Control theoretic modelling and design of admission control mechanisms for server systems

Kihl, Maria; Robertsson, Anders; Wittenmark, Björn

Published in:
NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications / Lecture Notes in Computer Science

2004

Link to publication

Citation for published version (APA):
Kihl, M., Robertsson, A., & Wittenmark, B. (2004). Control theoretic modelling and design of admission control mechanisms for server systems. In *NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications / Lecture Notes in Computer Science* (Vol. 3042, pp. 1366-1371). Springer. http://springerlink.metapress.com/content/9n1vv1jy0k1pf65u/fulltext.pdf

Total number of authors:
3

Control theoretic modelling and design of admission control mechanisms for server systems

Maria Kihl[a], Anders Robertsson[b], and Björn Wittenmark[b]
[a]Department of Communication Systems, [b]Department of Automatic Control
Lund University, BOX 118, 221 00 Lund, Sweden
email: maria@telecom.lth.se, fax: +46 46 14 58 23, tel.no: +46 46 222 9010

Abstract

The admission control mechanism is an important part of many communication systems. In this paper we investigate load control mechanisms for server systems, that is systems that may be modelled as queueing systems. We show how control theory can be used when designing controllers for a *G/G/1*-system, in this case a PI-controller and an RST-controller. These controllers, both commonly used in automatic control are compared with one static controller and one step controller, both commonly used in telecommunication systems. Also, the paper contains a nonlinear stability analysis of the system in which a stability region is determined where the PI-controller based on linear design is shown to also guarantee stability for the nonlinear queuing system.

Methods keywords-- System design, Queueing theory, Control theory, Admission Control.

1. Introduction

Modern communication networks, for example the PSTN, GSM, UMTS, or the Internet, consist of two types of nodes: switching nodes and service control nodes. The switching nodes enable the transmission of data across the network, whereas the service control nodes contain the service logic and control. All service control nodes have basically the same structure as any classical Stored Program Control (SPC) system [14]. The node consists of a server system with one or more servers processing incoming calls at a certain rate. Each server has a waiting queue where calls are queued while waiting for service. Therefore, a service control node may be modelled as a queueing system including a number of servers with finite or infinite queues.

One problem with all service control nodes is that they are sensitive to overload. The systems may become overloaded during temporary traffic peaks when more calls arrive than the system is designed for. Since overload usually occurs rather seldom, it is not economical to overprovision the systems for these traffic peaks, instead admission control mechanisms are implemented in the nodes. The mechanism can either be static or dynamic. A static mechanism admits a predefined rate of calls whereas a dynamic mechanism contains a controller that, with periodic time intervals, calculates a new admission rate depending on some control objective.

The controller in a dynamic control mechanism bases its decision from measurements of a so-called control variable. The control objective is usually that the value of the control variable should be kept at a reference value. Traditionally, *server utilization* or *queue lengths* have been the variables mostly used in admission control schemes. Other solutions are to use the processing delay in the system or, if possible, the users’ response times.

The research concerning admission control has shown that the problem of optimally controlling the arrivals at a queueing system is a difficult task. The main problem comes from the fact that queueing systems usually are analyzed with queueing theory. However, there are no queueing theoretic methods that can be used when developing and designing controllers for the systems. Another solution is, therefore, to use control theory.

Control theory has since long been used to analyze different types of automatic control systems. Also, it contains a number of mathematical tools that may be used to analyze both the stability of a controlled system and to find good control schemes with respect to performance. One well-known controller in automatic control is the PID-controller, which enables a stable control for many types of system (see, for example, [26]). The PID-controller uses three actions: one proportional, one integrating, and one derivative.

There are numerous early papers about admission control of SPC systems. An overview of this research field is given in [15]. One classical controller is the step-controller [10][21][25]. The objective of the control law is to keep the value of the control variable between an upper and a lower level. If the value of the variable is higher than the upper level, the admittance rate is decreased linearly. If the value is below the lower level, the admittance rate is increased.

Recent research on admission control mechanisms for server systems has mainly been focussed on web servers. In [6] a queue length control with priorities was developed. By optimizing a reward function, a static control was found in [8]. An on-off load control mechanism regulating the admittance of client sessions was developed in [9]. In [24] a control mechanism was proposed that combines a load control for the CPU with a queue length control for the network interface.

Very few papers have investigated admission control mechanisms for server systems with control theoretic methods. In [20] a queue length control was developed for an M/M/1-system using stochastic control theory. In [1] and [2] a web server was modelled as a static gain to find controller parameters for a PI-controller. A scheduling algorithm for an Apache web server was designed using system identification methods and linear control theory in [17]. In [7] a PI-controller is used in an admission control mechanism for a web server. However, no analysis is presented on how to design the controller

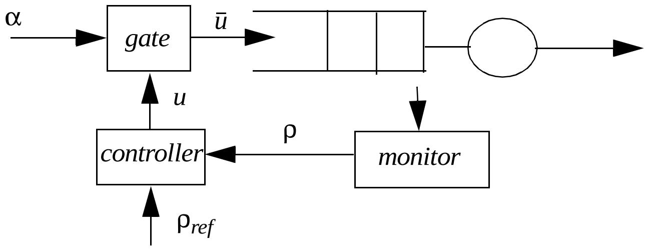

Fig. 1. Investigated system.

parameters. Some papers have investigated PID-controllers in ATM flow control, see for example [13].

In [11] and [18], we analyzed queue length controllers for M/G/1-system. We developed a nonlinear fluid flow model, based on the model in [3], and used this model for designing a PI-controller for the system. We demonstrated that linear models of this system are insufficient, since the nonlinearities in the gate and queue introduce system dynamics that must be considered in the design process.

In this paper we instead analyze load control mechanisms. In [12], we developed and validated a control theoretic model of a G/G/1-system that can be used for the design of load control mechanisms. In [4] we show that the model is valid for an Apache web server. In this paper, we design controller parameters for a PI-controller and a so-called RST-controller. We compare the behavior of the controllers with one static controller and one step-controller. We show that both the PI-controller and the RST-controller have better overall behavior than the other controllers. Also, we perform a nonlinear stability analysis and determine a stability region for the PI-controller. Finally, the paper contains a discussion about the limitations with both linear control theoretic models of queueing systems and linear design methods.

2. System model

The system model is shown in Fig. 1. We assume that the system may be modelled as a G/G/1-system with an admission control mechanism. The admission control mechanism consists of three parts:

a *gate*, a *controller*, and a *monitor*. Continuous control is not possible in computer systems. Instead, time is divided into control intervals of length *h* seconds. Time interval [*kh-h*, *kh*] is denoted interval *kh*.

The monitor measures the *control variable*, in this case the average server utilization during interval *kh*, ρ(*kh*). At the end of interval *kh*, the controller calculates the desired admittance rate for interval *kh+h*, denoted *u(kh+h)*, from the measured average server utilization during interval *kh*, and the reference value, ρ_{ref}. The objective is to keep the server utilization as close as possible to the reference value. The gate rejects those requests that cannot be admitted. The requests that are admitted proceed to the rest of the system. The variable representing the number of arrivals during control interval *kh* is denoted α(*kh*). Since the admittance rate may never be larger than the arrival rate, the actual admittance rate, $\bar{u}$ =min[*u*, α].

2.1 Controllers

There are a number of different controllers that can be used. Here follows a description of the controllers that are used in this paper.

2.1.1. Step controller. The step-controller is a classical controller in the telecommunication field. The objective of the control law is to keep the control variable between an upper and a lower level. If the value of the variable is higher than the upper level, the admittance rate is decreased linearly. If the value is below the lower level, the admittance rate is increased. This means that the control law is as follows:

$$u(kh+h) = \begin{cases} u(kh) - s & \rho(kh) > \rho_{ref} + \varepsilon \\ u(kh) + s & \rho(kh) < \rho_{ref} - \varepsilon \end{cases}$$

where the value of *s* decides how much the rate is increased/decreased and the value of ε decides how much the control variable may deviate from the reference value.

2.1.2. PI-controller. The PI-controller is a well-known controller in automatic control. It uses two actions: one proportional and one integrating. The control law is given by:

$$u(kh+h) = K \cdot e(kh) + \frac{K}{T_i} \cdot \sum_{i=0}^{k-1} e(ih) \tag{1}$$

where $e(kh) = \rho_{ref} - \rho(kh)$ is the error between the control variable and the reference value. The gain K and the integral time T_i are the controller parameters that are set so that the controlled system behaves as desired.

2.1.3. RST-controller. For a more general controller structure, the polynomial methods proposed in [26] can be used. The control law in discrete-time for a general so-called RST-controller is given by

$$R(q)u(kh) = T(q)y_{ref}(kh) – S(q)y(kh) \tag{2}$$

where *R(q)*, *T(q)*, and *S(q)* are functions expressed in the forward-shift operator q (i.e. $qu(kh) = u(kh+h)$). *R*, *S*, and *T* are designed so that the closed loop system behaves as desired.

2.2 Gate

There are a number of well-known gates that can be used. In this paper we use a token bucket algorithm to reject those requests that cannot be admitted. Rejected requests are assumed to leave the system without retrials. An arriving request is only admitted if there is an available token. New tokens are generated at a rate of *u(kh)* tokens per second during control interval *kh*.

3. Control theoretic model

We use the discrete-time control theoretic model shown in Fig. 2. This model has been validated in [12] for the single server queue in Section 2. The model is a flow or liquid model in discrete-time.

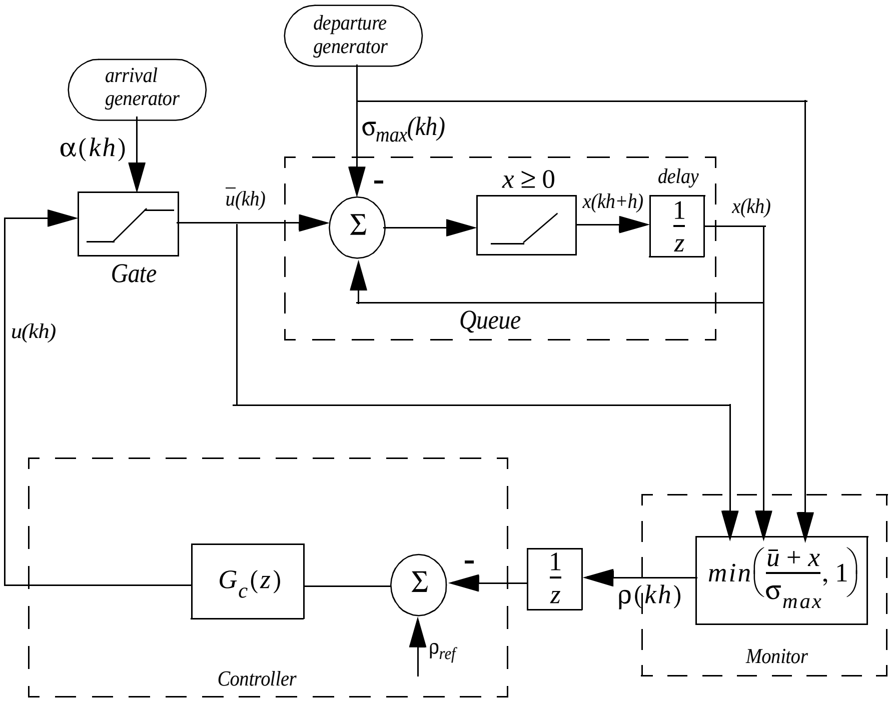

Fig. 2. A control theoretic model of a G/G/1-system with admission control.

The model is an averaging model in the sense that we are not considering the specific timing of different events, arrivals, or departures from the queue.

There are two stochastic traffic generators in the model. The *arrival generator* feeds the system with new requests. The number of new requests during interval *kh* is denoted $\alpha(kh)$. $\alpha(kh)$ is an integrated stochastic process over one sampling period with a distribution obtained from the underlying interarrival time distribution. The *departure generator* decides the *maximum* number of departures during interval *kh*, denoted $\sigma_{max}(kh)$. $\sigma_{max}(kh)$ is also a stochastic process with a distribution given by the underlying service time distribution.

The *gate* is constructed as a saturation block that limits the number of admitted requests during interval *kh*, $\bar{u}(kh)$, to be

$$\bar{u}(kh) = \begin{cases} 0 & u(kh) < 0 \\ u(kh) & 0 \leq u(kh) \leq \alpha(kh) \\ \alpha(kh) & u(kh) > \alpha(kh) \end{cases}$$

The *queue* is represented by its state *x(kh)*, which corresponds to the number of requests in the system at the end of interval *kh*. The difference equation for the queue is given by

$$x(kh+h) = f(x(kh) + \bar{u}(kh) - \sigma_{max}(kh))$$

where the limit function, *f(w)*, equals zero if *w*<0 and *w* otherwise. The limit function assures that $x(kh+h) \geq 0$. When the limit function is disregarded then the queue is a discrete-time integrator.

The *monitor* must estimate the server utilization since this is not directly measurable in the model. The server utilization during interval *kh*, $\rho(kh)$, is estimated as

$$\rho(kh) = min\left(\frac{\bar{u}(kh) + x(kh)}{\sigma_{max}(kh)}, 1\right)$$

The objective of the *controller* is to minimize the difference between the server utilization during interval *kh*, $\rho(kh)$, and the reference value, ρ_{ref}. The control law is given by the transfer function, $G_c(z)$.

4. Controller design

In the numerical investigations, we compare different controllers for the system described in Section 2. The system we investigated had an average service time of 0.02 seconds and the reference load, ρ_{ref}, was set to 0.8. However, before implementing a controller into the system, the controller parameters must be designed so that the system behaves as desired.

4.1 Some notes on control design and analysis

In this section we will use linear control design methods for finding parameters for PI- and RST-controllers. This means that we during the design consider a deterministic system with no active saturations. However, the real queueing system will for instance only allow positive queue lengths. In the end of this section we, therefore, analyse the closed loop system where the nonlinearity in the queue is taken into consideration. A stability region is determined where the PI-controller based on linear design is shown to also guarantee stability for the nonlinear system.

4.2 Static controller

We used a static controller as a benchmark controller when investigating the other controllers. A static controller uses a fixed acceptance rate, u_{fix}, that is set so that the average value of the control variable should be equal to the reference value. u_{fix} is given by

$$u_{fix} = \rho_{ref} \cdot \sigma \cdot h$$

which in this case is equal to 40 jobs per second.

4.3 Step-controller

The parameters for the step-controller are the step size, s, the marginal, ε, and the sampling period, h. These parameters are usually chosen by ad-hoc methods, that is with “trial and error” methods. Our investigations demonstrated a problem with the step-controller. If the sampling period is too short, the controller will overestimate the load of the system, and thereby create a stationary error in the controlled load. For our system, a sampling period of 2 seconds was necessary to eliminate the stationary error. Also, we selected a step size of 5, and a marginal of 0.05. With these parameters the steady-state behavior of the controlled system became close to the benchmark controller. It was not possible to find a step-controller that had a better steady-state behavior than the static controller.

4.4 PI-controller

The control law for the PI-controller expressed in z-transform is given by

$$G_c(z) = K\left(1 + \frac{1}{T_i} \cdot \frac{h}{z-1}\right)$$

The linear transfer function from the desired utilization, ρ_{ref}, to the (delayed) output, ρ, will be

$$G_{cl} = \frac{G_c(1+G_q)G_m}{1+G_c(1+G_q)G_m} = \frac{z \cdot K(T_i z - T_i + h)}{z \cdot (z^3 \sigma T_i + (KT_i - \sigma T_i)z^2 + Kzh + (-KT_i + Kh))} \qquad \textbf{(3)}$$

where $G_c = K\left(1 + \frac{1}{T_i}\frac{h}{z-1}\right)$, $G_q = \frac{1}{z-1}$, and $G_m = \frac{1}{\sigma} \cdot \frac{1}{z}$ are the transfer functions for the controller, for the queue, and for the monitor, respectively. σ is the average value of σ_{max}. The characteristic polynomial for the linear closed loop system will be

$$z \cdot \left(z^2 + \frac{K - 2\sigma}{\sigma}z + \frac{-KT_i + Kh + \sigma T_i}{\sigma T_i}\right) \tag{4}$$

where the pole at z=0 is cancelled in the transfer function from the input (the load reference) to the desired output (the load). Assume that the desired characteristic equation is

$$z(z^2 + a_1 z + a_2) = 0$$

The values of the controller parameters that gives this are

$$K = 2\sigma + a_1\sigma \qquad T_i = h \cdot \frac{2 + a_1}{1 + a_1 + a_2}$$

The controller parameters K and T_i influence the closed loop response for the system and need to be determined with respect to stability and robustness. The behavior of the PI-controller becomes better when the sampling period is short (should match desired dynamics). Therefore, for these investigations we used a sampling period of 0.2 seconds (h=0.2). This means that $\sigma = 10$, since σ is the average maximum number of departured jobs during a control interval.

Root locus arguments will show that reasonable parameters gives a stable closed loop system. For large gains there will be a closed loop pole located on the negative real axis, see Fig. 3. This may cause undesired behavior if neglected in the design. Choosing {K, T_i}={12, 0.6} the roots of the characteristic polynomial in eq. (4) will be $0, 4 \pm 0, 2i$. These poles are rather well damped and the transients from the pole on the real axis will decay fast. This set of controller parameters can, therefore, be seen as a “good” choice.

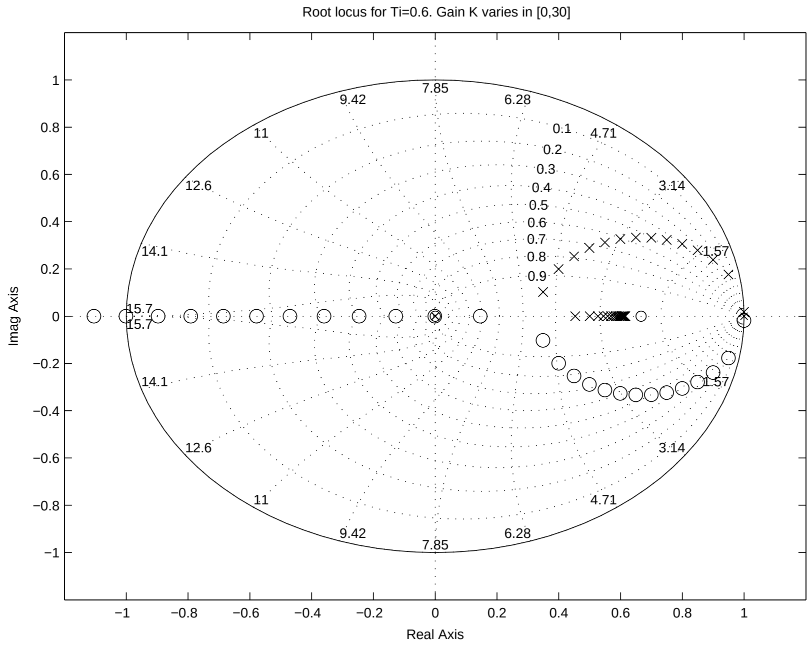

Fig. 3. Root locus diagram for T_i=0.6. The gain K varies between [0,30].

4.5 RST-controller

As an alternative to the PI-design in the previous section we can use polynomial design for so called RST-controllers, see [26]. The controller will be a two-degree-of-freedom controller, which allows for separate feedback and prefiltering design (pole-placement and handling of reference values).

For a pole placement design the controller polynomials (R, S, and T) are found by solving a Diophantine equation,

$$A(q)R(q) + B(q)S(q) = A_m(q)A_o(q) \qquad \mathbf{(5)}$$

describing the relationship of the process polynomials (A and B), the controller polynomials and the desired closed loop polynomials (A_m and A_o). The linear system has

$$\begin{aligned} A(q) &= q^2 - q \\ B(q) &= \frac{1}{\sigma} \cdot q \end{aligned} \qquad \mathbf{(6)}$$

A_m and A_0 are chosen so that the poles of the closed loop system are placed as desired. Note that there is a stable pole-zero cancellation at z=0 in the transfer function from the control signal *u* to the load. If the desired closed loop poles are chosen as {0.4, 0.2} we get

$$A_m A_0 = q^2 - 0,6q + 0,08 \tag{7}$$

If we impose the controller to have integral part, the controller polynomials are given by

$$\begin{aligned} R(q) &= q - 1 \\ S(q) &= 14q - 9,2 \\ T(q) &= 6q - 1,2 \end{aligned} \tag{8}$$

4.6 Stability analysis

To analyze the stability of the closed loop system where we also take the queue nonlinearity into account, we partition the system into a linear part in feedback connection with the nonlinearity, ϕ, see Fig. 4. The system is controlled with a PI-controller. The linear subsystem, G_z, can be shown to have the characteristic polynomial

$$G_z = z \cdot \sigma T_i \left(z^2 + \left(-1 + \frac{K}{\sigma} \right) z + \frac{K(h - T_i)}{\sigma T_i} \right) \tag{9}$$

To guarantee stability of the nonlinear closed loop system we use the *Tsypkin criterion* [22][23][16]. Sufficient conditions for stability are that G_z has all its poles within the unit circle $|z| < 1$ and that there exists a (positive) constant η, such that

$$Re[(1 + \eta(1 - z^{-1}))G_z(z)] + \frac{1}{k} \geq 0 \quad \text{for} \quad z \leq e^{iw}, w \geq 0 \tag{10}$$

where the nonlinearity ϕ belongs to the cone [0, *k*=1].

In the upper plot of Fig. 5 we have the *stability triangle* for the characteristic polynomial of Eq. (3) (see [26] for more details). The plot shows the relationship between the coefficients of the charac-

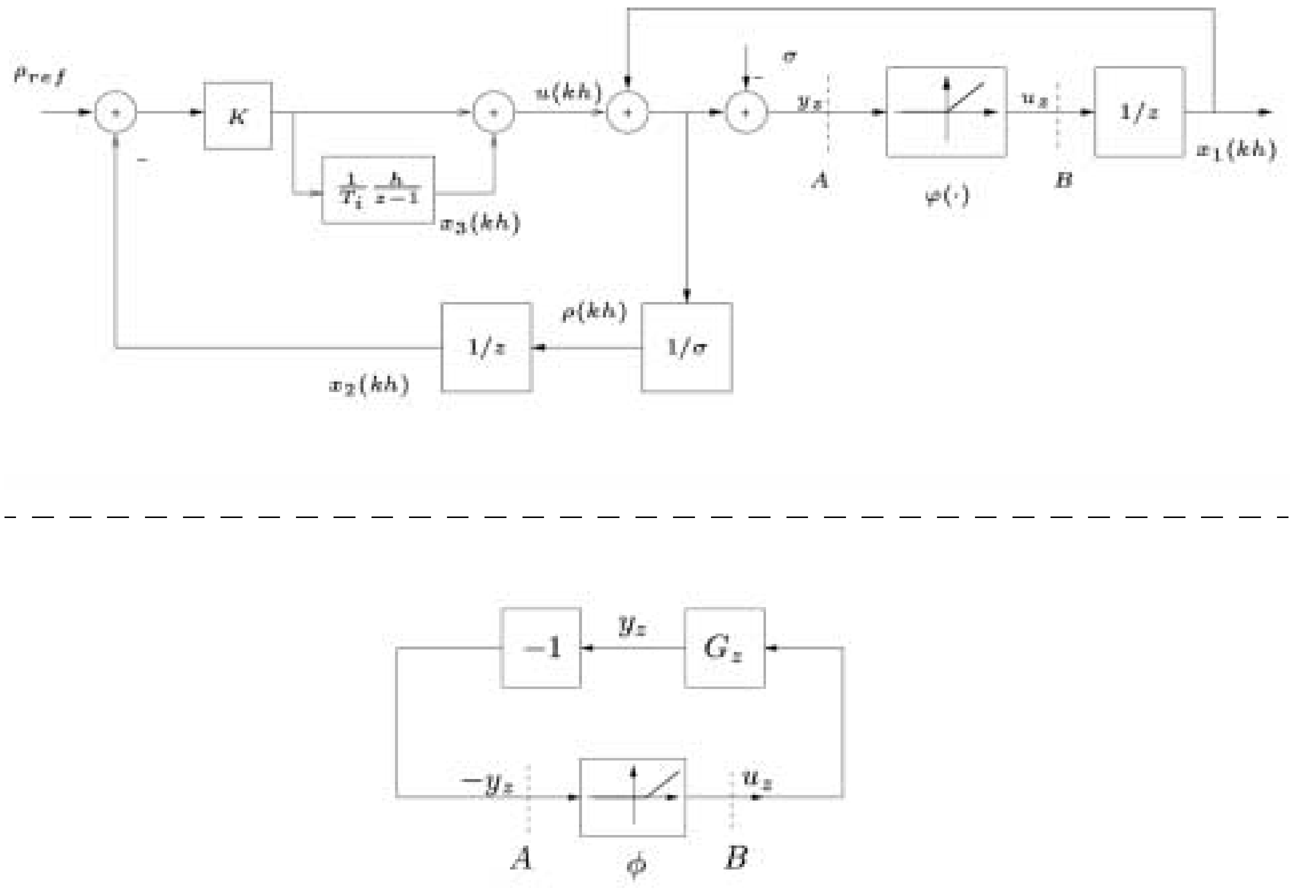

Fig. 4. Decomposition into a linear block (G_z) and a nonlinear block (ϕ) under negative feedback.

teristic polynomials for the closed loop linear model, area *A1*, and for the linear transfer function G_z, area *A2*, respectively. As an alternative representation, the lower plot in Fig. 5 shows the corresponding pole locations. As both area *A1* and *A2* lie within the stability triangle the corresponding poles will be located within the unit circle. Fig. 6 shows a graphical representation of the Tsypkin condition (10) for this set of control parameters. The dashed non-intersecting line in Fig. 6 corresponds to the existence of a positive parameter η satisfying (2). Thus, absolute stability for the nonlinear system also is guaranteed for this choice of parameters. Details on the calculations in this section can be found in [19].

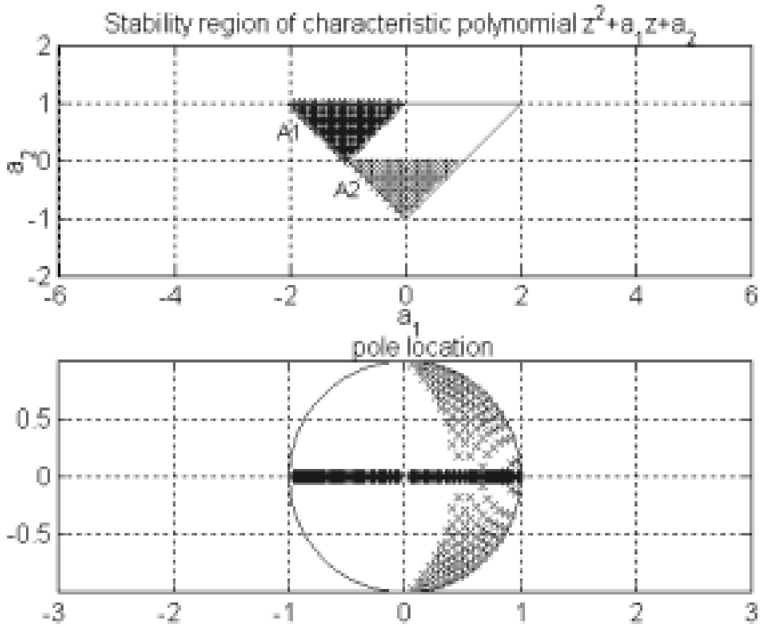

Fig. 5. Stability area (upper) and pole locations (lower) for stabilizing PI-controllers.

Remark: The Tsypkin criterion guarantees stability for any cone bounded nonlinearity in [0,1] and we can thus expect to have some robustness in addition to the stability in our case.

5. Numerical investigations

The numerical investigations contain a comparison of the controllers described in the previous section. The queueing model was represented by a discrete-event simulation program implemented in C, and the control theoretic model was implemented with the Matlab Simulink package.

Two systems were used in the numerical investigations: one *M/M/1*-system and one $M/H_2/1$-system. In both cases the average service time was 0.02 seconds. The parameters for the H_2-distribution was μ_1=20, μ_2=600, and p_1=0.38 which gives a squared coefficient of variance of 3.74. Note that the controller design is independent of the type of arrival process and the service time distribution, since the system dynamics only depend on the average service time. However, the system becomes more difficult to control if there is more variation in the system. This will be shown in the results.

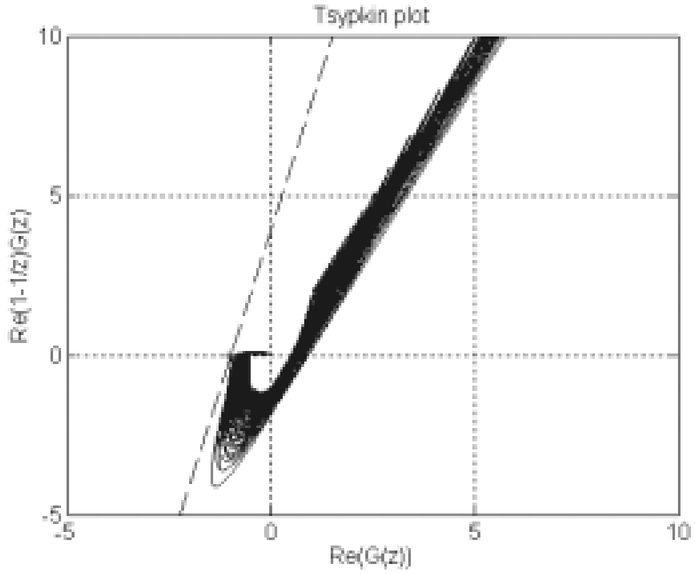

Fig. 6. Set of Tsypkin plots for G_z=G_z*(K,Ti)*, where (*K*,T_i) correspond to pole locations in Fig. 5.

5.1 Performance metrics

The admission control mechanism has two control objectives. First, it should keep the control variable at a reference value, i.e the control error, $e=y_{ref}-y$, should be as small as possible. Second, it should react rapidly to changes in the system, i.e the so-called settling time should be short.

Therefore, we compare the controllers in two ways. First, we show the steady-state distribution of the server utilization, by plotting the estimated distribution function, i.e. $P(\rho \le \rho_0)$ where $0 \le \rho_0 \le 1$. The distribution function was estimated from 5000 measurements of the server utilization for a specific parameter setting. Each measurement is the average server utilization during one second. The distribution function shows how well the controller meets the first control objective. Second, we plot the step response when starting with an empty system. The step responses show the transient behavior of the controllers.

5.2 Controller comparisons

This section contains a comparison of the controllers that were designed in Section 4. During all investigations, the reference load was set to 0.8, and the average arrival rate was 150 jobs per second.

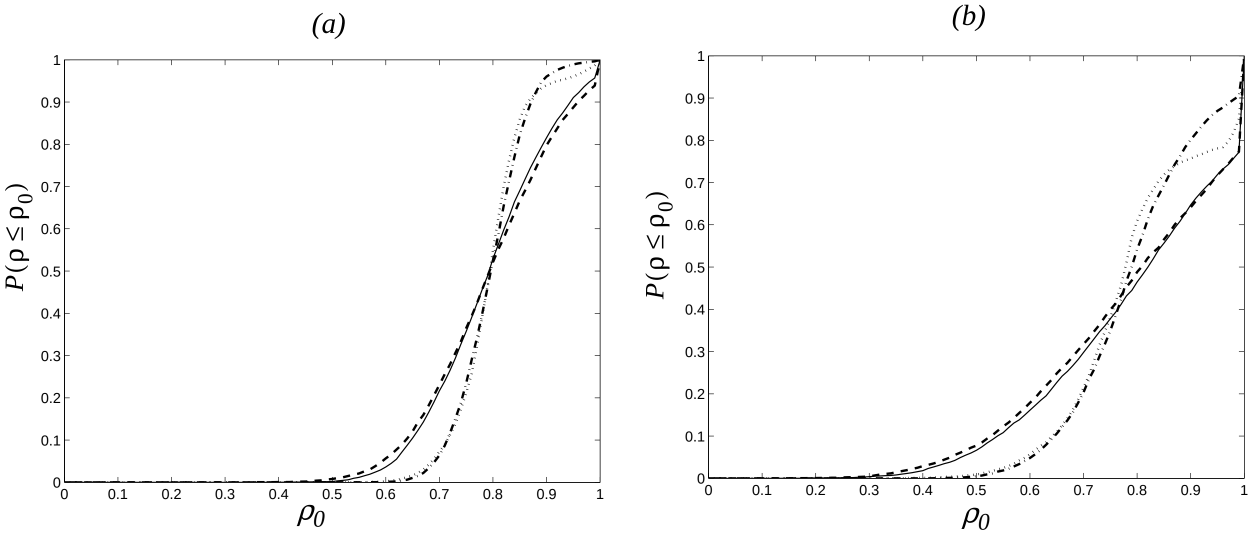

Fig. 7. Distribution functions: (a) M/M/1-system, (b) M/H_2/1-system.
solid line: static controller, dashed line: step-controller,
dotted line: PI-controller, dash-dotted line: RST-controller

5.2.1. Distribution function. The distribution functions for the two systems are shown in Fig. 7. The optimal distribution function is zero for $0 \le \rho \le 0,8$ and one for $0,8 \le \rho \le 1$. The systems with a step-controller behaves similar to the system with a static controller. we can conclude that the M/H_2/1-system is more difficult to control than the M/M/1-system, since the distribution functions are remoter from the optimum function. However, as can be seen, the systems with a PI-controller and an RST-controller actually behave better than the system with a static controller. This phenomenon is due to that those controllers can adapt to the stochastic variations in the system. This behavior requires a short sampling period. With a longer sampling period, for example one second, the controllers behave as the static controller.

5.2.2. Average step response. The step responses for the load controlled M/M/1-system are shown in Fig. 8. The step responses for the M/H_2/1-system are similar. As can be seen, the controllers have very divergent step responses. The fastest controller is of course the static controller, since it already from start is set to an accurate admittance rate. However, the PI-controller has found a correct admittance rate only after a few seconds, whereas the RST-controller is slightly slower. The step-controller

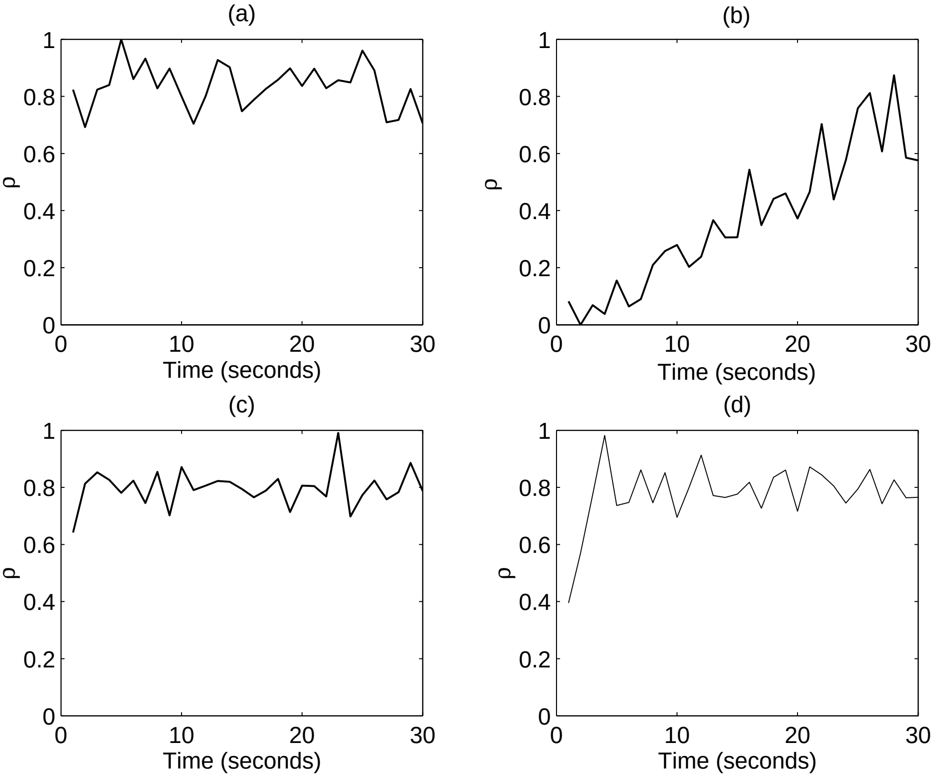

Fig. 8. Step responses for the load controlled M/M/1-system.
(a) static controller, (b) step-controller, (c) PI-controller, (d) RST-controller

is very slow. Even after 30 seconds it has not managed to find a correct admittance rate. This is the main problem with the step-controller. In order to achieve a good steady-state behavior, the step size must be small and the sampling period must be long. However, this means that it takes a long time for the controller to adapt to changes in the system.

5.2.3. Robustness. A good controller should maintain a good performance even when the system parameters change, that is the controller should be robust to modelling errors. In a real system, it is likely that the average service time will change slowly with time, for example due to changes in the user behavior. Therefore, we investigated the robustness of the controllers by changing the average service time in the system. In the first case, the average service time was increased with 30% ($\bar{x} = 0,026$ seconds), and in the second case the average service time was decreased with 30% ($\bar{x} = 0,014$ seconds).

All controllers were tested, however we only show the results for the PI-controller in Fig. 9. As can be seen, the PI-controller works well even when the average service time is changed. The step-controller has a similar result, The RST-controller is very robust, since the distribution function is the same even when the average service time is changed with 30%. The static controller is, of course, dependent on a correct service time, which means that it cannot operate properly when the service times change.

5.3 Limitations with linear design

The results from our investigations show that linear design methods seem to work well for load control mechanisms. However, there are some limitations with linear design that should be considered.

For example, a predicted instability of the linear system does not necessarily show up either in the nonlinear model nor in simulations or experiments. The controller parameters $\{K, T_i\}=\{4, 0.15\}$ give unstable poles outside the unit circle, which means that the closed loop system should oscillate. However, the nonlinear system behaves very well, as shown in Fig. 10(a). The distribution function for the previous design is shown as comparison.

The analysis in 4.6 gives sufficient conditions and a region for control parameters which guarantee stability of the nonlinear closed loop as well as for the simplified linear model. We are of course not restricted to choose parameters from only this region as the main objective is that the nonlinear system should be stable. It is possible to find some pole locations where we can use the Tsypkin criterion to show stability of the nonlinear system which could not be predicted by linear system analysis. However, the results are only sufficient and simulations indicate that control parameters which would render the linear system unstable (poles outside the unit circle) and which do not satisfy the Tsypkin criterion actually show good performance. See [19] for further comments on this topic.

Another limitation is that a linear model allows for arbitrary pole placement where the corresponding control signal may not be realizable due to e.g. bounded signals. In our case, this is readily

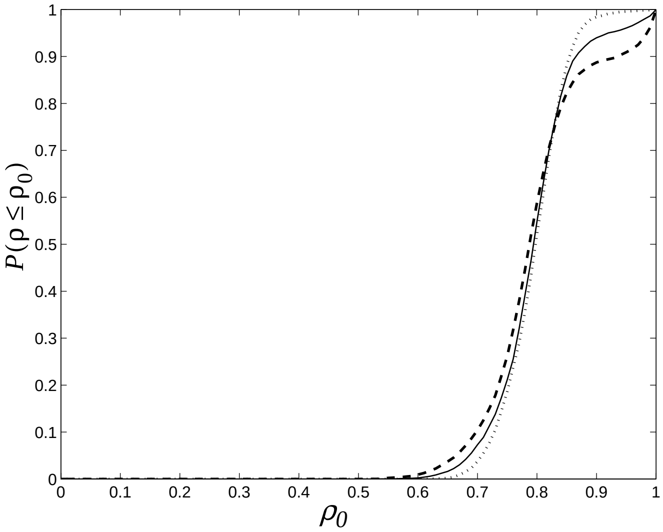

Fig. 9. Robustness of PI-controller: Server utilization distribution function
Solid line: $\bar{x} = 0,02s$, dotted line: $\bar{x} = 0,014s$, dashed line: $\bar{x} = 0,026s$

shown by comparing step responses. If the closed loop poles in the RST design are chosen as {0.4, 0.4, 0.2} we get

$$\begin{aligned} R(q) &= q^2 - 1 \\ S(q) &= 13,2q - 10,3 \\ T(q) &= 3,6q - 0,72 \end{aligned}$$

With this design, the linear system has a very short settling time. However, simulations show that the real system has a much longer settling time, see Fig. 10(b). Note that in the linear model, the server utilization can be negative. The most important limitation in the transient phase is the nonlinearity in the queue. The effect of large negative changes in the queue length will be bounded (as they should) and will cause the step response of the nonlinear system to be slower than what a pure linear design would predict. The results from the nonlinear model align well with the simulations.

6. Conclusions

Admission control mechanisms have since long been developed for various server systems. Traditionally, queueing theory has been used when investigating server systems, since they usually can be modelled as queueing systems. However, there are no mathematical tools in queueing theory that can be used when designing admission control mechanisms. Therefore, these mechanisms have mostly

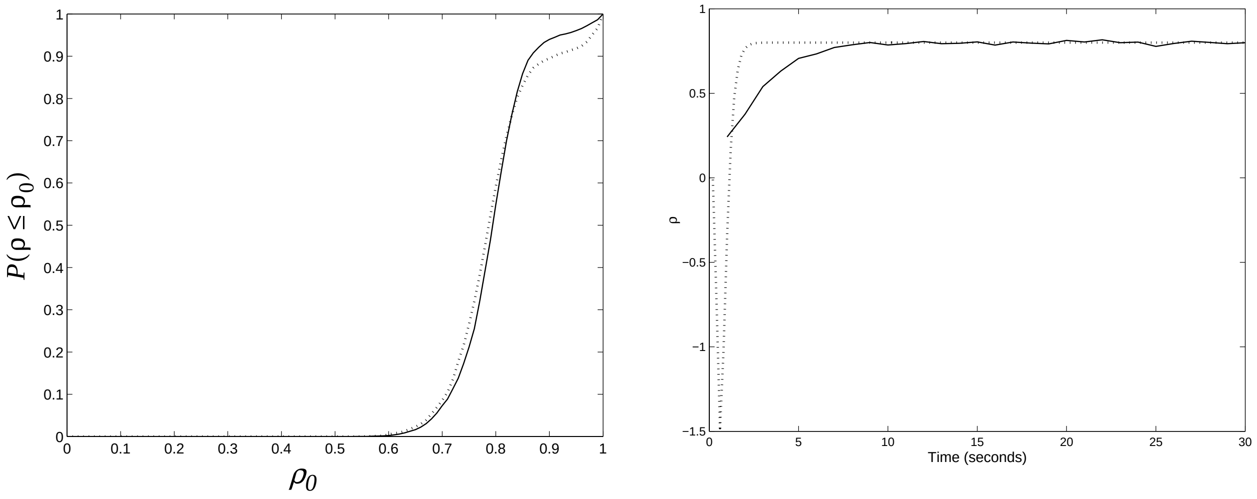

Fig. 10. Limitations with linear design,
(a) Distribution function for PI-controller; Solid line: $\{K, T_i\}=\{12, 0.6\}$, Dotted line: $\{K, T_i\}=\{4, 0.15\}$
(b) Step response for RST-controller; Solid line: discrete-event simulations (average of 100 simulations)
Dotted line: linear model

been developed with empirical methods. Control theory contains many mathematical tools that can be used when designing admission control mechanisms.

In this paper, we have designed load control mechanisms for a $G/G/1$ -system with control theoretic methods. We have compared a PI-controller and an RST-controller, both commonly used in automatic control, with a static controller and a step controller, both commonly used in telecommunication systems. We have shown that, when considering transient and stationary behavior, and robustness, both the PI-controller and the RST-controller behave better than the other controllers.

Also, we perform a nonlinear stability analysis for a PI-controlled system. We show that linear design is sufficient for stability of the nonlinear system. However, there are some limitations with linear design, which should be considered. One conclusion of this paper is that it is possible to use control theoretic methods when designing admission control mechanisms for server systems. The designs have been verified with simulations for discrete-event systems based on queuing theory.

References

[1] T.F. Abdelzaher and C. Lu, “Modeling and performance control of Internet servers”, Proc. of the 39th IEEE Conference on Decision and Control, 2000, pp 2234-2239.

[2] T.F. Abdelzaher, K.G. Shin and N. Bhatti, “Performance guarantees for web server end-systems: a control theoretic approach”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 1, Jan 2002, pp 80-96.

[3] C.E. Agnew, “Dynamic modeling and control of congestion-prone systems”, *Operations Research*, Vol. 24, No. 3, May-June 1976, pp 400-419.

[4] M. Andersson, M. Kihl, and A. Robertsson, “Modelling and design of admission control mechanisms for web servers using non-linear control theory”, Proc. of SPIE ITCom, 2003.

[5] “Apache web server”, http://www.apache.org

[6] G. Banga and P. Druschel, “Measuring the capacity of a web server”, USENIX Symposium on Internet Technologies and Systems, December 1997, pp. 61-71.

[7] P. Bhoj, S. Ramanathan, and S. Singhal, “Web2K: Bringing QoS to web servers”, HP Labs Technical report, HPL-2000-61, 2000.

[8] J. Carlström, R. Rom, Application-aware admission control and scheduling in web servers, Proc. Infocom, 2002.

[9] L. Cherkasova, P. Phaal, Predictive admission control strategy for overloaded commercial web server, Proc. 8th International IEEE Symposium on modeling, analysis and simulation of computer and telecommunication systems, 2000, pp. 500-507.

[10] R.A. Farel, M. Gawande, Design and analysis of overload control strategies for transaction network databases, Proc. ITC-13, A. Jensen and V.B. Iversen (Eds.), Teletraffic and Datatraffic in a period of change, Elsevier, 1991, pp. 115-120.

[11] M. Kihl, A. Robertsson, and B. Wittenmark, “Analysis of admission control mechanisms using non-linear control theory”, Proc. of IEEE International Symposium on Computer Communcations, 2003.

[12] M. Kihl, A. Robertsson, and B. Wittenmark, “Performance Modelling and Control of Server Systems using Non-linear Control Theory”, Proc. of 18th International Teletraffic Congress, 2003.

[13] A. Kolarov and G. Ramamurthy, “A control-theoretic approach to the design of an explicit rate controller for ABR service”, *IEEE/ACM Transactions on Networking,* Vol. 7, No. 5, Oct. 1999, pp 741-753.

[14] U. Körner and C. Nyberg, “Overload control in communication networks”, *Proc. of Globecom’91*, pp 1331-1335.

[15] U. Körner, Overload control of SPC systems, Proc. ITC-13, A. Jensen and V.B. Iversen (Eds.), Teletraffic and Datatraffic in a period of change, Elsevier, 1991, pp. 105-114.

[16] M. Larsen and P.V. Kokotovic, “A brief look at the Tsypkin criterion: from analysis to design”, International Journal of Adaptive Control and Signal Processing, 15:2, 2001, pp. 121-128.

[17] C. Lu, T.F. Abdelzaher, J.A. Stankovic and S.H. Son, “A feedback control approach for guaranteeing relative delays in web servers”, Proc. of the 7th IEEE Real-Time Technology and Applications Symposium, 2001, pp 51-62.

[18] A. Robertsson, B. Wittenmark, and M. Kihl, “Analysis and design of admission control in web-server systems”, American Control Conference, 2003.

[19] A. Robertsson, B. Wittenmark, M. Kihl, and M. Andersson, “Design and evaulation of load control in web-server systems”, Submitted to the American Control Conference, 2004.

[20] Y. Satake, A telephone network overload control scheme using adaptive control, Electronics and Communications in Japan, Part 1, Vol. 71, No. 10, 1988, pp. 63-71.

[21] P. Somoza, A. Guerrero, Dynamic processor overload control and its implementation in certain single-processor and multiprocessor SPC systems, Proc. ITC-9, 1979.

[22] Z.Y. Tsypkin, “Absolute stability of a class of nonlinear automatic sampled data systems”, Automation and Remote Control, 25:7, 1964, pp. 918-923.

[23] Z.Y. Tsypkin, “Frequency criteria for the absolute stability of nonlinear sampled-data systems”, Automation and Remote Control, 25:3, 1964, pp. 261-267.

[24] T. Voigt, P. Gunningberg, Adaptive resource-based web server admission control, Proc. 7th International Symposium on Computers and Communications, IEEE Computer Society, 2002.

[25] K. Wildling and T. Karlstedt, “Call handling and control of processor load in an SPC system, a simulation study”, *Proc. of the 9th International Teletraffic Congress*, 1979.

[26] K.J. Åström and B. Wittenmark, *Computer-controlled systems, theory and design*, Prentice Hall International Editions, 3rd Edition, 1997.