2 + 2, \end{aligned}$$

for Class 2,

$$(7) \quad \begin{aligned} \tilde{x}_1 &= x_1 + 2, \\ \tilde{x}_3 &= x_3 - 2, \end{aligned}$$

and for Class 3,

$$(8) \quad \begin{aligned} \tilde{x}_2 &= x_2 - 2, \\ \tilde{x}_3 &= x_3 + 2. \end{aligned}$$

Consider Class 4. If  $\rho_1 \leq \rho_2$  then  $\tau_4 = \rho_1$  because  $x_1$  is the only queue that has the opportunity to decrease and it decreases by 2 at exactly  $t = \rho_1$ . The map for this case is:

$$(9) \quad \begin{aligned} \tilde{x}_1 &= x_1 - 2, \\ \tilde{x}_2 &= x_2 + \frac{2\rho_2 - 2\rho_1}{\rho_2}, \\ \tilde{x}_3 &= x_3 + \frac{2\rho_1}{\rho_2}. \end{aligned}$$

If, however,  $\rho_1 > \rho_2$ , then

$$(10) \quad \tau_4 = \min\left(\rho_1, \frac{\rho_1 \rho_2}{\rho_1 - \rho_2}\right),$$

which implies that

$$(11) \quad \tau_4 = \rho_1 \quad \text{if } \rho_2 < \rho_1 \leq 2\rho_2,$$

and

$$(12) \quad \tau_4 = \frac{\rho_1 \rho_2}{\rho_1 - \rho_2} \quad \text{if } \rho_1 \geq 2\rho_2.$$

Equation (9) shows the map that results from the first case, and for the second case (12) the map looks like:

$$(13) \quad \begin{aligned} \tilde{x}_1 &= x_1 - \frac{2\rho_2}{\rho_1 - \rho_2}, \\ \tilde{x}_2 &= x_2 - 2, \\ \tilde{x}_3 &= x_3 + \frac{2\rho_1}{\rho_1 - \rho_2}. \end{aligned}$$

The same type of analysis can be performed to conclude that there are also two possibilities in Class 5: If  $\rho_2 \leq \rho_3$  then we find a map of the form

$$(14) \quad \begin{aligned} \tilde{x}_1 &= x_1 + \frac{2\rho_2}{\rho_3}, \\ \tilde{x}_2 &= x_2 - 2, \\ \tilde{x}_3 &= x_3 + \frac{2\rho_3 - 2\rho_2}{\rho_3}. \end{aligned}$$

If  $\rho_2 > \rho_3$ , then

$$(15) \quad \tau_5 = \min\left(\rho_2, \frac{\rho_2 \rho_3}{\rho_2 - \rho_3}\right),$$

which implies that

$$(16) \quad \tau_5 = \rho_2 \quad \text{if } \rho_3 < \rho_2 \leq 2\rho_3,$$

and

$$(17) \quad \tau_5 = \frac{\rho_2 \rho_3}{\rho_2 - \rho_3} \quad \text{if } \rho_2 \geq 2\rho_3.$$

Equation (14) shows the map that results from the first case, and for the second case, when  $\rho_2 \geq 2\rho_3$ , the equations are:

$$(18) \quad \begin{aligned} \tilde{x}_1 &= x_1 + \frac{2\rho_2}{\rho_2 - \rho_3}, \\ \tilde{x}_2 &= x_2 - \frac{2\rho_3}{\rho_2 - \rho_3}, \\ \tilde{x}_3 &= x_3 - 2. \end{aligned}$$

**2.2.2. Pull Policy 2.** In the case of Pull Policy 2 the determination of  $\tau_i$  is simpler. First, Classes 1, 2, and 3 are exactly the same as in Pull Policy 1 (as shown in systems 6–8). This is because machine  $M_a$  will have finished a full batch of jobs ( $B=2$  jobs) and therefore will become idle at the same time as a queue decreases by exactly 2, the number of jobs in the batch.

In Class 4, the value of  $\tau_4 = \min(\rho_1, \rho_2)$  and in Class 5,  $\tau_5 = \min(\rho_2, \rho_3)$ . So there are again two possibilities for Class 4 and two possibilities for Class 5. For  $\rho_1 < \rho_2$  in Class 4 the same equation is obtained as for Pull Policy 1. For  $\rho_1 > \rho_2$  in Class 4, the map will be

$$(19) \quad \begin{aligned} \tilde{x}_1 &= x_1 - \frac{2\rho_2}{\rho_1}, \\ \tilde{x}_2 &= x_2 + \frac{2\rho_2 - 2\rho_1}{\rho_1}, \\ \tilde{x}_3 &= x_3 + 2. \end{aligned}$$

Similarly, in Class 5, if  $\rho_2 < \rho_3$  the map will be

$$(20) \quad \begin{aligned} \tilde{x}_1 &= x_1 + \frac{2\rho_2}{\rho_3}, \\ \tilde{x}_2 &= x_2 - 2, \\ \tilde{x}_3 &= x_3 + \frac{2\rho_3 - 2\rho_2}{\rho_3}, \end{aligned}$$

and

$$(21) \quad \begin{aligned} \tilde{x}_1 &= x_1 + 2, \\ \tilde{x}_2 &= x_2 - \frac{2\rho_3}{\rho_2}, \\ \tilde{x}_3 &= x_3 + \frac{2\rho_3 - 2\rho_2}{\rho_2}, \end{aligned}$$

if  $\rho_2 > \rho_3$ .

The two policies discussed produce different discrete dynamical systems unless all the processing rates are the same. In this case the observation time is  $\rho$  for all the five classes. The two maps become identical and are given by the following function  $f$ :

$$(22) \quad f \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{cases} \begin{pmatrix} x_1 - B \\ x_2 + B \\ x_3 \end{pmatrix} & \text{if } x_1 \geq B \text{ and } x_2 < B \text{ and } x_3 < B, \\ \begin{pmatrix} x_1 + B \\ x_2 \\ x_3 - B \end{pmatrix} & \text{if } x_2 < B \text{ and } x_3 \geq B, \\ \begin{pmatrix} x_1 \\ x_2 - B \\ x_3 + B \end{pmatrix} & \text{if } x_1 < B \text{ and } x_2 \geq B \text{ and } x_3 < B, \\ \begin{pmatrix} x_1 - B \\ x_2 \\ x_3 + B \end{pmatrix} & \text{if } x_1 \geq B \text{ and } x_2 \geq B \text{ and } x_3 < B, \\ \begin{pmatrix} x_1 + B \\ x_2 - B \\ x_3 \end{pmatrix} & \text{if } x_2 \geq B \text{ and } x_3 \geq B. \end{cases}$$

**3. Periodic orbits.** We analyze the periodic orbits occurring in the maps derived in the last section. As an example, we deal with the three-step, two-machine model with equal processing rates. Recall that in this case Pull Policy 1 and Pull Policy 2 are equivalent

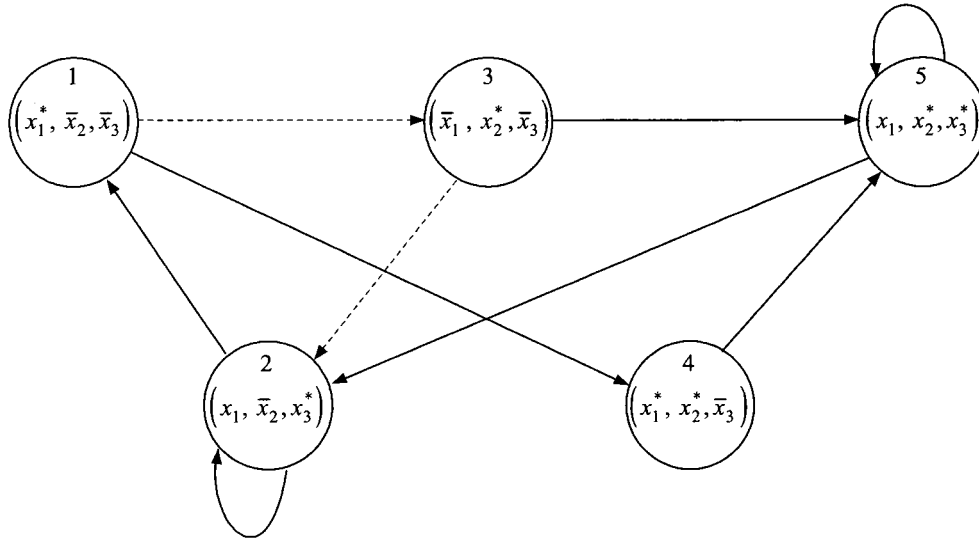


FIGURE 2. In this transition diagram  $(x_1, \bar{x}_2, x_3^*)$  (Class 2) represents the fact that  $x_1$  could attain any value,  $x_2 < B$  and  $x_3 \geq B$ , similarly for  $(x_1, x_2^*, \bar{x}_3)$  (Class 5). The dashed lines represent the transitions for small loads.

TABLE 3. The integer programming model for the three-step, two-machine system when all processing rates are equal to  $\rho$  can be determined by using the information provided in this table

Queue Change	Class 1	Class 2	Class 3	Class 4	Class 5
$x_1$	$-B$	$+B$	$0$	$-B$	$+B$
$x_2$	$+B$	$0$	$-B$	$0$	$-B$
$x_3$	$0$	$-B$	$+B$	$+B$	$0$

and the map generated is given in Equation (22). For this map all orbits are periodic of short period (Period 3 or 4). Consider an initial condition of  $B$  jobs in Queue 1 and a total load of exactly  $L=B$ . Then the trajectory that describes the dynamics of the queues obtained by iterating the map presented in Equation (22) is given by:

$$(23) \quad (B, \bar{x}_2, \bar{x}_3) \rightarrow (\bar{x}_1, B, \bar{x}_3) \rightarrow (\bar{x}_1, \bar{x}_2, B) \rightarrow (B, \bar{x}_2, \bar{x}_3).$$

Each periodic orbit is characterized by its length and the order in which the classes are visited. In this case the trajectory is characterized by

$$(24) \quad \text{Class 1} \rightarrow \text{Class 3} \rightarrow \text{Class 2} \rightarrow \text{Class 1}.$$

We call this a period three orbit of type 1-3-2. We present the system dynamics by *transition diagrams*, which are a directed network where the nodes represent the classes and the arrows represent the allowable transitions between them. See Figure 2 for an example. Table 3 shows the changes in the queue lengths for the transition graph in Figure 2. In this example if the system is in Class 1 then the length of Queue 1, represented by  $x_1$ , decreases by  $B$ ; similarly  $x_2$  increases by  $B$  and  $x_3$  does not change. The other four classes are described similarly in the table. An *integer programming model* is set up to analyze this system, in which the decision variable  $a_i$  represents the number of visits to class  $i$ . The objective function is to minimize the total number of visits to each of the classes. A necessary condition for a periodic orbit to exist can be derived from the fact that over one period the changes in all queues have to add up to zero. Hence the period

of the orbit is given by the smallest integer solutions for the  $a_i$  to the equations

$$(25) \quad \begin{aligned} -Ba_1 + Ba_2 - Ba_4 + Ba_5 &= 0, \\ Ba_1 - Ba_3 - Ba_5 &= 0, \\ -Ba_2 + Ba_3 + Ba_4 &= 0. \end{aligned}$$

It needs to be emphasized that this is only a necessary condition. Hence, additional information needs to be used to determine whether there are additional periodic solutions with larger periods. Inspection of Tables 2 and 3 allows us to determine the throughput: Any time queue  $x_1$  increases by  $B$  we have finished  $B$  jobs and released  $B$  new jobs into the system. This happens when the system is in Class 2 and Class 5. A type 1-3-2 orbit therefore produces an output of  $B$  everytime the system is in Class 2. Since all processing rates are  $\rho$  we produce  $B$  jobs every  $3\rho$  minutes and therefore the periodic orbit has a throughput of  $T = (B \text{ jobs}) / (3\rho \text{ min})$ .

**THEOREM 3.1.** *For the three-step, two-machine system in which all processing rates are the same, only periodic orbits of period three or four exist, independent of the number of jobs in the system, i.e., the load  $L$ , and the batch size  $B$ .*

**COROLLARY 3.1.** *All periodic orbits are of type: 1-3-2 or 1-4-5-2.*

**PROOF 3.1.** We can construct the transition diagram (Figure 2) analytically by considering all possible mappings of values of the initial state  $(x_1(0), x_2(0), x_3(0)) = (a, b, c)$  for all classes. For example, while in Class 1, where  $a \geq B$ ,  $b < B$  and  $c < B$  then the map  $f$  (recall Equations (22)) takes  $(a, b, c)$  to  $(a - B, b + B, c)$ . If  $a < 2B$  then Queue 1 has decreased to a value less than  $B$  and Queue 2 is now greater or equal to  $B$ . Queue 3 has stayed the same. Therefore from Class 1 the system goes to Class 3 which is denoted  $(\bar{x}_1, x_2^*, \bar{x}_3)$  since Queue 1 and Queue 3 are less than a full batch size  $B$ . If however  $a \geq 2B$ , then the system goes from Class 1 to Class 4, which is denoted by  $(x_1^*, x_2^*, \bar{x}_3)$ . By following this analysis for each class we can construct the transition diagram.

The necessary conditions for the existence of a periodic orbit are given by Equations (25). They can be simplified to

$$(26) \quad a_1 - a_5 = a_2 - a_4,$$

$$(27) \quad a_1 - a_5 = a_3.$$

We need to find integer solutions to these equations. If  $a_1 = 0$  then, according to Figure 2, we can never visit Class 3 or Class 4 unless we start there. In addition, a Class 5 is either returning to Class 5 or allows a transition to Class 2. Every time the orbit returns from Class 5 to Class 5 this reduces queue 2 by  $B$ . Hence we eventually transit to Class 2. Since every returning step in the region for Class 2 reduces Queue 3 by  $B$  we eventually transit to Class 1 which is a contradiction to the assumption  $a_1 = 0$ . Hence we set  $a_1 = 1$  and follow the combinatorial tree which is based on Equations (26) and (27).

Notice the case  $a_2 = a_4 = 0$  is not possible because of the observations made from the transition diagram. Since Class 4 transits only to Class 5,  $a_4 \leq a_5$  which excludes any additional branch in Figure 3 coming off the rightmost node  $(a_3 = 1, a_5 = 0)$ . Since we are looking for the smallest integer solution  $a_2 = 2$ ,  $a_4 = 1$  and similar higher integer solutions are not allowed. Therefore there are only two possible minimum period solutions:

$$(28) \quad a_1 = a_2 = a_4 = a_5 = 1, \quad \text{which gives the Period 4 orbits,}$$

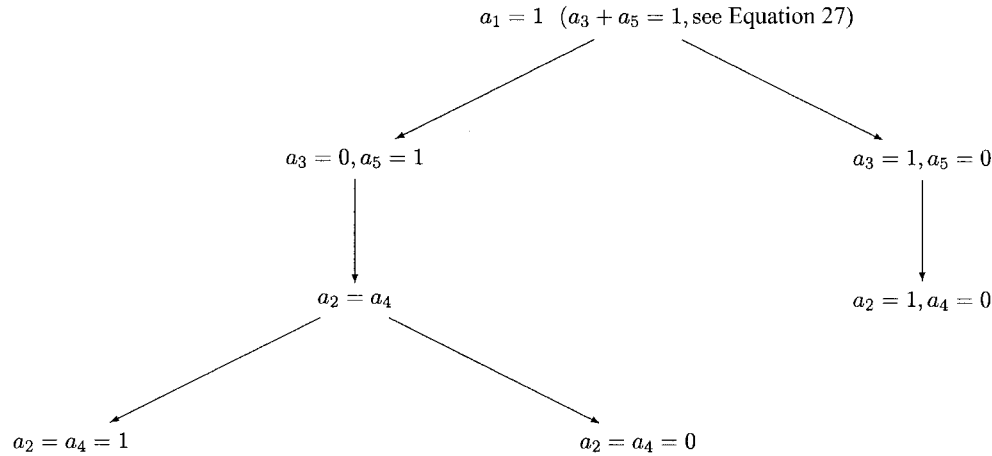


FIGURE 3. Combinatorial tree based on Equations 26 and 27.

TABLE 4. This table summarizes the queue change for the three-step, two-machine system special case in which  $\rho_2 = 53$

Queue Change	Class 1	Class 2	Class 3	Class 4	Class 5
$x_1$	-2	+2	0	$-\frac{106}{114}$	$+\frac{106}{167}$
$x_2$	+2	0	-2	-2	-2
$x_3$	0	-2	+2	$+\frac{334}{114}$	$+\frac{228}{167}$

and

$$(29) \quad a_1 = a_2 = a_3 = 1, \quad \text{which gives the Period 3 orbits.}$$

The type of periodic orbit is inferred from the transition diagram as a 1-4-5-2 orbit or a 1-3-2 orbit. The period three orbit only occurs when the load in the system is low ( $x_1 < 2B, x_2 < B, x_3 < B$ ). In a similar way we can show that no new periodic orbits are created for  $a_1 > 1$ . Hence in this specific case when all the processing rates are the same, we can prove that these two orbits are the only periodic orbits. This has been verified through extensive simulations.

**3.1. A nontrivial case.** Let  $\rho_1 = \rho_3 = 167$  min and  $\rho = 53$  min,  $B = 2$  and let the scheduling policy be the pull policy 1.

The constraint equations

$$(30) \quad \begin{aligned} -2a_1 + 2a_2 - \frac{106}{114}a_4 + \frac{106}{167}a_5 &= 0, \\ 2a_1 - 2a_3 - 2a_4 - 2a_5 &= 0, \\ -2a_2 + 2a_3 + \frac{334}{114}a_4 + \frac{228}{167}a_5 &= 0, \end{aligned}$$

are generated from Table 4. To simplify these equations the transition diagram is considered (see Figure 4). By studying the transition diagram it is determined that the connection between Class 1 and Class 3 exists only when the total load in the system is strictly less than 6. Therefore for low load the trajectories are periodic orbits of Period 3 and of type 1-3-2 (shown in dashed lines in Figure 4). Whenever the load is greater than or equal to 6, then this connection is lost. If the total load is larger than 6 then, excluding the transients, the transition diagram simplifies to the transition diagram presented in Figure 5.

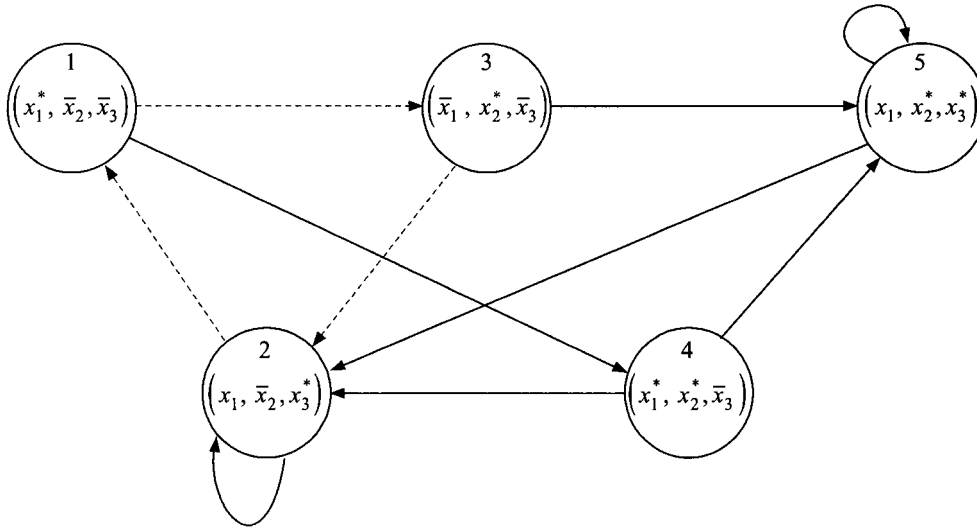


FIGURE 4. Transition diagram for Pull Policy 1 when  $\rho_2 = 53$  min.

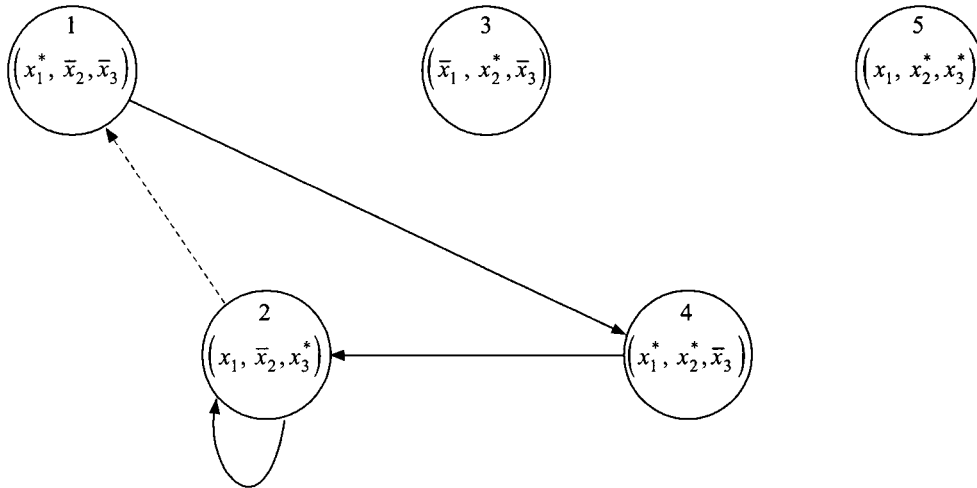


FIGURE 5. Transition diagram for the periodic orbit.

Using this information it can be concluded that variables  $a_3$  and  $a_5$  must be zero and that  $a_1 = a_4$ . Therefore the system of Equations (30) reduces to:

$$(31) \quad 167a_1 = 114a_2.$$

Recall  $a_i$  is integer and even though this relationship has many solutions the minimal one is  $a_1 = a_4 = 114$  and  $a_2 = 167$ . Therefore the periodic orbit has length 395.

Inspecting the sequence of classes in the orbit of length 395 we find that, if the system is in a Class 4 it will map into Class 2 where it may stay for one additional iteration, followed by Class 1 after which the system moves back to Class 4. Therefore only two patterns are found: 4-2-2-1 and 4-2-1. In addition, the 4-2-2-1 sequence is always preceded and followed by the 4-2-1 sequence.

This period 395 orbit takes 55,778 minutes. Class 2 gets visited 167 times and each time 2 jobs are done. Therefore the throughput  $T = (334 \text{ jobs}) / (55,778 \text{ min})$ , which simplifies

to  $T = (1 \text{ job}) / (167 \text{ min})$ . Notice jobs also get done  $((106/167) \text{ jobs})$  while the system is Class 5, which in this case is a transient state.

**3.2. The influence of the processing rates on the period.** In this section we will vary the processing rates and determine their influence on the period of the periodic orbits. We will determine the period of the periodic orbits for arbitrary integer processing rates for the example of a three-step, two-machine problem. We infer that, in general, Pull Policy 1 produces shorter periods than Pull Policy 2. We will prove that, for both policies and rational processing rates, this system does not have nonperiodic orbits and hence chaos cannot exist. We will concentrate our discussion on the regime  $\rho_2 < \rho_3$ ,  $\rho_1 \geq 2\rho_2$ . All other parameter regimes can be treated similarly. However, in some cases some partial simulations might be necessary (see below).

**THEOREM 3.2.** *Assume that  $\rho_2 < \rho_3$ ,  $\rho_1 \geq 2\rho_2$ ,  $B = 2$  and  $L \geq 6$ . Then the minimal period that a periodic orbit can have is*

$$(32) \quad 3\rho_1 - 2\rho_2,$$

for Pull Policy 1, if all the  $\rho$ 's are integer and  $\rho_1$  and  $\rho_1 - \rho_2$  are relatively prime and

$$(33) \quad 3\rho_1\rho_3 + 2\rho_2\rho_3 - \rho_2^2,$$

for Pull Policy 2 provided the  $\rho$ 's are integer and  $\rho_3\rho_1$  and  $\rho_3(\rho_1 + \rho_2) - \rho_2^2$  are relatively prime.

**PROOF 3.2.** Using the results shown in §2, the maps that describe the discrete events are given by  $g_1$  and  $g_2$  for Pull Policy 1 and Pull Policy 2, respectively.

$$(34) \quad g_1 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{cases} \begin{pmatrix} x_1 - 2 \\ x_2 + 2 \\ x_3 \end{pmatrix} & \text{if } x_1 \geq 2 \text{ and } x_2 < 2 \text{ and } x_3 < 2, \\ \begin{pmatrix} x_1 + 2 \\ x_2 \\ x_3 - 2 \end{pmatrix} & \text{if } x_2 < 2 \text{ and } x_3 \geq 2, \\ \begin{pmatrix} x_1 \\ x_2 - 2 \\ x_3 + 2 \end{pmatrix} & \text{if } x_1 < 2 \text{ and } x_2 \geq 2 \text{ and } x_3 < 2, \\ \begin{pmatrix} x_1 - \frac{2\rho_2}{\rho_1 - \rho_2} \\ x_2 - 2 \\ x_3 + \frac{2\rho_1}{\rho_1 - \rho_2} \end{pmatrix} & \text{if } x_1 \geq 2 \text{ and } x_2 \geq 2 \text{ and } x_3 < 2, \\ \begin{pmatrix} x_1 + \frac{2\rho_2}{\rho_3} \\ x_2 - 2 \\ x_3 + \frac{2(\rho_3 - \rho_2)}{\rho_3} \end{pmatrix} & \text{if } x_2 \geq 2 \text{ and } x_3 \geq 2, \end{cases}$$

$$(35) \quad g_2 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{cases} \begin{pmatrix} x_1 - 2 \\ x_2 + 2 \\ x_3 \end{pmatrix} & \text{if } x_1 \geq 2 \text{ and } x_2 < 2 \text{ and } x_3 < 2, \\ \begin{pmatrix} x_1 + 2 \\ x_2 \\ x_3 - 2 \end{pmatrix} & \text{if } x_2 < 2 \text{ and } x_3 \geq 2, \\ \begin{pmatrix} x_1 \\ x_2 - 2 \\ x_3 + 2 \end{pmatrix} & \text{if } x_1 < 2 \text{ and } x_2 \geq 2 \text{ and } x_3 < 2, \\ \begin{pmatrix} x_1 - \frac{2\rho_2}{\rho_1} \\ x_2 - \frac{2(\rho_1 - \rho_2)}{\rho_1} \\ x_3 + 2 \end{pmatrix} & \text{if } x_1 \geq 2 \text{ and } x_2 \geq 2 \text{ and } x_3 < 2, \\ \begin{pmatrix} x_1 + \frac{2\rho_2}{\rho_3} \\ x_2 - 2 \\ x_3 + \frac{2(\rho_3 - \rho_2)}{\rho_3} \end{pmatrix} & \text{if } x_2 \geq 2 \text{ and } x_3 \geq 2. \end{cases}$$

Let  $a_i$  be the number of times the trajectory visits class  $i$  ( $i = 1, 2, 3, 4, 5$ ). If an orbit is periodic then the total change over one period has to be zero and hence the following equations must have integer solutions.

Pull Policy 1:

$$(36) \quad \begin{aligned} -2a_1 + 2a_2 - \frac{2\rho_2}{\rho_1 - \rho_2} a_4 + \frac{2\rho_2}{\rho_3} a_5 &= 0, \\ 2a_1 - 2a_3 - 2a_4 - 2a_5 &= 0, \\ -2a_2 + 2a_3 + \frac{2\rho_1}{\rho_1 - \rho_2} a_4 + \frac{2(\rho_3 - \rho_2)}{\rho_3} a_5 &= 0; \end{aligned}$$

Pull Policy 2:

$$(37) \quad \begin{aligned} -2a_1 + 2a_2 - \frac{2\rho_2}{\rho_1} a_4 + \frac{2\rho_2}{\rho_3} a_5 &= 0, \\ 2a_1 - 2a_3 - \frac{2(\rho_1 - \rho_2)}{\rho_1} a_4 - 2a_5 &= 0, \\ -2a_2 + 2a_3 + 2a_4 + \frac{2(\rho_3 - \rho_2)}{\rho_3} a_5 &= 0. \end{aligned}$$

Once these equations are solved, the shortest possible period of a periodic orbit will be given by  $\min \sum_{i=1}^5 a_i$ . Note that the two systems are very similar and that in both systems only two of the equations are linearly independent.

It is easy to see that the transition diagram in both cases is given by Figure 4. We infer that for a total load  $L > 6$ , Class 3 is transient and hence, on the periodic orbit,  $a_3 = 0$  in both cases. In addition there is only one way to get into Class 4 and it is via Class 1. Class 4 is the only successor to Class 1. Hence  $a_1 = a_4$ . Inserting this into Equations (36) and (37) we get for Pull Policy 1,  $a_5 = 0$ , and

$$(38) \quad \rho_1 a_1 = (\rho_1 - \rho_2) a_2;$$

and for Pull Policy 2,

$$(39) \quad a_5 = \frac{\rho_2}{\rho_1} a_1,$$

$$(40) \quad (\rho_3(\rho_1 - \rho_2) - \rho_2^2) a_1 = \rho_3 \rho_1 a_2.$$

To solve Equations (38) and (40) we need to make assumptions about the rates  $\rho_i$ . For integer  $\rho_i$  and  $a_i$  and if the coefficients on the left- and right-hand side of Equations (38) and (40) are relatively prime, we find:

$$\text{Pull Policy 1: } a_2 = \rho_1 \text{ and } a_1 = (\rho_1 - \rho_2);$$

$$(41) \quad \text{Pull Policy 2: } a_1 = \rho_3 \rho_1 \text{ and } a_2 = \rho_3(\rho_1 + \rho_2) - \rho_2^2.$$

Summing up the  $a_i$  produces the periods of the theorem.

REMARK. This theorem gives a necessary condition for the shortest possible periodic orbit in the system. Uniqueness or even existence of a periodic orbit with a specific period is not guaranteed. However, in extensive simulations with many different systems we were always able to recover the minimal periodic orbit predicted by the theorem. In every case, only one periodic orbit was found for all initial conditions. This is illustrated in the following example.

EXAMPLE 3.1. Let  $\rho_1 = 167$ ,  $\rho_2 = 53$ ,  $\rho_3 = 167$ . Recall this is the case studied in §3.1. Then the solution for Pull Policy 1 is:

$$(42) \quad \begin{aligned} a_1 &= 114, \\ a_2 &= 167, \\ a_4 &= 114, \end{aligned}$$

and therefore the period  $\tau_1$  is (cf. Equation (32)):

$$\tau_1 = 114 + 167 + 114 = 3 \times 167 - 2 \times 53 = 395,$$

as was shown in Example 3.1. For Pull Policy 2,

$$(43) \quad \begin{aligned} a_1 &= 167^2, \\ a_2 &= 167^2 + 167 \times 53 - 53^2, \\ a_4 &= 167^2, \\ a_5 &= 167 \times 53, \end{aligned}$$

and therefore the period  $\tau_2$  is:

$$(44) \quad \tau_2 = 167^2 + 167^2 + 167 \times 53 - 53^2 + 167^2 + 167 \times 53 = 98,560.$$

In both cases, simulations with many different initial conditions lead to periodic orbits with the predicted periods.

THEOREM 3.3. *There exist rational processing rates such that the minimal period of any periodic orbit becomes arbitrarily long. No periodic orbits exist for irrational processing rates.*

PROOF. 3.3. Let  $k = \rho_1 - \rho_2$  be a decimal number with  $n$  digits. In this case Equation (38) becomes:

$$(45) \quad \rho_1 \times a_1 = k \times a_2,$$

for Pull Policy 1. This problem is equivalent to

$$(46) \quad \rho_1 10^n \times a_1 = k 10^n \times a_2,$$

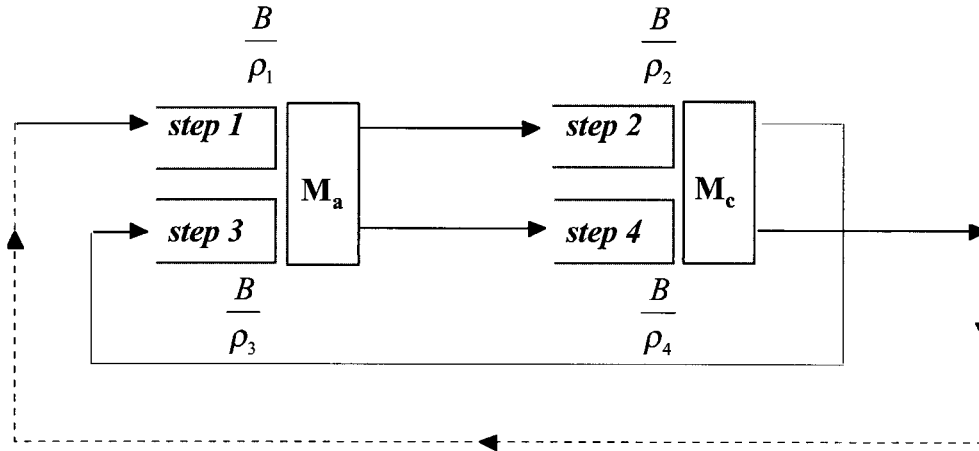


FIGURE 6. Four step, two-machine system.

and hence, as long as  $167 \cdot 10^n$  and  $k \cdot 10^n$  are relatively prime the smallest integer solution to this equation is  $a_1 = k \cdot 10^n$  and  $a_4 = 167 \cdot 10^n$ . This implies that there exist rational processing rates for which the minimal period of any orbit can be made arbitrarily large. In particular, irrational processing rates lead to quasiperiodic behavior (the orbit of any initial point approaches that point arbitrarily closely infinitely many times but never hits the initial point exactly (Arrowsmith and Place 1992).

**THEOREM 3.4.** *The three-step, two-machine system under a pull policy with rational processing rates does not have chaotic orbits. All orbits are either periodic or eventually periodic.*

**PROOF 3.4.** It is easy to verify that the queues  $(x_1(t), x_2(t), x_3(t))$  always change by rational numbers. Hence for a given load  $L$ , the phase space for these queues is partitioned by a finite grid. The grid size is determined by the common denominators over all queue changes. Hence the dynamics of the map generated by the interevent times corresponds to maps defined on a finite (possibly very large) number of attainable values for all queues (a finite state automaton). Hence the system will have to come back to a state that it has already visited and therefore there exist only periodic or eventually periodic orbits.

**4. Conclusion.** We have shown for a three-step, two-machine problem how to generate a model of piecewise linear maps whose long term dynamics are periodic orbits. We have developed a methodology to calculate a priori the periods of these orbits. The main ingredients of this method scales up to larger systems in the following ways: For any pull or push scheduling policy, we will get piecewise constant systems of differential equations, which, after choosing an observation time, will lead to piecewise linear maps. Each part of such a piecewise linear map then defines a class and transition diagrams can be set up. Again, as all processing rates are rational all orbits will become periodic or eventually periodic. Since one of the necessary conditions of chaos is the existence of nonperiodic orbits we find that no chaos can occur in these systems. The ability to calculate periodic orbits explicitly rests on our ability to simplify the transition diagram by inspection of a combinatorial tree like in Figure 3. For larger systems such trees will become much too complicated. However, combining this combinatorial approach with information obtained from simulations has allowed us to analyze more complicated systems (Diaz-Rivera 1997). Figure 6 shows a four-step, two-machine system. Its phase space has

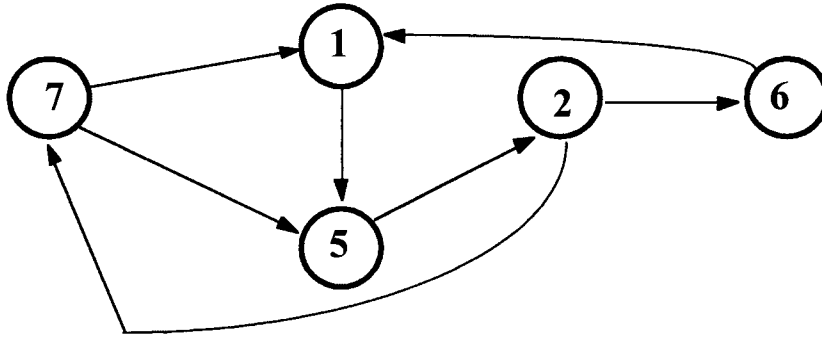


FIGURE 7. Transition diagram for Pull Policy 1 used in the four-step, two-machine model.

TABLE 5. Summary of queue changes for four-step, two-machine system when all processing rates are different and relatively prime

Queue Length	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
$x_1$	-2	0	0	+2	$-\frac{106}{170}$	$+\frac{334}{96}$	$+\frac{304}{223}$	0
$x_2$	+2	0	-2	0	-2	0	$+\frac{142}{223}$	-2
$x_3$	0	-2	+2	0	$+\frac{446}{170}$	$-\frac{142}{96}$	0	$+\frac{228}{167}$
$x_4$	0	+2	0	-2	0	-2	-2	$+\frac{106}{167}$

8 classes. Simulations of the system for

$$(47) \quad \rho_1 = 223 \text{ min}, \quad \rho_2 = 53 \text{ min}, \quad \rho_3 = 167 \text{ min}, \quad \rho_4 = 71 \text{ min},$$

and Pull Policy 1 lead to orbits that are not periodic for millions of iterations. The general 8-class transition diagram does not lead to an integer programming model that is tractable. However, a much simpler transition diagram can be constructed by running the analytical simulation for a few iterations (see Figure 7). We find that, after transient behavior, Class 3, Class 4, and Class 8 are never visited and that Class 2 and Class 5 are visited with the same frequency. Typically, about a 1,000 iterations have been enough to be able to make this observation reliably. Using the integer programming type of model this translates to having:

$$(48) \quad a_3 = a_4 = a_8 = 0 \quad \text{and} \quad a_2 = a_5.$$

Table 5 shows the queue changes in this situation. Using an integer programming solver the minimal solution is:

$$(49) \quad \begin{aligned} a_1 &= 1,097,944, \\ a_2 &= 1,345,805, \\ a_5 &= 1,345,805, \\ a_6 &= 567,312, \\ a_7 &= 778,493, \\ a_8 &= a_3 = a_4 = 0. \end{aligned}$$

Therefore the period has length 5,135,359 which was verified by running a simulation. Table 5 shows that jobs are done when the system visits Class 4, Class 6, and Class 7. Therefore the throughput for this system (the rate at which jobs get done in jobs/min) can be determined by  $T = ((334/96) \times a_6 + 2 \times a_7) / 688,498,005$ ; this simplifies to  $T = (1 \text{ job}) / (195 \text{ min})$ , which is equivalent to  $T = 2 / ((223) + (167))$ .

Future papers will describe details of the dynamics of these systems in phase space and their dependence on processing rates. Specifically, we will show that resonances between the processing rates define bifurcation points. Systems near such bifurcation points show dramatically different periods and throughputs in addition to extremely long transients (Diaz-Rivera et al. 2000). Further study will concentrate on identifying the parameter space in which optimal throughput can be achieved.

**Acknowledgments.** The authors would like to thank Karl Kempf (INTEL) for his introduction to the problems of manufacturing systems, his continued support and many interesting discussions. This work was partially supported by a grant from INTEL and NSF grant DMS 9501077.

### References

- Alligood, K. T., T. D. Sauer, J. A. Yorke. 1996. *Chaos, An Introduction to Dynamical Systems*. Springer-Verlag, New York, 48.
- Arrowsmith, D. K., C. M. Place. 1992. *Dynamical Systems*. Chapman and Hall.
- Beaumariage, T., K. Kempf. 1994. The nature and origin of chaos in manufacturing systems. *Proc. of the 5th SEMI/IEEE Adv. Semiconductor Manufacturing Conf.* 489–496.
- Chase, C. J., J. Serrano, P. Ramadge. 1993. Periodicity and chaos from switched flow systems: Contrasting examples of discretely controlled continuous systems. *IEEE Trans. Automat. Control.* **38** 70–83.
- Dai, J. G., G. Weiss. 1996. Stability and instability of fluid models for re-entrant lines. *Mathe. Oper. Res.* **21**(1) 115–134.
- , K. H. Yeh, C. Zhou. 1997. The QNET method for re-entrant queueing networks with priority disciplines. *Oper. Res.* **45** 610–623.
- Diaz-Rivera, I. 1997. The dynamics of queues of re-entrant manufacturing systems. Ph.D. Thesis, Arizona State University, AZ.
- , D. Armbruster, T. Taylor. 2000. Phase space, bifurcations and resonances in re-entrant manufacturing systems. Working paper, Arizona State University, Tempe, AZ.
- Kleinrock, L. 1976. *Queueing Systems, Volume 2: Computer Applications*. John Wiley & Sons, New York.
- Kumar, P. R. 1993. Re-entrant lines. *Queueing Systems Theory Appl.* **13** 87–110.
- , S. P. Meyn. 1995. Stability of queueing networks and scheduling policies. *IEEE Trans. Automat. Control* **40** 251–260.
- Lu, S. H., P. R. Kumar. 1991. Distributed scheduling based on due dates and buffer priorities. *IEEE Trans. Automat. Control* **36** 1406–1416.
- Walsh J. A., G. R. Hall, B. Elenbogen. 1996. Computer protocol and torus maps. *Dynam. Stability Systems: Internat. J.* **11** 239–263.
- Wolff, R. W. 1989. Stochastic models and the theory of queues. *International Series in Industrial and Systems Engineering*, Prentice-Hall, Englewood Cliffs, NJ.

I. Diaz-Rivera: Department of Mathematics, Arizona State University, Tempe, Arizona 85287-1804

D. Armbruster: Department of Mathematics, Arizona State University, Tempe, Arizona 85287-1804;  
e-mail: armbruster@asu.edu

T. Taylor: Department of Mathematics, Arizona State University, Tempe, Arizona 85287-1804