

Control and synchronization in switched arrival systems

Bart Rem

Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Dieter Armbruster^{a)}

Department of Mathematics, Arizona State University, Tempe, Arizona 85287-1804

(Received 16 April 2002; accepted 13 June 2002; published 21 February 2003)

A chaotic model of a production flow called the switched arrival system is extended to include switching times and maintenance. The probability distribution of the chaotic return times is calculated. Scheduling maintenance, loss of production due to switching, and control of the chaotic dynamics is discussed. A coupling parameter to couple switched arrival systems serially, based on lost production, is identified. Simulations of three parallel and three serial levels were performed. Global synchronization of the switching frequencies between serial levels is achieved. An analytic model allows one to predict the self-balancing properties of the serial system. © 2003 American Institute of Physics. [DOI: 10.1063/1.1500284]

Simple abstractions of real production systems in factories have previously been shown to lead to chaotic behavior. Specifically, a system of parallel machines with one switching server shows chaotic switching behavior. Since that model is lossless, the chaotic dynamics does not influence the production output. This research studies the behavior of such production systems under more realistic assumptions. We introduce production loss due to switching and due to maintenance and study their influence on controlling the system onto periodic orbits. In addition since most factories are a mixture of parallel and serial machines, we study serially coupled layers of parallel machines. We show that they can lead to self-balanced chaotic production. This result suggests that attempts to fine tune the equal distribution of work inside a factory might not be necessary since the system balances itself.

I. INTRODUCTION

A switched arrival system is a manufacturing system that consists of N parallel machines and one switching server that distributes work over the machines. A machine is modeled as a buffer which contains a continuous amount of work that drains at a constant work rate. The server continues to fill a buffer until another buffer empties. At that moment the server switches instantaneously to the empty buffer. The rate at which the server fills a buffer is equal to the sum of all the work coming out of all the buffers. This models a policy known in industrial engineering as CONWIP policy (constant work in progress) which releases a new product into a machine or factory once a finished product has been released. Hence the system is closed and the total amount of work in the system is constant. Let $x_i(t)$ denote the amount of work in buffer i at time t . Wlog we choose the total amount of work in all of the buffers to be one, i.e.,

$$\sum_{i=1}^N x_i = 1. \quad (1)$$

We also choose the time to produce the total amount of work as the characteristic time and set it to one also. Hence the production rates ρ_i satisfy

$$\sum_{i=1}^N \rho_i = 1. \quad (2)$$

Let j be the position of the server. The server switches to the first buffer that empties. This happens after a time

$$\tau = \min_{i \neq j} \{x_i(t) / \rho_i\}. \quad (3)$$

For $t_0 \leq t \leq t_0 + \tau$ the buffer state is determined by the following linear equations:

$$x_i(t) = \begin{cases} x_i(t_0) - \rho_i(t - t_0) & \text{for } i \neq j \\ x_i(t_0) + (1 - \rho_i)(t - t_0) & \text{for } i = j. \end{cases} \quad (4)$$

When this system is sampled at the times when a buffer empties the continuous model becomes a discrete-event model of the form

$$G(x) = x + \min_{k \neq j} \left(\frac{x_k}{\rho_k} \right) (\mathbf{1}_j - \rho), \quad (5)$$

where $\mathbf{1}_j$ is a vector with all zeros except for a one in the j th position and ρ is a vector containing the work rates ρ_i . The buffer state evolves on the simplex $\sum x_i = 1$. For $N=3$ machines (see Fig. 1) this simplex becomes an equilateral triangle and $G(x)$ maps the boundaries of the triangle onto each other. Since the total work is conserved, $G(x)$ can be further reduced to a one-dimensional map. For the completely symmetric case, $\rho_i = 1/3$, this map becomes a simple $1 - 2x \bmod 1$ map which is chaotic (Chase *et al.*¹). Geometrically this is reflected in the fact that each side of the equilateral triangle is mapped onto the two other sides, leading to expansion of the map, a positive Liapunov exponent, and sensitive dependence on initial conditions. Figure 2 shows a typical orbit on the simplex as well as for the associated buffers for equal work rates. Every time the trajectory hits the side of the equilateral triangle, a buffer is empty and the server switches to the empty buffer. The sides of the triangle correspond to states when one buffer is empty. The trajectory starts at the open circle and ends at the full circle.

^{a)}Electronic mail: dieter@source.la.asu.edu

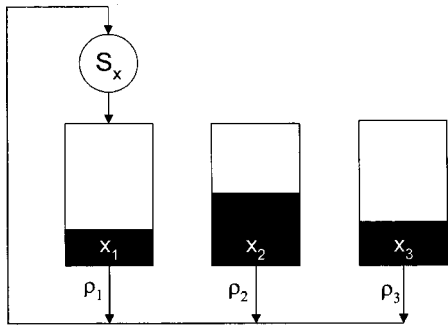


FIG. 1. A switched arrival system for three machines.

II. CONTROL

A. Lossless control

Horn and Ramadge² investigated the effect of upper and lower limits placed on the volumes of the buffers. Upper and

lower limits change the moment that a switch is made. By adding a lower limit, the server switches to the buffer that has reached the lower limit. So, buffers never run empty. Lower limits do not change any of the system dynamics. In state space, the equilateral triangle shrinks in size, but maintains the same shape.

Upper limits do change the system dynamics. The server either switches when another buffer has run empty or, when the upper limit of the buffer that the server is filling, is reached. In state space, this changes the trajectory triangle to a hexagon. When the side of the hexagon is reached the server switches. By adding a large enough upper limit, invariant regions are created. The remainder of the map is expanding; so all solutions are pushed into the invariant regions. The system stabilizes on a certain periodic orbit.

Horn and Ramadge² proposed that upper limits could be used to control the system on any arbitrary periodic orbit. A

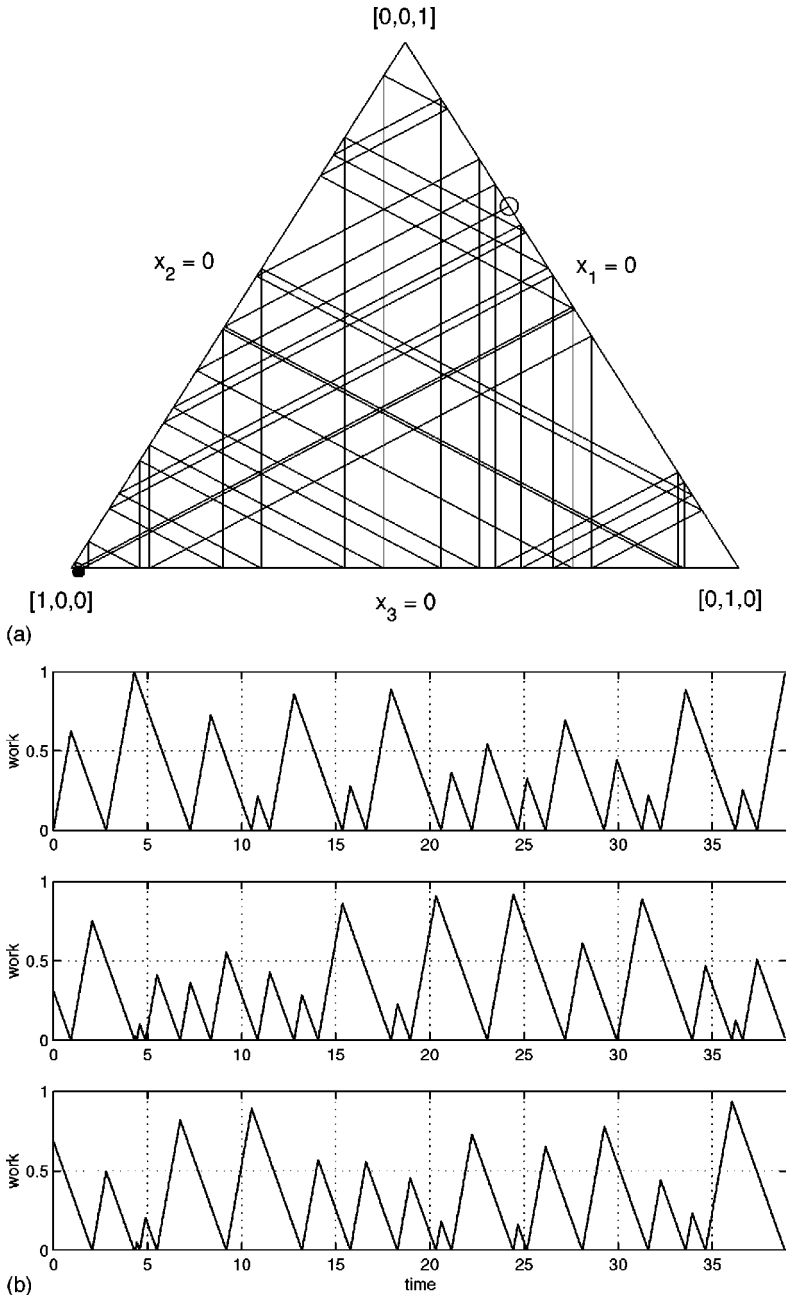


FIG. 2. (a) Trajectory in state space; (b) buffer levels as a function of time.

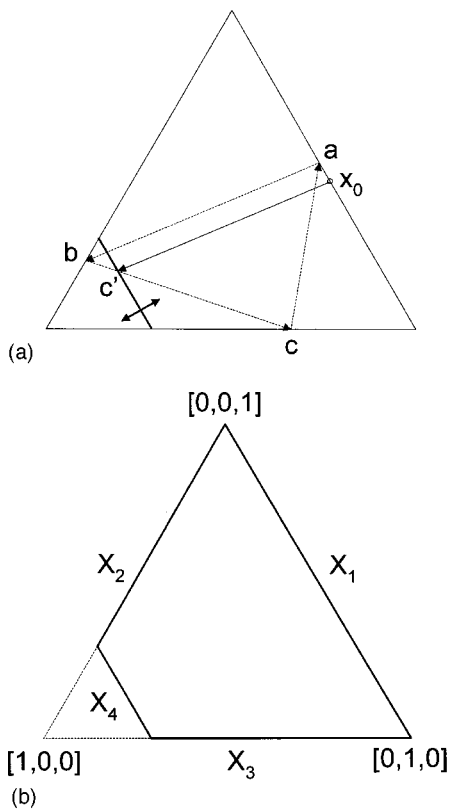


FIG. 3. Control to a periodic orbit by varying the maximal processing time.

close resemblance of this idea was used by Ueda *et al.*^{3,4} to control the switched arrival system to a period-3 orbit. Their control law states that the server remains at its current location until a buffer empties or the server has been filling the current buffer for a time larger than a specified time Δ . Δ is chosen in such a way that the trajectory moves onto the desired periodic orbit. It is calculated for every iteration.

Stabilization of the period orbit is based on changing the simplex X . Going from one side of the triangle to another side takes a certain amount of time. By limiting this time, the simplex X is changed in a way that the corners regions are cut off. Figure 3 explains geometrically how the control law works. Let the initial buffer state be x_0 and the periodic orbit be (a, b, c) . Then Δ is selected in such a way that the side X_4 goes through c' . This forces the server to make an early switch. The trajectory is now on the periodic orbit, and the feedback is switched off. Since errors grow exponentially, the difference between the desired trajectory and the actual trajectory is monitored and control is initiated once the difference exceeds a predetermined value ϵ . Katzorke *et al.*⁵ have proposed a similar control rule to the one that we propose in Sec. II C. They attach a fixed cost to every switching and find numerically that the period-3 orbit is the one with the lowest switching costs.

B. Switching time and maintenance

Previous control attempts did either not include any evaluation function³ or just penalized switches by an arbitrary amount.⁵ The fundamental reason for this lack of a control goal is that all the chaotic dynamics was in a practi-

cal sense irrelevant since the major performance measure for all production systems, the output, was not affected. Since all queues always produced to their full ability the output always was equal to 1. Hence, in order to improve the model we introduced an effect that leads to a reduction in output. One way, arguably a very natural way, is to assume that switching between buffers takes a time τ_s . During this time the empty machine produces nothing. Specifically we choose the following switching scenario: A buffer runs empty; for the next τ_s time units the server continues to serve its old buffer but with a reduced influx equal to the production of the two nonempty buffers. After a time τ_s the server switches into the empty buffer and all three machines resume full production. As a result, switching costs production. For simplicity the switching time is chosen equal for all buffers. This scenario can also account for regular shutdowns for maintenance which would be set whenever a buffer runs empty. The maintained machine is taken out of production and the server switches back and forth between the two working machines. Note that if a second buffer runs empty while a first one is empty already and prepares to switch, the system is run as a single machine until the first switch can be executed followed after the appropriate switching time with the second switch.

Return times

Before we start controlling the system we need to understand the return time dynamics. We define the return time τ_n as the time it takes for the server to return to the same empty buffer the n th time. We restrict ourselves to the case where all work rates are equal and hence $\rho_i = \frac{1}{3}$. Figure 4 shows a plot of the return time τ_{n+1} as a function of the return time τ_n . This figure resembles an infinitely repeated and sharpened tent map. The return time zero corresponds to the orbit going into one of the corner points. Every image τ_{n+1} seems to have infinitely many preimages. This fact can be geometrically understood from Fig. 5. The maximum return time can be calculated by going backwards in time for the orbit that ends in one corner and starts in another. Numerically we find $\tau_{max} = 4.5$. Geometrically we can calculate the orbit from Fig. 6 to be

$$L_{a \rightarrow b} = \frac{\sqrt{3}}{2} + \sqrt{3} \sum_{n=1}^{\infty} \frac{1}{2^n} = \frac{3}{2} \sqrt{3}.$$

The scaling between phase space and time is $\alpha = \sqrt{3}$, and hence we find the maximal time as $\tau_{max} = \frac{9}{2}$.

From geometrical inspection of Figs. 6 and 5 we also note that the numerically generated repeated tent maps can be calculated analytically. They have slopes $|2^{n+1}|$ and a width of $(1/2^n) \tau_{max}$ for the n th tent map. For every τ_{n+1} its two preimages on the first tent map correspond to an orbit that has just one switch before returning to the same side of the triangle, the two preimages on the n th tent map correspond to an orbit with n switches before returning.

Since the preimages of a given τ_{n+1} have the same slopes for all τ_{n+1} , the Frobenius Perron operator maps the uniform density to itself.⁶ Therefore we can find the invariant density for the return times as

$$p_n = \frac{2}{9}, \tag{6}$$

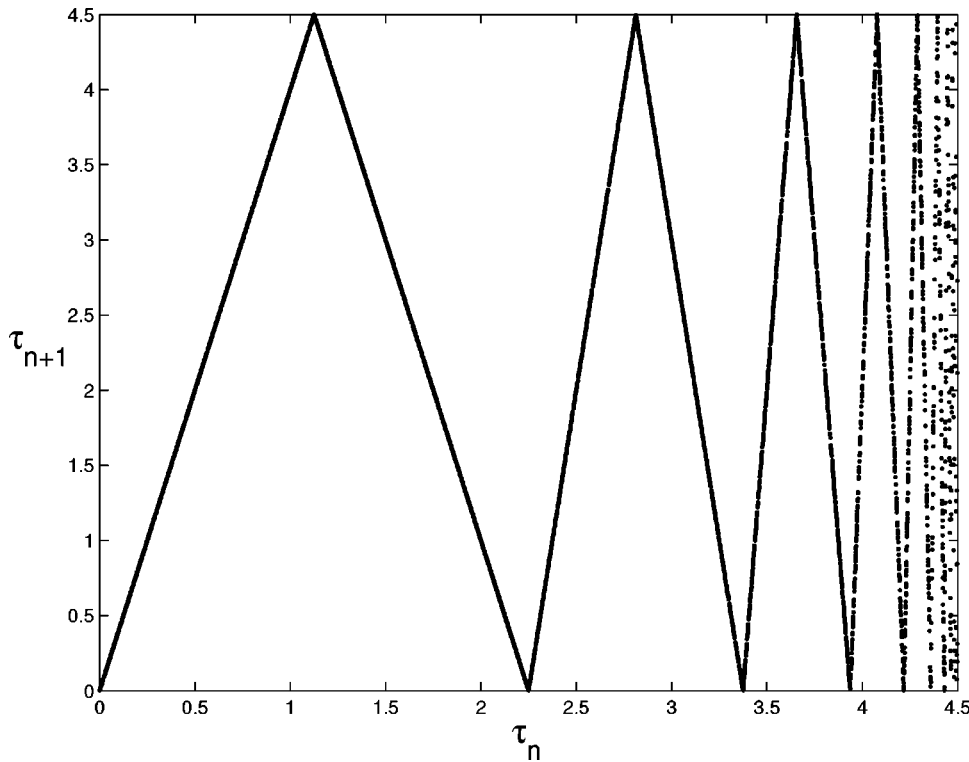


FIG. 4. Return time plot for equal work rates.

which gives us an average return time for the dense orbit

$$\tau_d = \frac{1}{2} \tau_{\max} = \frac{9}{4} \tag{7}$$

and a variance

$$\sigma_\tau^2 = \langle \tau^2 \rangle - \tau_d^2 = \frac{27}{16}. \tag{8}$$

These are relevant quantities in particular for maintenance: average return time and maximal return time describe the average and maximal time interval between maintenance.

We can also use Fig. 4 to prove almost rigorously that the period-3 orbit has the lowest switching frequency. From Fig. 4 we can read off the times between successive returns to the same empty buffer. While this is a necessary condition for a periodic orbit it is clearly not sufficient since the levels in the other two buffers need not be the same. For instance,

the return time for an orbit that makes just one switch before returning to the same edge of the simplex is bounded by

$$0 < \tau_2 < \frac{9}{4},$$

which leads to a switching frequency estimate of

$$\infty > \nu_2 > \frac{8}{9}.$$

In a similar way we can get the frequency estimates for orbits that make two and three switches before returning to the same buffer,

$$\frac{4}{3} > \nu_3 > \frac{8}{9},$$

$$\frac{32}{27} > \nu_4 > \frac{64}{63}.$$

The frequency of the period-3 orbit is 1. It is easy to see that the frequency of all orbits that make $n - 1$ switches before returning to the same buffer is bigger than 1 for $n \geq 4$. This

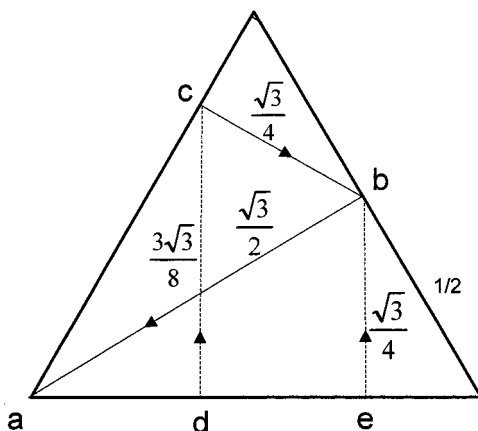


FIG. 5. Shortest orbit before the server returns to the original buffer: $e \rightarrow b \rightarrow a$. The next shortest orbit before the server returns to the original buffer is $d \rightarrow c \rightarrow b \rightarrow a$.

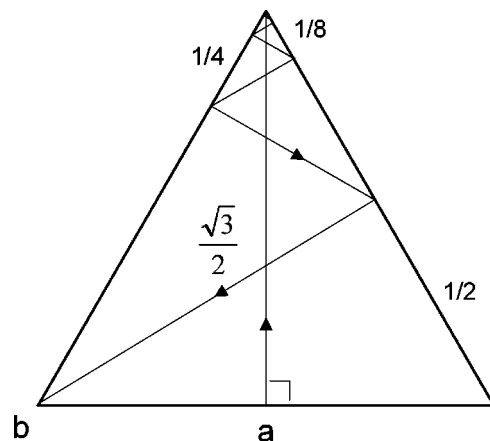


FIG. 6. Longest trajectory before the server returns to the same buffer.

TABLE I. Switching frequency for different values of ρ .

ρ_1	ρ_2	ρ_3	Switching freq
1/18	1/18	16/18	0.40
1/9	1/9	7/9	0.74
2/9	1/9	6/9	0.98
2/9	2/9	5/9	1.18
3/9	1/9	5/9	1.13
3/9	2/9	4/9	1.29
3/9	3/9	3/9	1.33
4/9	1/9	4/9	1.19

leaves periodic orbits of higher period that are composed of orbits that return to the same empty buffer several times before the orbit becomes actually periodic. The following cases may happen.

(1) If there is a suborbit that switches more than two times before returning, the overall orbit will have a frequency bigger than 1.

(2) If the period of the orbit is even (say $2n$) then it could possibly be constructed through n suborbits with just one switch before returning to the empty buffer which would have a lower limit of $\nu_{2n} = \frac{8}{9}$. However, not all of these switches can go between just two buffers. There must be at least one switch that happens at the third buffer. Hence relative to this buffer there exists a suborbit with more than two switches before the orbit returns and hence this periodic orbit has a frequency higher than 1.

(3) The argument in (2) does not hold if $2n$ is also divided by 3. In that case we may have that every periodic orbit is composed of an equal number of suborbits with two switches before a return to the same empty buffer. Therefore the estimates for the switching frequency are the same as for ν_3 . Period 6 is the first such case and while it can be shown explicitly that the period-6 orbits have higher switching frequency we do not know how to prove that for instance the period-39 orbit does not have a smaller switching frequency than 1.

(4) The dense orbit will have the average return time of $\frac{9}{4}$ and will do three switches on average. Hence the frequency for the dense orbit is $\nu_{\text{dense}} = \frac{4}{3}$.

The switching frequency also depends on the work rates of the buffers. For unequal work rates the tent maps become asymmetric and do not lie on top of each other. Simulating the dense orbit we find the switching frequencies of Table I. For equal processing rate we recover the theoretical value for the dense orbit of 1.33. Note that the highest switching frequency is reached for the case with equal production rates. This leads to an interesting result when production has to be increased for a short-term boost. With the assumption that increased production can be achieved by increasing start rates temporarily, then a higher production is reached when the excess production is produced in one single machine rather than split over all machines equally assuming that the single machine can increase its start rate to the desired level.

C. Control law

The only cause for the production rate to be less than optimal is the assumption that switching costs time. During

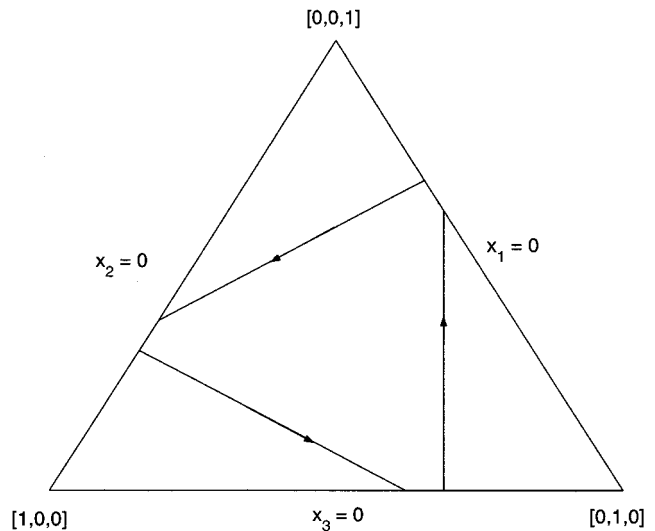


FIG. 7. Period-3 orbit with switching time $\tau_s = 0.2$.

this time the empty machine does not produce anything. The most favorable orbit to stabilize is the one that generates the highest production rate. Since production is lost when switches are made, the orbit with the lowest switching frequency is optimal, which is the period-3 orbit. It must be noted that the discussion in Sec. II B did not yet contain switching time. If a buffer turns empty, that buffer is taken out of the equation for the time it takes the server to switch; the system dynamics are formed with two instead of three buffers. Now the trajectories evolve on the boundary of the equilateral triangle (see Fig. 7). For small switching time τ_s the comparison of switching frequencies between different orbits performed in Sec. II B is still valid, since the constant amount τ_s is added to all return times. If τ_s is large, then it may change the dynamics. We disregard this case since we cannot imagine its practical relevance.

Since the two period-3 orbits can be calculated directly the control law is very simple: If a buffer empties and the level in the buffer that the server is filling is more than ϵ_c lower than the target level for the periodic orbit, the server does not switch but keeps filling the same buffer until the target level is reached within an ϵ_t precision. Figure 8 shows a situation when control is applied. Buffer 3 has just emptied and the current position in buffer 1 is lower than the target level. Now, buffer 3 is kept empty for a certain time. During this time, the buffer levels of the other two buffers reach the target position. The values for ϵ_t and ϵ_c are chosen as follows: ϵ_t states how close we can get to the periodic orbit and is a property of the machines; the smaller ϵ_t the better. When the trajectory is outside an ϵ_c region around the target the orbit is controlled back to the periodic orbit. Note that the map expands everywhere and therefore all errors grow exponentially. Since there is no initial setup time for a control action, control costs the least when the orbit is controlled after every switch. Hence we choose $\epsilon_c = \epsilon_t$.

Figure 9 shows production loss for an orbit controlled on period 3 versus production loss for the uncontrolled (dense) orbit as a function of the switching time. While the controlled orbit has lower switching frequency and hence lower

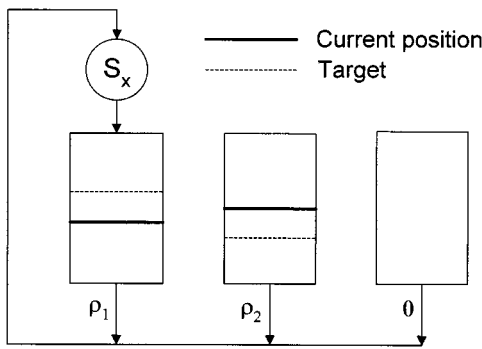


FIG. 8. Example of a moment when control is applied: keep buffer 3 empty until current position reaches target.

production loss due to the switching time we have additional control costs due to the fact that we need to keep the trajectory on the unstable period-3 orbit. Figure 9 does not include the initial control since we are comparing the steady state flows. However, in a real production environment we might run the system only for a finite amount of time. In that case the initial control cost may play a bigger role. To minimize the initial control costs we can, rather than control immediately onto the period-3 orbit, wait until the dense orbit gets close enough to a period-3 point and then start the control. We note that the exponential growth away from the period-3 orbit is not strong enough to cause significant additional production loss due to control. However, we can use the switching frequency to get a good estimate on the amount of uncertainty that would make the loss due to controlling onto the period-3 orbit equal to the gain due to lower switching losses relative to the dense orbit. Since the average time between switches for the period-3 orbit is one we have the total time between switches,

$$\tau_{total} = 1 + \tau_{switch} + \tau_{control},$$

which leads to a production loss of

$$l_c = \frac{\tau_{switch} + \tau_{control}}{\tau_{total}} \rho,$$

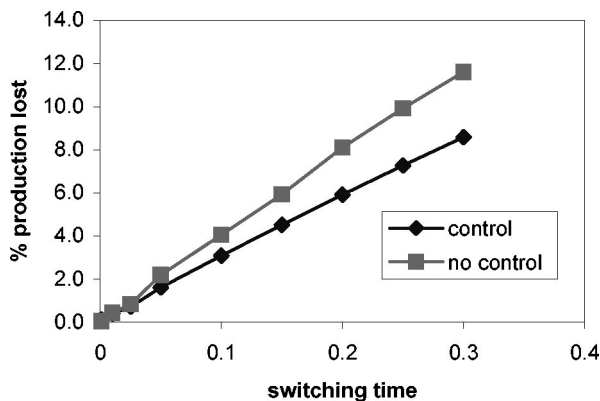


FIG. 9. Production loss in percent as a function of switching time for the dense and the controlled period-3 orbit for equal production rates ρ_i and $\epsilon_i = 1/1000$.

assuming a constant production rate ρ for all machines. The average time between switches for the dense orbit is $3/4$ and hence the production loss due to switching is

$$l_d = \frac{\tau_{switch}}{\frac{3}{4} + \tau_{switch}} \rho.$$

By setting $l_c = l_d$ we get that $\tau_{control} = \tau_{switch}/3$. Since time and buffer levels are related via ρ and since the system grows errors by a factor 2 from one switching to the next we find that the critical error level immediately after switching approximately becomes

$$\epsilon = \rho^{\frac{1}{2^3}} \tau_{switch}. \tag{9}$$

Hence for an uncertainty in the buffer level that is greater than ϵ the cost due to control is bigger than the gain due to lower switching.

III. SYNCHRONIZATION IN NETWORKS OF SWITCHED ARRIVAL SYSTEMS

A typical production line does not just exist of parallel machines. Typically there are several layers of parallel machines feeding into each other. This chapter focuses on the behavior of networks of sequential switched arrival systems. To be specific we study three layers X,Y,Z of three parallel machines each. Layer X feeds into layer Y, Y feeds into layer Z and Z feeds back to X. The individual systems and the switching policy are the same as discussed for the single system case. No control is applied. There is a global mass balance,

$$\sum_{i=1}^3 x_i(t) + \sum_{i=1}^3 y_i(t) + \sum_{i=1}^3 z_i(t) = 3. \tag{10}$$

However, we do not attempt to constrain the mass balance for every layer. Figure 10 shows a schematic of the production flow.

A. Coupling parameter

As discussed previously, in the regular switched arrival system there is never any variation in the system output and therefore the fact that the server switches chaotically does not register. Hence, in that case the three servers are completely independent and, as long as all production rates are $\sum \rho_i = 1$, $\sum \kappa_i = 1$, and $\sum \mu_i = 1$ each one of the layers conserves work and switches chaotically. However, any parameter that causes the output of a layer to reflect the chaotic nature of the production will be a coupling parameter between the different layers. We choose a finite production loss due to an unproductive switching time as the coupling parameter. Notice, since the output reduces through the idling of machines, the influx into the next machine will have discrete values. Figure 11 shows a typical sequence: The outflow from the X level is reduced by ρ if one machine idles and by 2ρ if two machines switch. This assumes that the influx into level X is big enough to sustain the nominal outflux from all buffers. If the outflux from Z is too low there might not be enough influx to X and hence additional outflux levels are possible. The chaotic dynamics of the one layer production system will lead to a chaotic sequence of changes in the outflux.

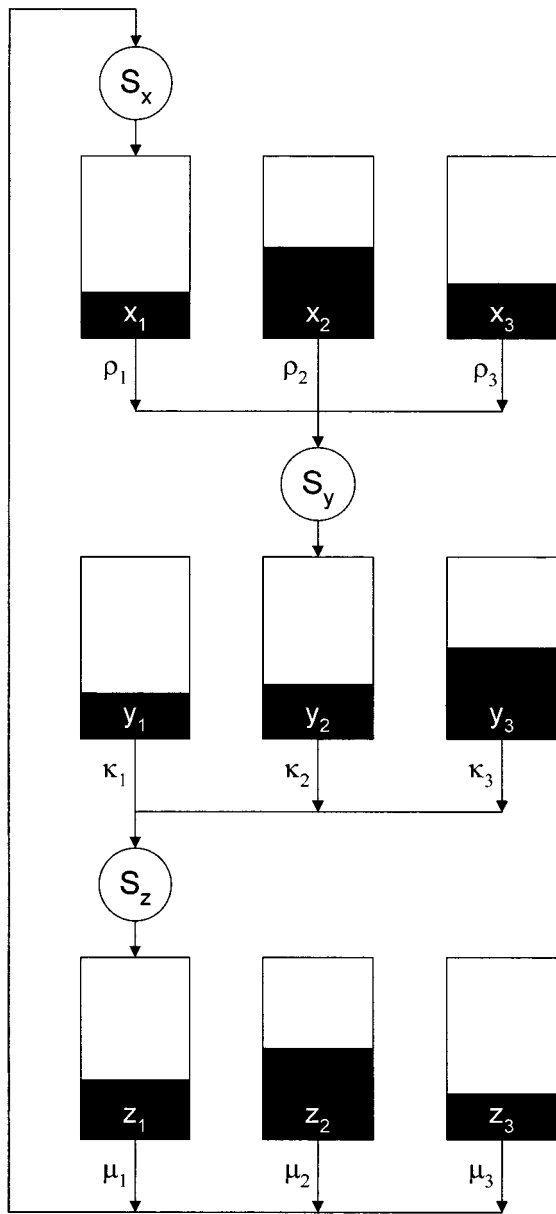


FIG. 10. A network of three layers of three parallel machines with switched arrival protocol.

B. Simulations

Figure 12 shows a simulation of the three-layer system with switching times $\tau_{sX}=0.05$, $\tau_{sY}=0.1$, $\tau_{sZ}=0.15$ for each buffer in the three buffer levels, respectively. We show the total amount of work in each layer. The buffer levels vary chaotically but surprisingly each layer has a well-defined average value of $X_{av}=0.5$, $Y_{av}=1$, $Z_{av}=1.5$ which adds up to three as required for the conservation of work. Based on the synchronization of coupled chaotic oscillators⁷ one could expect some sort of synchronization between the various chaotic levels. However this is not the case. Figure 13 shows the difference between production rates for the X layer minus the production rate for the Y layer. The difference is zero for most of the time when all levels work at full capacity. The difference jumps between a small number of positive and

negative values reflecting the fact of reduced outputs during switching. Clearly, the two chaotic processes are not amplitude synchronized.

On the other hand there are transient processes that clearly indicate coherent behavior between the three chaotic levels: Figure 14 shows the three total work levels as a function of time for very similar switching times, $\tau_{sX}=0.05$, $\tau_{sY}=\tau_{sX}+5\%$, $\tau_{sZ}=\tau_{sX}+10\%$. The dotted lines in Fig. 14 indicate some traveling chaotic waves in this simulation.

C. Self-balancing buffer levels

The coherent chaotic balance that was seen in the simulations in Fig. 12 and the chaotic traveling waves seen in Fig. 14 can be explained through the dependency of the clearing time on the average amount of work in the system. Call W_i the total amount of work in layer i of a total of N layers. There is a balance equation for the work in every level of the form

$$\frac{dW_i}{dt} = o_{i-1} - o_i, \tag{11}$$

where o_i indicates the outflux from layer i to layer $i+1$. We assume that the system evolves on two time scales—a fast one representing individual switches and a longer one, representing the long term transfer of work from one layer to another. We average Eq. (11) to get the evolution of the total work on the long time scale

$$\frac{d\bar{W}_i}{dt} = \bar{o}_{i-1} - \bar{o}_i \tag{12}$$

with a bar indicating averaged quantities. The averaged outflux at a level depends on the number of switches in that level

$$\bar{o}_i = \sum_j \rho_i^j - \sum_j \rho_i^j \tau_i^j \nu_i^j, \tag{13}$$

where ρ_i^j , τ_i^j , ν_i^j stands for the production rate, the switching time, and the average switching frequency of the j th machine in the i th layer, respectively. For simplicity let us assume that all production rates and the switching times are equal for each machine of a certain level. Clearly the average time between switches will depend linearly on both the average amount of work in a buffer and on the switching time. Hence

$$\nu_i = \frac{1}{c\bar{W}_i + d\tau_i}, \tag{14}$$

where c, d are constants. With this, Eq. (12) becomes

$$\frac{d\bar{W}_i}{dt} = N\tau_i \frac{1}{c\bar{W}_i + d\tau_i} - N\tau_{i-1} \frac{1}{c\bar{W}_{i-1} + d\tau_{i-1}}. \tag{15}$$

Equation (15) has an equilibrium when

$$\frac{\bar{W}_i}{\bar{W}_{i-1}} = \frac{\tau_i}{\tau_{i-1}}. \tag{16}$$

With $\sum_i \bar{W}_i = N$ we can therefore calculate the equilibrium levels. For the example of Fig. 12 we calculate $\bar{W}_1=0.5$, $\bar{W}_2=1$, $\bar{W}_3=1.5$ in agreement with the simulation. We can also calculate the stability of that equilibrium. For the case

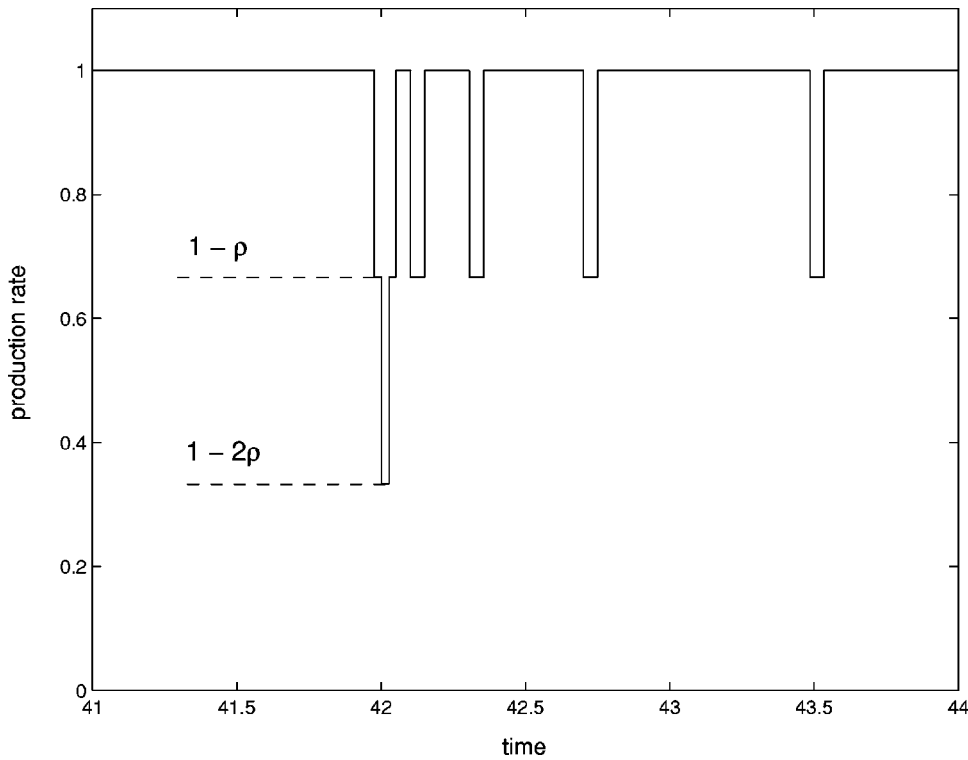


FIG. 11. Production rate of level X as a function of time for equal work rates. Every time a switch is made, the production drops.

$N=3$ we find one zero eigenvalue, corresponding to the fact that the dynamics evolves on the simplex, and two complex conjugates eigenvalues of the form (modulo a constant factor)

$$\lambda_{\pm} = -\frac{1}{2} \sum_i \tau_i \pm \sqrt{\sum_i \tau_i^2 - \sum_i \sum_{j, i \neq j} \tau_i \tau_j}. \quad (17)$$

The eigenvalues are always complex conjugate and the imaginary part is largest if all τ_i are equal. Hence the equilibrium is a spiral sink. We believe that this explains the coherent “chaotic traveling waves” seen in Fig. 14. It is obvious from Eq. (12) that an equilibrium will exist if the processing rates ρ_i^j are not the same, or if the switching time

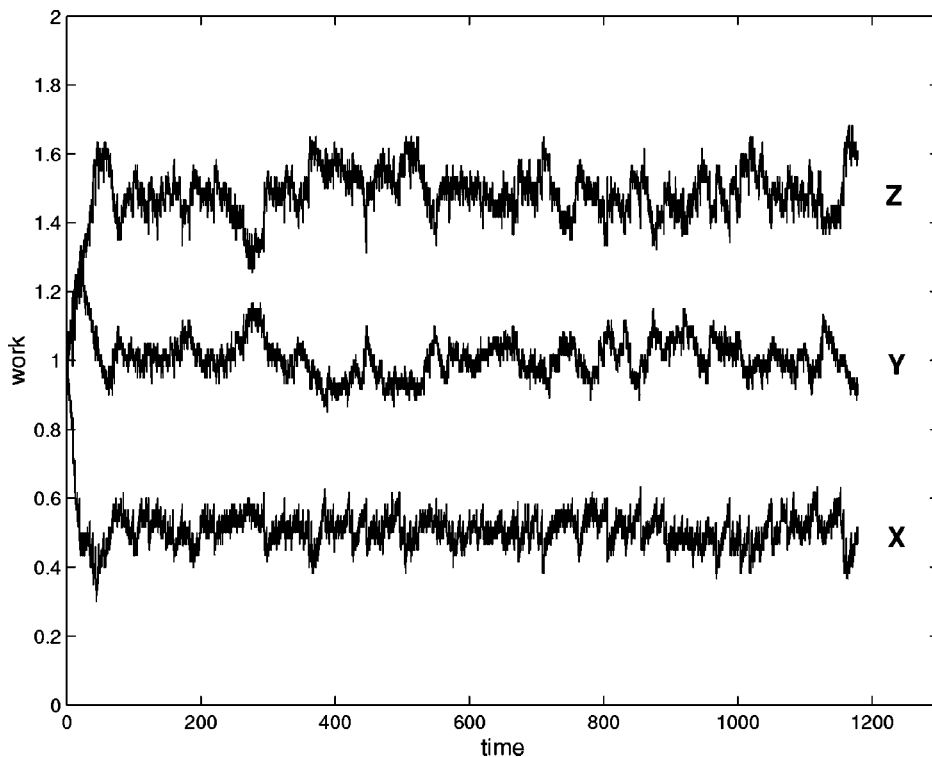


FIG. 12. Work levels as a function of time for the three-layer system. There is strong variation in coupling strengths.

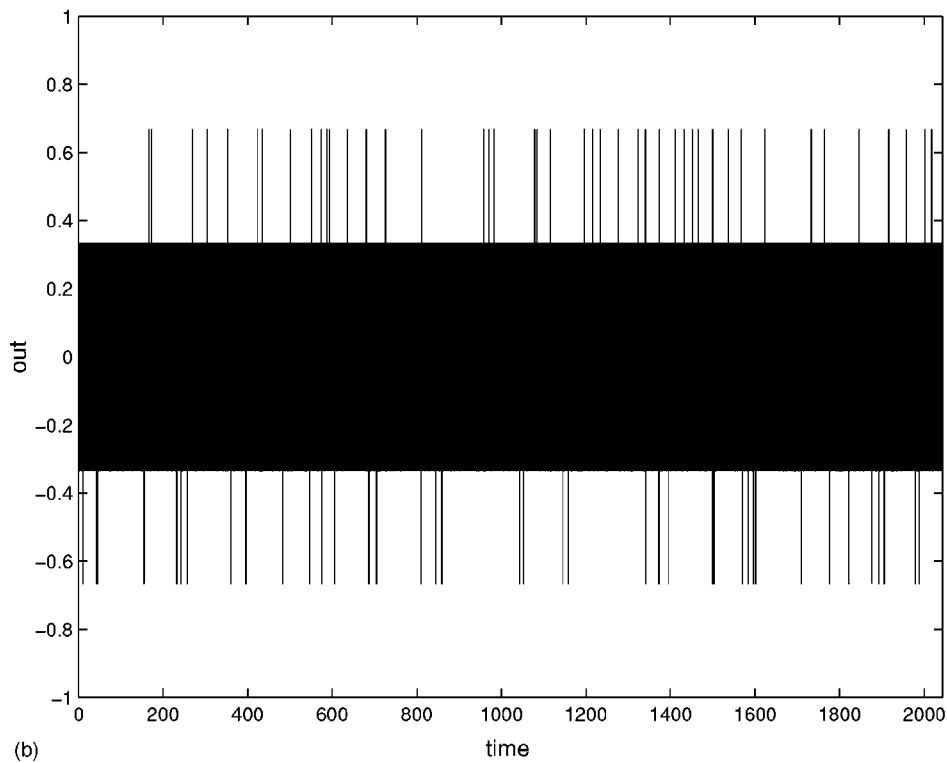
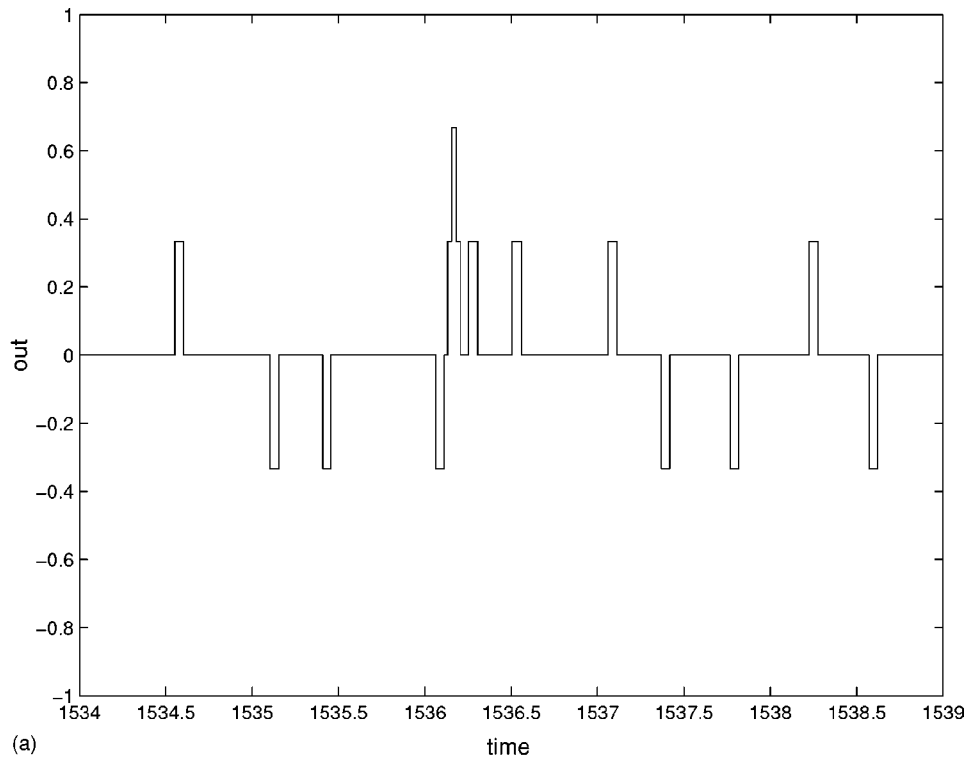


FIG. 13. Production rate X —production rate Y , zoomed in on $[1530, 1535]$, equal work rates and equal switching times $\tau_s = 0.05$ (left), for a longer time interval (right).

becomes unequal in different machines. Similarly, the processing rates within a layer do not have to add up to one. Small imbalances of the production rates between different layers will just lead to different average levels. We have performed simulations that confirm this for small deviation from the symmetric cases. Qualitatively nothing new seems to happen.

IV. CONCLUSION AND FUTURE WORK

We have extended the model of the switched arrival system to include some so far neglected realistic constraints: We have included switching times and maintenance into the model. Both reduce the dynamics of switched arrival system to the dynamics on an edge of a simplex. We have solved the

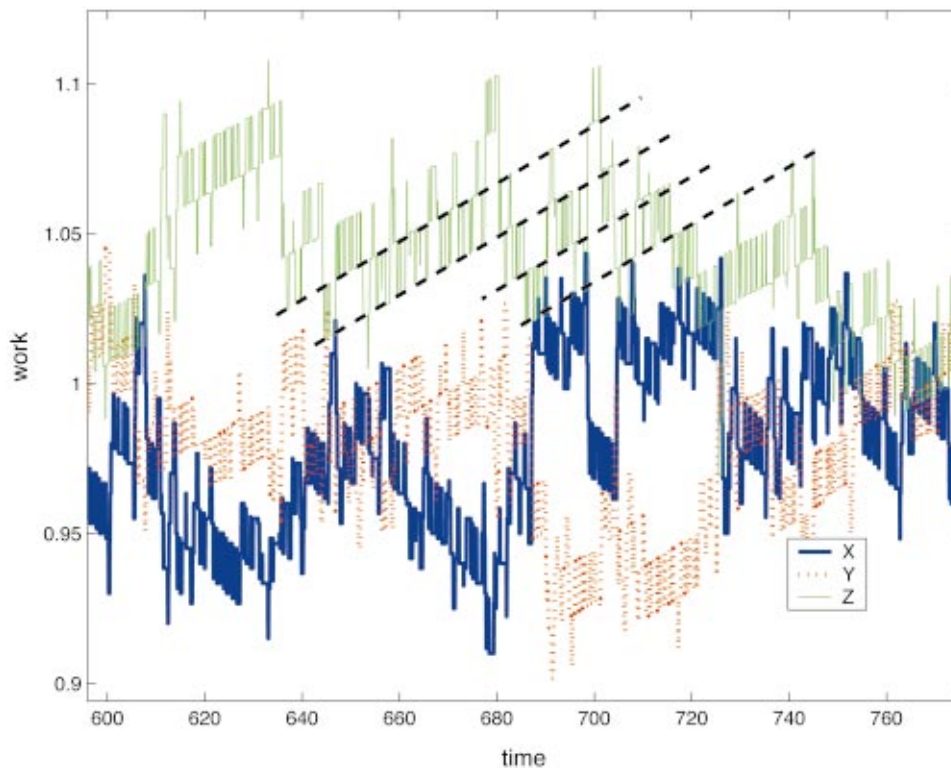


FIG. 14. (Color) Transient behavior near equilibrium for the three-layer system with almost equal switching times. Chaotic traveling waves can be seen.

Frobenius–Peron equation for the probability distribution of the return times which allows us to calculate the statistical quantities for the return times which are important for scheduling maintenance and for the loss of production due to switching. Further extensions to model parallel as well as serial networks of the switched arrival system are discussed. A coupling parameter based on lost production was identified. Simulations of three parallel and three serial levels were performed. They showed no individual synchronization of machines but a global synchronization of the switching frequencies between serial levels. An averaged model was introduced that allowed one to predict the equilibrium states for the average work levels in the serial layers as well as their stability characteristics. It is important to point out that the whole sequential production line globally follows a CONWIP protocol, i.e., the work in progress is kept constant by matching influx and outflux. However, while at every level operating chaotically, it is nowhere *a priori* locally balanced, yet it self-organizes to a chaotically balanced state.

This work is the starting point for extensions in many directions: We are planning

- (1) large-scale simulations for realistic factories, using input from industry (Intel).
- (2) Simulation and analysis of the three parallel by three serial layer model when all production rates and all switching times are not equal. We expect self-sustained chaotic traveling waves between the different levels.
- (3) Control of factory production by controlling a parallel and serial network only at specific control points.

In addition, we think the issue of individual synchronization of machines deserves further study. In hindsight it is clear that by coupling layers with identical machines the coupling gets averaged over a whole layer and information about a specific chaotic process is lost. This might not be the case if all processing rates are different. Hence in that case we might be able to synchronize across different layers.

ACKNOWLEDGMENTS

This work was supported by NSF Grant Nos. DMS 0075041 and DMI 0075655 and a grant from Intel corporation. Stimulating discussions with Karl Kempf (Intel Corporation) are gratefully acknowledged.

¹C. J. Chase, J. Serrano, and P. J. Ramadge, “Periodicity and chaos from switched flow systems: Contrasting examples of discretely controlled continuous systems,” *IEEE Trans. Autom. Control* **38**, 70–83 (1993).

²C. Horn and P. J. Ramadge, “A topological analysis of a family of dynamical systems with non-standard chaotic and periodic behavior,” *Int. J. Control* **67**, 979–996 (1997).

³T. Ushio, H. Ueda, and K. Hirai, “Controlling chaos in a switched arrival system,” *Syst. Cont. Lett.* **26**, 335–339 (1995).

⁴T. Ushio, H. Ueda, and K. Hirai, “Control of chaos in switched arrival systems with N buffers,” *Electron. Commun. Jpn., Part 3* **83**, 81–86 (2000).

⁵I. Katzorke and A. Pikovsky, “Chaos and complexity in simple models of production dynamics,” *Discrete Dyn. Nat. Soc.* **5**, 179–187 (2000).

⁶J. L. McCauley, *Chaos, Dynamics and Fractals* (Cambridge University Press, Cambridge, 1993).

⁷A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization* (Cambridge University Press, Cambridge, 2001).