Mathematical Methods in Operations Research 2005
Special issue in honor of *Arie Hordijk*

Controlling an oscillating Jackson-type network having state-dependent service rates

Arnon Arazi[1], **Eshel Ben-Jacob**[2], **Uri Yechiali**[1]

[1] Department of Statistics & Operations Research, School of Mathematical Sciences, Raymond & Beverly Sackler Faculty of Exact Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel
[2] School of Physics and Astronomy, Raymond & Beverly Sackler Faculty of Exact Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel

Received: January 2005 / Revised version: April 2005

Abstract We consider a Jackson-type network comprised of 2 queues having state-dependent service rates, in which the queue lengths evolve periodically, exhibiting noisy cycles. To reduce this noise a certain heuristic, utilizing regions in the phase space in which the system behaves almost deterministically, is applied. Using this heuristic, we show that in order to decrease the probability of a customers overflow in one of the queues in the network, the server in that same queue - contrary to intuition - should be shut down for a short period of time. Further noise reduction is obtained if the server in the second queue is briefly shut down as well, when certain conditions hold.

Key words Queueing Networks, Oscillations, Fluid Approximation, Control, Dominant Probability.

1 Introduction

The study of queueing networks usually focuses on the stationary distributions of such systems, with some attention given to transient behaviours. There is almost no work dealing with specific time-dependent dynamic behaviours, such as oscillations or chaos (for an example of the few exceptions, see [3]).

In [1], we have presented a 2-queue Jackson-type network with state-dependent service rates, which roughly exhibits an oscillatory behaviour. To produce such a network, the functional form of the model rates and probabilities are chosen so that the fluid approximation of the network is equivalent to a (deterministic) dynamical system featuring a limit cycle in

the *phase space* (the phase space is defined here as the $L_1 - L_2$ plane, where L_i is the number of customers in queue i ($i = 1, 2$), including the customer being served). The extent to which the stochastic queueing network agrees with this approximation depends, among other things, on the workload in the system: the more customers present, the closer are the trajectories produced by the network to that of the fluid approximation, as the change induced by an event involving a single customer (i.e. arrival, service completion or movement between queues) is less significant. Therefore, we focus our discussion on highly-loaded networks.

Our main interest in this paper is in the *control* of the above network. More precisely, we aim at reducing the noise, i.e. the variance of the maximal lengths (amplitudes) obtained for both queues in each cycle, and of the period (cycle duration) of the actual trajectories. To accomplish this, a certain heuristic is employed, arising from the observations delineated below.

In each state of the network, several Poisson processes "compete", each having a different probability of being the first to produce the next event. We use the term *dominant probability* to denote the highest of these probabilities. Since the rates and probabilities defining the network are state-dependent, the value of the dominant probability varies in the phase space. In particular, there are regions in the phase space in which the dominant probability is close to 1. When its trajectories traverse these regions, the system behaves almost deterministically, whereas in the other parts of the phase space, random fluctuations are apparent in its conduct. Throughout this paper, we refer to the former regions using the term *nearly-deterministic*.

We use control strategies which interfere with the network's servers activity, so that the relative "winning" probabilities of the "competing" processes are altered in certain regions of the phase space. The heuristic we suggest argues that such an interference is mostly effective when employed to the system just before it visits the nearly-deterministic portions of the phase space. The reason for this is that changes made in the system in the midst of the noisier regions are lost shortly after being performed; and, in the nearly-deterministic regions themselves, one process tends to dominate, regardless of any interference. In contrast, changes made just before the entry to a nearly-deterministic region can have immediate and significant implications, as they determine the conduct of the system almost completely, until the system leaves this region.

The effectiveness of this heuristic is examined through numerical simulations. While the use of the control strategies is limited to a relatively small region of the phase space, a considerable overall noise-reduction is obtained. More specifically, we show that:

1. In order to decrease the probability that the length of the first queue exceeds a certain value, the server in this *first* queue - contrary to intuition - should be shut down whenever the length of the second queue reaches some threshold;
2. By employing a strategy in which *both* servers are shut down for certain periods of time, depending on the length of the second queue, the

variance of the maximal lengths reached by the queues, as well as the variance of the period (i.e., cycle duration), are reduced considerably.

The organization of this paper is as follows. In section 2 we present the 2-node Jackson-type network mentioned above and demonstrate its periodic time evolution. The manner in which this network was constructed is explained in section 3, as part of a more comprehensive discussion regarding the integration of specific desired dynamic behaviours into generalized queueing networks. The control strategies are applied to the 2-node network in section 4, and results confirming their effectiveness are presented.

2 An oscillating Jackson-type network

The network is defined as follows. Let γ_i denote the external arrival rate to queue i ($i = 1, 2$), and let μ_i be the service rate at this queue. The probability that a customer, upon completing his service at queue i, will move to queue j, is denoted by p_{ij}; the probability that such a customer will leave the system, is denoted by $d_i = 1 - p_{ij}, j \neq i$ (we set $p_{ii} = 0, i = 1, 2$). Let L_i denote the number of customers in queue i (including the served customer).

Specifically, with A and B positive constants satisfying $B > 1 + A^2$, set:

$$\begin{array}{llll} \gamma_1 = A & \mu_1 = (B+1)L_1 & p_{1,2} = \frac{B}{B+1} & d_1 = \frac{1}{B+1} \\ \gamma_2 = 0 & \mu_2 = L_1^2 L_2 & p_{2,1} = 1 & d_2 = 0 \end{array}$$

This network is depicted in Figure 1. The results of actual simulation runs of the network appear in Figure 2. As can be seen, the lengths of the 2 queues oscillate periodically, in a coordinated manner, with some apparent noise (Figures 2a and 2c). This leads to a motion of the system roughly along a cycle in the $L_1 - L_2$ plane (Figure 2e). For example, starting in a state where the first queue is crowded while the second queue is nearly empty, the network will witness the graduate passage of customers from the first queue to the second, reaching a state where most of them wait in the second queue; this process is then reversed, as the system returns to its starting point.

The somewhat irregular form of the service rates requires interpretation. The linear service rate in the first queue suggests an infinite number of servers in that queue, where each server provides a service lasting an exponential time, with rate $B + 1$. More peculiar is the service rate in the second queue, which increases with the length of the first queue. To explain this oddity, we suggest the notion of *customers acting as servers*: customers waiting in the first queue participate, in the meantime, in the service of customers standing in the second queue, enhancing the rate of service there. Furthermore, this enhancement of service results from connections, or a cooperation, forming between pairs of customers waiting in the first queue;

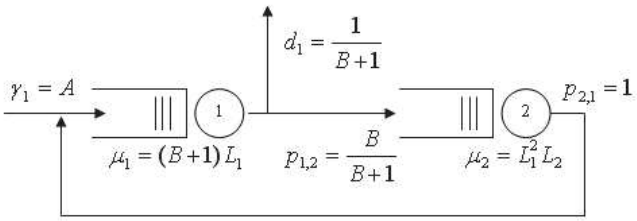

Fig. 1 A Jackson-type queueing network with state-dependent service rates, exhibiting noisy oscillations. Customers completing service in queue 2, return to queue 1.

hence, the service rate in the second queue increases with L_1^2, rather than L_1. Finally, the linear dependence on L_2 of this rate implies the presence of an infinite number of servers in the second facility, utilizing the customers waiting in the first queue.

Scenarios in which waiting customers supply some service in the meantime may actually be quite natural in networks composed of nodes both receiving and providing service. In computer networks this situation is rather common. Following is an example illustrating this point.

Consider a hypothetical computer network, capable of handling jobs of 2 types: Type-I jobs, which have a high priority, are composed of a sequence of steps, each being either a calculation of some sort, or an access to a database (DB). Type-II jobs, which have a lower priority, require some pre-processing, followed by a series of independent calculation steps, which can be performed in parallel. The network is assumed to consist of 3 types of nodes: (a) A group of computers capable of performing efficient calculations (type-A nodes); (b) A DB server (type-B node); and (c) A group of computers which can handle the pre-processing required by the type-II jobs (type-C nodes). A type-I job entering the network arrives directly to a node of type A, which from that point on is responsible for the complete execution of the job, including sending requests to the DB server, if these are required. A type-II job entering the network arrives to a node of type C, which pre-processes it, and then manages its calculation in parallel by several A-nodes. The queues in front of the A-nodes are managed according to priority. Furthermore, the steps involving a DB access are considerably lengthier than the steps involving calculations. It is assumed that the jobs of type I arrive to the system in a high rate, such that the idle time of nodes of type A is negligible.

In such a setting, type-II jobs are likely to "starve" while waiting to be served in nodes of type A. A reasonable solution to this problem will be to allow the type-II jobs to be processed by A-nodes waiting for a response from the DB server. In this case, the effective service rate of type-II jobs (and hence the effective service rate of each type-C node) will be linear in the number of A-nodes queueing in front of the DB server. That is, this

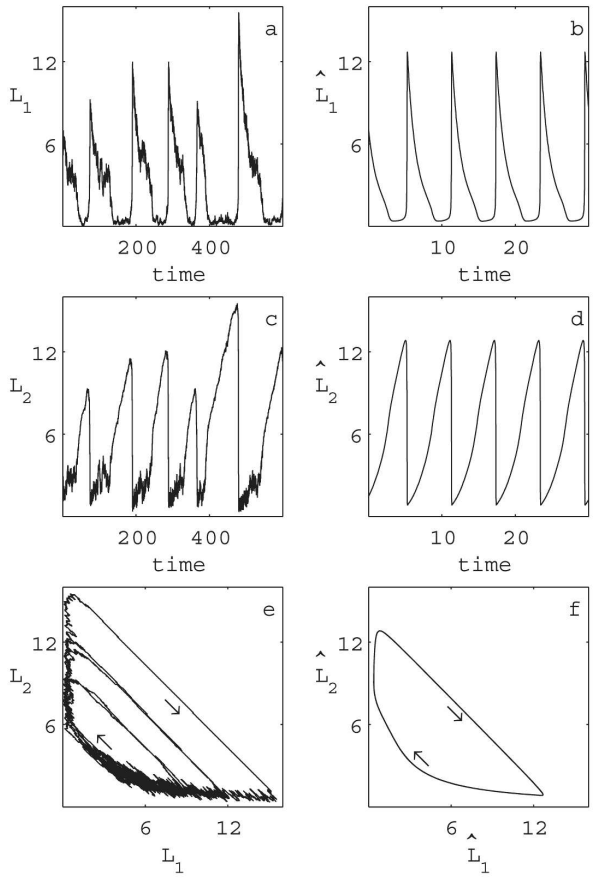

Fig. 2 The behaviour of the 2-node Jackson network defined in the text (left) alongside the behaviour of its fluid approximation (right, defined in section 3), as generated by numerical simulations. Figures a) and b) depict the time evolution of the length of the first queue. Figures c) and d) show the time evolution of the length of the second queue. Figures e) and f) portray the phase space trajectories of the systems. The occurrence of cycles in the phase space is apparent in both cases. In the trajectories of the stochastic system, noisy regions, as well as smoother ones, are discernable. Note, in addition, the existence of parallel paths in these trajectories, on which the length of the second queue decreases and that of the first queue increases (see section 4). In this and the following figures, unless stated otherwise, $A = 3; B = 10.5; \Delta = 0.05$. A video demonstration of the cyclic time-dependent behaviour of both systems is available at http://www.math.tau.ac.il/∼uriy/Publications.html.

example shows a situation where the service rate in one server depends on the length of a queue in front of another server, due to service being supplied by waiting customers.

3 Introducing specific dynamic behaviours into queueing networks

In this section we explain how the functional form of the probabilities and rates of the network presented in section 2 were chosen. We open, however, with a general discussion regarding the introduction of specific dynamic behaviours into queueing networks.

We aim at producing a queueing network in which the time-dependent queue lengths roughly exhibit a certain dynamic behaviour. Such a behaviour may be more precisely defined as a solution of a specific set of deterministic differential equations. The goal, therefore, is to present a queueing network whose time-dependent evolution imitates, approximately, this (deterministic) solution.

One possible way to obtain such a queueing network is to allow the arrival and service rates defining it, as well as the probabilities which govern the movement of customers between queues, to be state-dependent. One should then inspect the fluid approximation of such a queueing network, expressed as a set of deterministic differential equations, where the variables approximate the average queue lengths. By comparing the approximation equations to those of the deterministic dynamic system whose behaviour one wishes to imitate, it may be possible to map between the terms appearing in the two sets of equations. Such a mapping points to the proper selection of the rates and probabilities defining the network, leading to a stochastic queueing network whose average behaviour is approximated by the imitated deterministic system. The quality of this approximation depends, among other things, on the workload present in the queueing network, as indicated in the introduction.

We'll next derive the fluid approximation of a generalized queueing network (a G-network), where stochastic events other than mere customer arrival or service completion are possible (see [4] for a pioneering work, and [2] for a review of works dealing with such networks). For the sake of simplicity, the discussion here is limited to the case where each stochastic event changes the length of a queue by no more than a single customer (a single event may affect the lengths of several queues, though). In addition, only the case where all queues are not empty is considered. Since the fluid approximation serves only as a guide to the choice of rates and probabilities, and since we focus our discussion on highly-loaded networks, the inspection of this case only is sufficient for our purposes.

Consider, then, a network of queues, where customers arrive to service facilities, leave them or move between them due to some stochastic events, generated by Poisson processes. Let us inspect the j-th queue. Denote the rates of the events increasing the length of the queue by $\{r_{j1}(\mathbf{L}),,r_{jR}(\mathbf{L})\}$, and of those decreasing the queue by $\{q_{j1}(\mathbf{L}),,q_{jQ}(\mathbf{L})\}$, where $\mathbf{L}(t) = (L_1(t),,L_n(t))$ is the time-dependent vector of queue lengths (including the customers being served), and R and Q are the numbers of processes producing the events of each type. The rates are considered to be some functions of

the state of the system. Note that we refer here to *effective* rates which are possibly the product of a rate and a probability (for example, the product of a service rate in one queue and the probability to move from this queue to another, representing the effective transition rate from the former queue to the latter, given that the former queue is not empty).

Since all the inspected events result from Poisson processes, the probability of an arrival of a customer to queue j during an infinitesimally short interval of time, $(t, t+h]$ is given by

$$P_j^+(\mathbf{L}) = \sum_{i=1}^{R} r_{ji}(\mathbf{L})h + o(h) \tag{1}$$

and the probability of a departure of a customer from queue j is equal to

$$P_j^-(\mathbf{L}) = \sum_{i=1}^{Q} q_{ji}(\mathbf{L})h + o(h) \tag{2}$$

The length of the queue at time $t+h$, given its length at time t, is

$$L_j(t+h) = L_j(t) + \xi_j(\mathbf{L}) \tag{3}$$

where ξ_j is the random variable designating the change in the queue's length due to an event occurring in the interval $(t, t+h]$. ξ_j can be either 0, -1 or 1, with probabilities depending on $P_j^+(\mathbf{L})$ and $P_j^-(\mathbf{L})$; hence, ξ_j depends indeed on the lengths of the queues in the network.

We are interested in estimating the average behaviour of this system. To this end, the fluid approximation $\hat{\mathbf{L}} = \hat{\mathbf{L}}(t)$ is presented. The j-th element of this vector, which approximates the time-dependent evolution of the average length of queue j, satisfies the equation

$$\begin{aligned} \hat{L}_j(t+h) \;=\; & \hat{L}_j(t) + 1 \cdot P_j^+(\mathbf{L}) + (-1) \cdot P_j^-(\mathbf{L}) = \\ & \hat{L}_j(t) + h[\sum_{i=1}^{R} r_{ji}(\hat{\mathbf{L}}) - \sum_{i=1}^{Q} q_{ji}(\hat{\mathbf{L}})] + o(h) \end{aligned} \tag{4}$$

Rearranging equation (4), dividing by h and taking the limit as $h \to 0$, leads to the following differential equation:

$$\frac{d\hat{L}_j}{dt} = \sum_{i=1}^{R} r_{ji}(\hat{\mathbf{L}}) - \sum_{i=1}^{Q} q_{ji}(\hat{\mathbf{L}}) \tag{5}$$

We now wish to express the fact that the network is highly-loaded. To do so, we alter the size of a change induced by a single stochastic event. Instead of assuming that each arrival increments the queue length by 1, and each departure decrements it by 1, these changes are set to be $\pm\Delta$, respectively, where $\Delta \ll 1$. The smaller Δ is, the less significant is the change induced on the system by an event involving a single customer. Thus, small values of Δ imply a crowded queueing network. By varying the size of Δ, it is

now possible to study the approximated average behaviour of the queueing system at different orders of magnitude of workloads. The introduction of Δ leads to the following equation:

$$\frac{d\hat{L}_j}{dt} = \Delta[\sum_{i=1}^{R} r_{ji}(\hat{\mathbf{L}}) - \sum_{i=1}^{Q} q_{ji}(\hat{\mathbf{L}})] \tag{6}$$

The multiplication by Δ implies a mere scaling of the time axis. The approximated average behaviour itself does not qualitatively change - only the time it takes for its manifestation changes. However, the divergence of the actual behaviour from this approximated average *does* depend on the size of Δ. Note also that by introducing Δ, $\hat{L}_j(t)$ ceases to approximate the average number of customers in queue j, but rather this number multiplied by $1/\Delta$.

As stated above, the fluid approximation equations (6) can now be compared to those of the dynamic system whose behaviour one wishes to imitate. This comparison may point, hopefully, to a choice of the rates and probabilities which will result in equivalent sets of differential equations.

Let us now apply this general procedure to produce the 2-node Jackson-type queueing network presented in section 2. Following the above considerations, we obtain the fluid approximation of this network, given by the equations

$$\begin{aligned} \frac{d\hat{L}_1}{dt} &= \Delta[(\gamma_1 + \mu_2 p_{2,1}) - \mu_1] \\ \frac{d\hat{L}_2}{dt} &= \Delta[(\gamma_2 + \mu_1 p_{1,2}) - \mu_2] \end{aligned} \tag{7}$$

where γ_i, μ_i and p_{ij} $(i, j = 1, 2,\ i \neq j)$ are generally allowed to be some functions of $\hat{\mathbf{L}}(t)$.

The dynamical system we wish to imitate here is known as the Brusselator model, studied in chemistry (see ref. [5]):

$$\begin{aligned} \frac{dX}{dt} &= A - (B+1)X + X^2Y \\ \frac{dY}{dt} &= BX - X^2Y \end{aligned} \tag{8}$$

This system describes the change in the concentrations of two types of hypothetical molecules, X and Y. A and B are constants. Under the condition $B > 1 + A^2$, it is possible to show that this system has a stable *limit cycle* in the phase space.

Let us associate X and Y with the fluid approximation of the first and the second queue lengths, respectively. Comparing the fluid approximation equations (7), and those of the specific dynamic system (8), it's easy to see that setting the rates and probabilities to the ones stated in section 2 results in identical sets of equations (up to a multiplication by Δ).

The behaviour of the resulting fluid approximation is depicted in Figures 2b, 2d and 2f.

4 Controlling the network

We now turn to discuss the control of the presented network. In particular, our aim is to decrease the variance of the maximal lengths reached by the two queues in each cycle (i.e., the amplitudes), as well as that of the time required for the completion of each cycle (that is, the period); this time is measured between consecutive crossings of an arbitrary line in the phase space.

Inspecting the trajectories produced by the network more closely, we make two observations (see also Figure 2e):

1. There are regions in the phase space where the trajectories fluctuate more, and regions where they exhibit a smoother conduct. More precisely, the system appears to be less noisy during the descent of the second queue's length and the concurrent ascent of the first queue's length, and noisier in the other parts of its trajectories;
2. As the length of the second queue grows larger, the system will, at one point, "break to the right", and start the descent of the second queue's length and the ascent of the first queue's length. The exact position of this turning point varies from cycle to cycle; this leads to the parallel paths appearing in the figure.

The reason for the existence of smooth and noisy regions is as follows: As the system evolves, in each moment several stochastic processes "compete" - which of them will be the first to produce the next event: the arrival of a customer to the first queue, the service completion of a customer in the second queue and his passage to the first queue, etc. At certain regions of the phase space, this "competition" is close; that is, the probabilities of some of these processes to "win" are about the same. In these regions, the "winning" process will alternate frequently, and fluctuations will be observable in the phase space. On the other hand, in some regions the probability of one of these processes to "win" is dominantly large; this process will, more often than not, "win". In such areas, the system will behave smoothly, advancing consistently in the phase space.

Figure 3 confirms this postulation, depicting the contours of the dominant probability across the relevant region of the phase space. The correlation between the values of the dominant probability, and the occurrence of fluctuations in the trajectories exhibited by the system, is apparent (compare with Figure 2e).

These observations lead to the notion of *influential regions* - areas in the phase space where the outcome of the random choices have a relatively large influence on the (short-term) future behaviour of the system. Consider, for example, the region in the phase space where the length of the second queue increases. As we have seen, at some point in this region, the trajectory will "break to the right", and the length of this queue will start its descent, while the length of the first queue will increase. It is apparent from the figures above, as well as from the fact that the system's behaviour there

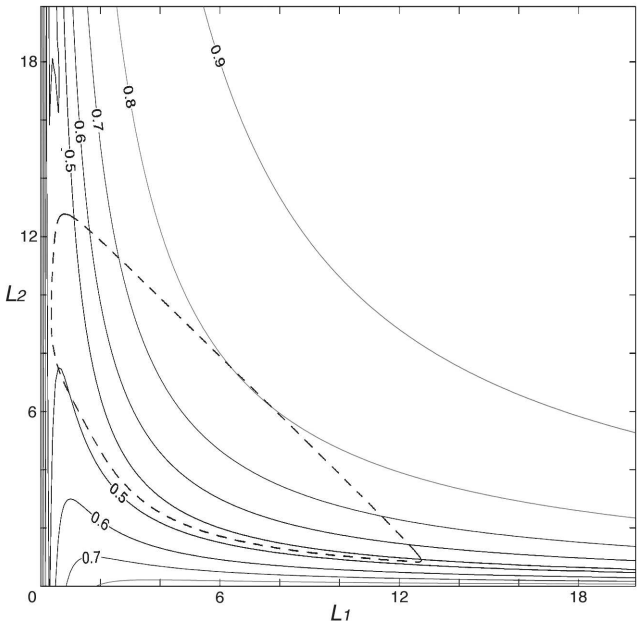

Fig. 3 The dominant probability contours, in the relevant region of the phase space. The limit cycle exhibited by the fluid approximation is drawn in a dashed line. During the decrease in the length of the second queue and the concurrent increase in the length of the first queue, the system traverses through a region of relatively high dominant probability values; the rest of the trajectory passes through a "valley" of these values.

is discernibly smoother, that the latter region is governed by a dominant stochastic process (that of service completion in the second queue, followed by a passage to the first queue). That is, once the system breaks out of the noisy area, it enters what we term a "nearly-deterministic" phase, which lasts until the second queue is almost emptied; this is also the point where the first queue reaches its maximal length. Thus, the "choice" made in the noisy region at one end of the phase space - the exact position of the point in which the second queue starts its decrease - *determines almost completely* the maximal length of the first queue, at the other end of the phase space. Note that, since the system then enters a noisy region again and returns to a region common to all cycles (low L_1 values, low L_2 values), the effect of this "choice" is lost shortly after. This is the reason why we've stated above that influential regions affect only the short-term future behaviour of the system - the duration of the current cycle only.

It stands to reason that influential regions may be quite useful in the task of controlling the system, that is, decreasing the variability it exhibits. In the example outlined here, controlling the position of the turning point

of the system necessarily results in controlling the maximal length of the first queue. This will now be demonstrated.

We start with the task of decreasing the probability that the length of the first queue exceeds some value. This goal can be considered quite reasonable in queueing theory terms: if the capacity of a queue is limited, it is often desirable to decrease the probability of losing arriving customers, who cannot join the system while the queue is full. As can be seen in Figure 4, the distribution of the maximal length reached by the first queue in each cycle (estimated from the results of numerical simulations) has a relatively heavy right tail. That is, the ability to predict with high precision the maximal length reached by the first queue in each cycle is limited.

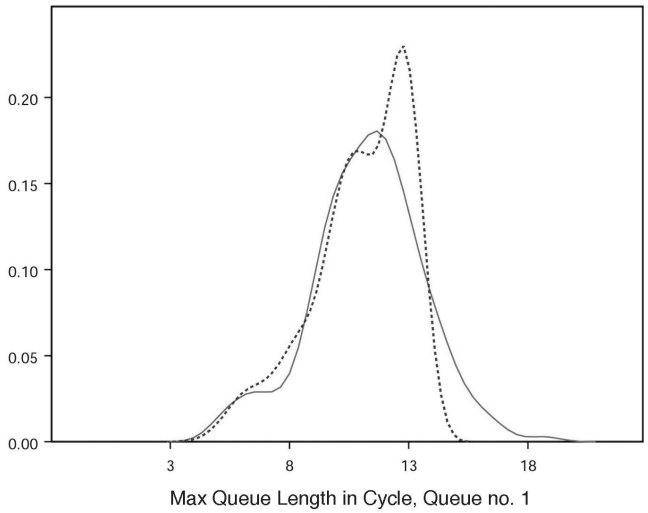

Fig. 4 The probability density function of the maximal length reached by the first queue in each cycle. The solid line refers to the case where no control strategy was applied; the dashed line refers to the case where the control strategy bounds the maximal length of the second queue.

From what was stated above, it's clear that in order to limit the length of the first queue, it is sufficient to limit the length the second queue reaches; the larger the latter is, the larger the former will be. Note that the second queue has no inflow of customers from outside the system; all the customers reaching it arrive from the first queue, after they were served there. Therefore, one way to limit the length of the second queue, is to shut down the server in the *first* queue, when the length of second queue reaches some threshold, and to turn it on again when the second queue's length falls beneath this threshold.

This control strategy was tested, with an arbitrary threshold. As we can see in Figure 4, the right tail of the distribution of the first queue's maximal

length was cut down considerably as a result of applying this strategy, with a sharp drop of the distribution function at high values. Thus, it turns out that paradoxically, in order to effectively bound the length the first queue reaches, one should *shut down*, for a short period of time, the server in the *very same* queue.

So far, we have succeeded in limiting the maximal length of the first queue. However, the variance of this size is still quite high (see Table 1), since there are cycles in which the peak of the first queue's length is relatively small, due to a "premature" appearance of the turning point discussed above. For the same reason, the variance of the period is still high. One can think of applications where it may be desirable to lower this variability, increasing the predictability of the system.

One strategy to obtain this goal, is to impose a *lower* bound on the maximal length of the second queue, in addition to the upper bound set earlier. Thus, we'll prevent the system from "breaking to the right" too soon. Note that this "break" actually means switching from a state where mainly the first service facility feeds the second one, to a state where the opposite is the common event, due to the work performed by the server in the second facility. Therefore, in order to avoid the a premature turning point, we can shut down the server in the *second* queue when the length of this queue starts climbing, and turn it on again only when the second queue's length surpasses a certain threshold.

This 2-fold strategy was applied and tested using numerical simulations. The results, suggesting a major drop in the variance of both the maximal

Control Strategy	1^{st} Queue Amplitude Variation Coefficient	Period Variation Coefficient
None	0.21	0.30
Shut Server 1 when 2^{nd} Queue's Length Exceeds Threshold	0.18	0.29
In Addition, Shut Server 2 When 2^{nd} Queue's Length Is Increasing, and in Predefined Range	0.05	0.16

Table 1 The effect of the suggested control strategies on the variance of the maximal length exhibited by the first queue, and on the variance of the period time. Although the first strategy decreases effectively the probability that the length of the first queue exceeds certain values, it has only a minor effect on the resulting variances. On the other hand, the extended strategy induces a considerable decrease in both variances.

length of the first queue and the period time, are summarized in Table 1, and shown in Figures 5 and 6.

We note that the effectiveness of the control strategies employed here relies on the existence of a nearly-deterministic region in the vicinity of the trajectories exhibited by the system. To see that this claim is indeed true, we examine an identical queueing network which *does not* have such a region. To accomplish this, we simply alter the parameters of the model (i.e., A and B). As a result, the system revolves in a different region of the phase space, exhibiting shorter queues. In these trajectories, the value of the dominant probability remains relatively low (below 0.4 all along the fluid approximation trajectory).

Again, we measure the variation coefficient of the maximal length of the first queue. The results are reported in Table 2. Indeed, when a nearly-deterministic region does not exist, the proposed strategies are rendered considerably less effective.

To summarize, we see that by manipulating the system in a relatively small region of the phase space, we've obtained a considerable overall noise-reduction in the exhibited cycles. The ability to do so effectively stemmed from the following 2 main reasons:

1. The existence of a nearly-deterministic region in the phase space, in the vicinity of the trajectories exhibited by the system;
2. The fact that small perturbations in the regions preceding the nearly-deterministic region induce considerable changes at the exit point from this region.

Note that we are not aware of previous usages of similar considerations in the control of queueing networks.

Max Dominant Prob. On Trajectory	Control Strategy	1^{st} Queue Amplitude Variation Coefficient
0.80	None	0.21
	Extended Strategy	0.05
0.38	None	0.26
	Extended Strategy	0.18

Table 2 The variance reduction obtained by means of employing the extended control strategy, both in the case where a nearly-deterministic region exists (first two lines, matching the case $A = 3$, $B = 10.5$), and in the case it does not (next two lines, resulting from setting $A = 1$, $B = 2.05$). Only in the first case a substantial reduction in the variance is observable.

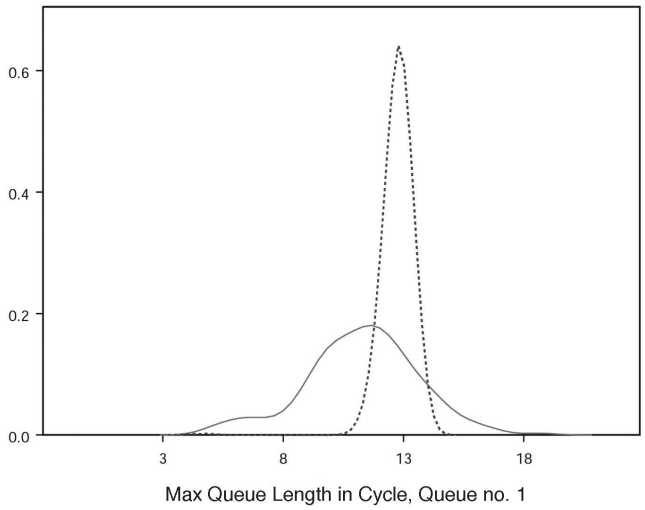

Fig. 5 The decrease in the variance of the maximal length of the first queue, in each cycle. The solid line depicts the probability density function of this variable, when no control strategy is employed. The dashed line describes the case where the extended control strategy is used.

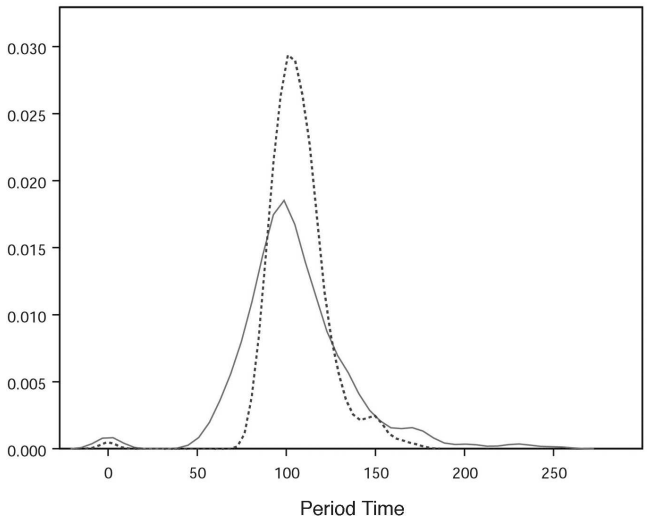

Fig. 6 The decrease in the variance of the period. The solid line depicts the probability density function of this variable, when no control strategy is employed. The dashed line describes the case where the extended control strategy is used. Note the existence of extremely short cycles (the occurrence of negative values of period duration is due to the use of a smoothing procedure in the preparation of the plot).

5 Conclusion

The control of an oscillating Jackson-type network with state-dependent service rates was investigated. A heuristic aiming at the identification of the most "influential" regions in the phase space was applied. These regions precede the stages in the evolution of the system where one stochastic process is dominant.

Using this heuristic, we were able to demonstrate that in order to decrease the probability of customers overflow in a certain queue, it is sometimes desirable, paradoxically enough, to shut down the server in that same queue. Furthermore, a control strategy consisting of the shutdown of both servers in the network for short periods of time can lead to a considerable stabilization of the system's behaviour, reducing the variance occurring in the generated cycles.

Acknowledgements We thank Zeev Schuss for fruitful discussions contributing to the ideas appearing in this work.

References

1. Arazi A, Ben-Jacob E, Yechiali U (2004) Bridging genetic networks and queueing theory. Physica A 332: 585-616
2. Artalejo JR (2000) G-networks: a versatile approach for work removal in queueing networks. Eur. J. Oper. Res. 126 (2): 233-249
3. Chase C, Serrano J, Ramadge PJ (1993) Periodicity and chaos from switched flow systems: contrasting examples of discretely controlled continuous systems. IEEE T. Automat. Contr. 38 (1): 70-83
4. Gelenbe E (1991) Product-form queueing networks with negative and positive customers. J. Appl. Probab. 28: 656-663
5. Thompson JMT, Stewart HB (1986) Nonlinear dynamics and chaos: geometrical methods for engineers and scientists. John Wiley & Sons, New York