

# Design and evaluation of load control in web server systems

A. Robertsson, B. Wittenmark, M. Kihl, and M. Andersson

**Abstract**—Nonlinear discrete-time modeling of a web server system is investigated. Server systems typically contain nonlinearities such as saturations and bounded queue lengths. The incoming traffic and service rates are best modeled by stochastic processes, well described and analyzed by queuing theory. Here, we develop and validate a control theoretic model of a general single server queue, a so-called G/G/1-system. Based on the nonlinear system model, design of admission controllers are presented and the closed loop stability is analyzed. The behavior of the server model is verified with respect to queue theoretic models.

Finally, experimental evaluation is performed on an Apache web server in a laboratory network. A traffic generator is used to represent client requests. The control of the Apache server has been re-written to implement our algorithms. We show that the control theoretic model aligns well with the experiments on the web-server. Measurements in the laboratory setup show the robustness of the implemented controller, and how it corresponds to the results from the theoretical analysis and the simulations.

## I. 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 [1]. 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 modeled as a queuing 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 over-provision the systems for these traffic peaks, instead admission control mechanisms are implemented in the nodes.

In [2] and [3] a web server was modeled as a static gain to find optimal 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 [4]. In [5] a discrete-time queuing system with geometrically distributed inter-arrival times and service

times was analyzed. An admission control algorithm was developed using optimal control theory. An other recent contribution combining queuing and control theoretic results is [6]. In [7] a discrete-time linear MIMO-model for an Apache web server was identified and experimentally evaluated.

In [8] and [9], we have analyzed queue length controllers for M/G/1-system based on the nonlinear flow model in [10], and used this model for designing a PI-controller for the system. We demonstrated that linear models of the server system are insufficient to explain the behavior, since the non-linearities in the gate and mainly in the queue affect both the stability and performance properties and should be considered in the design process.

In [11], 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 this paper, we use this model for nonlinear analysis and design of controller parameters for a PI-controller.

In Section II we briefly recapitulate the discrete-time server model and in Section III we examine the stability properties for the closed loop system when the admission controller is a PI-controller. Furthermore, the paper contains a discussion about the limitations with both linear control theoretic models of queuing systems and linear design methods. In Section IV the implementation of control algorithms and overload experiments on an Apache web server are described and reported. Finally there is a discussion of the stability results and the results from simulations and experiments in Section V. Section VI contains the conclusions.

## II. SYSTEM MODEL

The system model is shown in Fig. 1. We assume that the system may be modeled 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 which measures the average server utilization  $\rho(kh)$  during interval  $kh$ . Based on the the reference value,  $\rho_{ref}$  and the estimated or measured load situation  $\rho$  the controller calculates the desired admittance rate  $u(kh)$ . The server utilization can be estimated as

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

where  $\sigma$  is the service rate in the interval. The analysis in the next section considers a fixed average service rate.

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

Anders Robertsson and Björn Wittenmark are with the Department of Automatic Control Lund Institute of Technology, Box 118, SE-221 00 Lund, Sweden, {andersro|bjorn}@control.lth.se, Maria Kihl and Mikael Andersson are with the Department of Communication System, Lund Institute of Technology, Box 118, SE-221 00 Lund, Sweden, {maria|mike}@telecom.lth.se

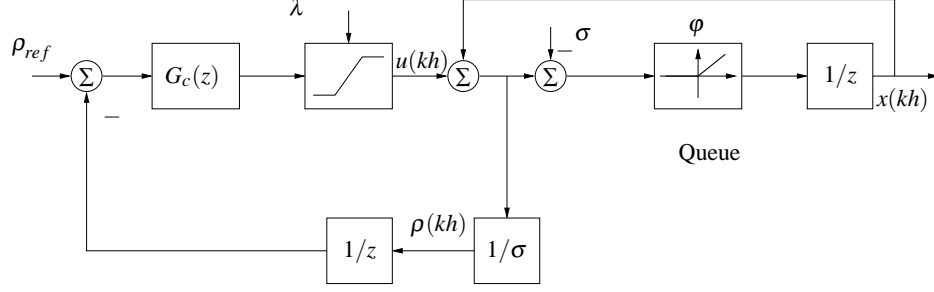


Fig. 1. Discrete-time model with controller saturation and saturation  $\varphi$  for positive queue lengths.

admitted proceed to the rest of the system. The variable representing the number of arrivals during control interval  $kh$  is denoted  $\lambda(kh)$  see Fig. 1. Since the admittance rate may never be larger than the arrival rate, the actual admittance rate  $u$  is saturated in the interval  $[0, \lambda]$ .

### III. ANALYSIS OF CLOSED LOOP SYSTEM

In this section we will consider the stability properties of the controlled server node, when using a PI-controller for load control. First we will consider an approach based on a linear queue model and compare with the admissible control parameters derived from nonlinear analysis. The analysis is based on the Tsytkin/Jury-Lee stability criterion (discrete-time versions of the Popov criterion) [12], [13], [14]. In the analysis only the dominating 'queue-limitation'  $\varphi$  will be considered. See Section V for comments on the saturation.

#### A. Linear design (neglecting saturations)

Neglecting the nonlinearities in Fig. 1 (assuming  $\varphi(z) = z$ , i.e., linear and no saturation) and using a standard PI-controller  $G_c(z) = K(1 + \frac{1}{T_i} \cdot \frac{h}{z-1})$  will result in the closed loop dynamics

$$G_c = \frac{G_c(1+G_q)G_m}{1+G_c(1+G_q)G_m} = \frac{z \cdot K/\sigma(z-1+h/T_i)}{z \cdot (z^2 + (K/\sigma - 2)z + (1 - K/\sigma + Kh/(\sigma T_i)))} \quad (1)$$

where  $G_q$  and  $G_m$  represent the queue and monitor dynamics, respectively. To match the characteristic polynomial

$$z \cdot (z^2 + (K/\sigma - 2)z + (1 - K/\sigma + Kh/(\sigma T_i))) \quad (2)$$

with a desired characteristic polynomial

$$z \cdot (z^2 + a_1 z + a_2) \quad (3)$$

we get the control parameters

$$K = (2 + a_1) \sigma, \quad T_i = h(2 + a_1)/(1 + a_1 + a_2)$$

Using the parameters of the PI-controller it is thus possible to make an arbitrary pole-placement, except for the pole  $z = 0$ , which corresponds to a time delay. A simplified linear

analysis will thus predict stability for the closed loop for all coefficients  $\{a_1, a_2\}$  belonging to the stability triangle

$$\{a_2 < 1, \quad a_2 > 1 + a_1, \quad a_2 > 1 - a_1\}, \quad (4)$$

see [15].

#### B. Model with queue limitation

Consider the admission control scheme in Fig. 2 where we have introduced the states  $\{x_1, x_2, x_3\}$  corresponding to the queue length, the (delayed) utilization  $\rho$  and the integrator state in the PI-controller, respectively.

The state space model will be

$$\begin{aligned} x_1(kh+h) &= \varphi(u + x_1(kh) - \sigma) \\ x_2(kh+h) &= \frac{1}{\sigma}(u + x_1(kh)) \\ x_3(kh+h) &= Kh/T_i(\rho_{ref} - x_2(kh)) + x_3(kh) \end{aligned} \quad (5)$$

where  $u = K(\rho_{ref} - x_2) + x_3$  and  $\varphi(\cdot)$  is the saturation function in Fig. 2. By introducing the *forward shift operator* and leaving out the time arguments, we get

$$qx_1 = \varphi(K(\rho_{ref} - x_2) + x_3 + x_1 - \sigma) \quad (6)$$

$$qx_2 = \frac{1}{\sigma}(K(\rho_{ref} - x_2) + x_3 + x_1) \quad (7)$$

$$qx_3 = Kh/T_i(\rho_{ref} - x_2) + x_3 \quad (8)$$

The equilibrium for the system (6–8) satisfies  $qx = x$ .

From (8) we get

$$x_3 = Kh/T_i(\rho_{ref} - x_2) + x_3 \Rightarrow x_2^o = \rho_{ref}$$

Inserting this in (6) and (7) we get

$$\begin{aligned} x_1^o &= \varphi(x_3^o + x_1^o - \sigma) \\ x_2^o &= \rho_{ref} = \frac{1}{\sigma}(x_3^o + x_1^o) \\ &\Rightarrow \\ x_1^o &= \varphi(\sigma(\rho_{ref} - 1)) \end{aligned} \quad (9)$$

As  $\rho_{ref} \in [0, 1]$  and using the fact that  $\varphi(z) = 0, \forall z \leq 0$  we get

$$\begin{cases} x_1^o = 0 \\ x_2^o = \rho_{ref} \\ x_3^o = \sigma x_2^o = \sigma \rho_{ref} \end{cases} \quad (10)$$

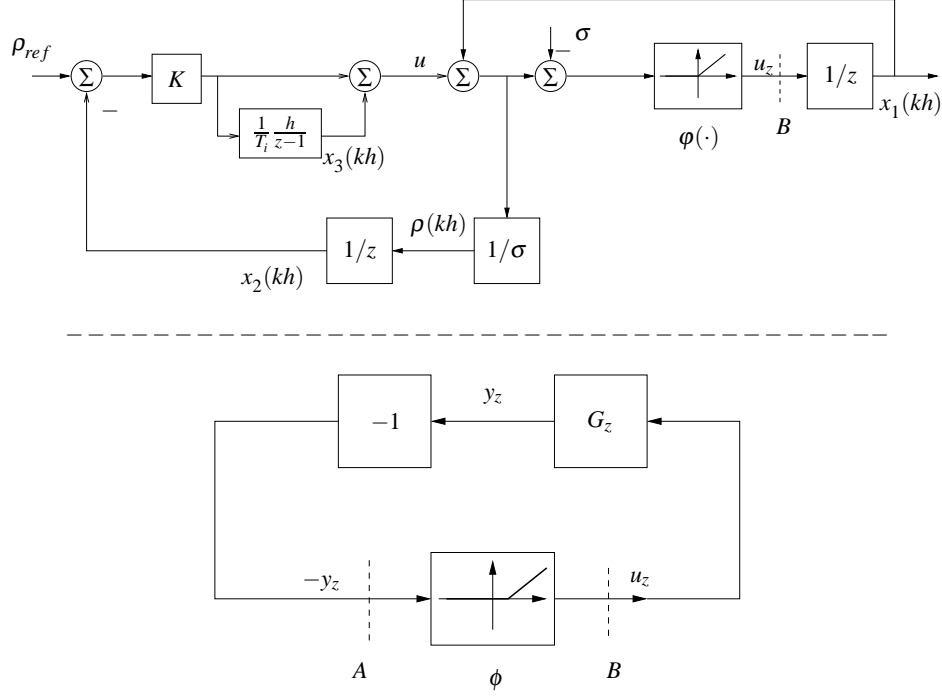


Fig. 2. Decomposition into a linear block ( $G_z$ ) and a nonlinear block ( $\phi$ ) under negative feedback.

By introducing the change of variables

$$\begin{cases} z_1 = x_1 - 0 \\ z_2 = x_2 - \rho_{ref} \\ z_3 = x_3 - \sigma \rho_{ref} \end{cases} \quad \text{or} \quad \begin{cases} x_1 = z_1 \\ x_2 = z_2 + \rho_{ref} \\ x_3 = z_3 + \sigma \rho_{ref} \end{cases}$$

we get

$$\begin{aligned} qz_1 &= qx_1 - 0 &= \phi(-Kz_2 + z_3 + z_1 - \sigma) \\ qz_2 &= qx_2 - \rho_{ref} &= \frac{1}{\sigma}(-Kz_2 + z_3 + \sigma \rho_{ref} + z_1) - \rho_{ref} \\ qz_3 &= qx_3 - \sigma \rho_{ref} &= -Kh/T_i z_2 + z_3 + \sigma \rho_{ref} - \sigma \rho_{ref} \end{aligned}$$

Rewriting this as a linear system in negative feedback with the nonlinear function  $\phi : y \rightarrow \phi(y - \sigma(1 - \rho_{ref}))$ , we get

$$\begin{aligned} qz &= A_z z + B_z u_z = A_z z + B_z \phi(-y) \\ y &= C_z z \end{aligned}$$

$$q \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1/\sigma & -K/\sigma & 1/\sigma \\ 0 & -Kh/T_i & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \phi(-y)$$

$$y = \begin{bmatrix} -1 & K & -1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Note that for  $\rho_{ref} \in [0, 1]$  the function  $\phi(\cdot)$  will belong to the same cone as  $\varphi(\cdot)$ , namely  $[\alpha, \beta] = [0, 1]$ , see Fig. 3. The incremental variation will also have the same maximal value ( $=1$ ).

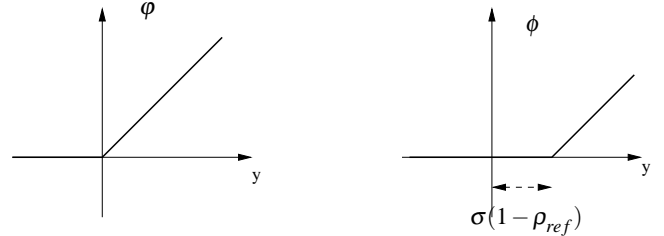


Fig. 3.  $\phi(y) = \varphi(y - \sigma(1 - \rho_{ref}))$  where  $\sigma > 0$  and  $\rho_{ref} \in [0, 1]$ .

The transfer function  $G_z = G_{u_z \rightarrow y_z}(z)$  from cut B to cut A in Fig. 2 will be

$$\begin{aligned} G_z &= C_z(zI - A_z)^{-1} B_z \\ &= \frac{-z \cdot (z - 1)}{z \cdot (z^2 + (-1 + K/\sigma)z + K(h - T_i)/(\sigma T_i))} \end{aligned} \quad (11)$$

For the forthcoming stability analysis we determine for which control parameters the linear subsystem  $G_z$  is stable.

The poles of (11) are stable for the area depicted in Fig. 4 for the normalized parameters  $K/\sigma$  and  $h/T_i$ .

### C. Stability analysis for discrete-time nonlinear system

To determine the stability for the nonlinear system in (11) we can use the Tsytkin criterion or the Jury-Lee criterion which are the discrete-time counterparts of the Popov criterion for continuous time systems [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  $\eta$  such that

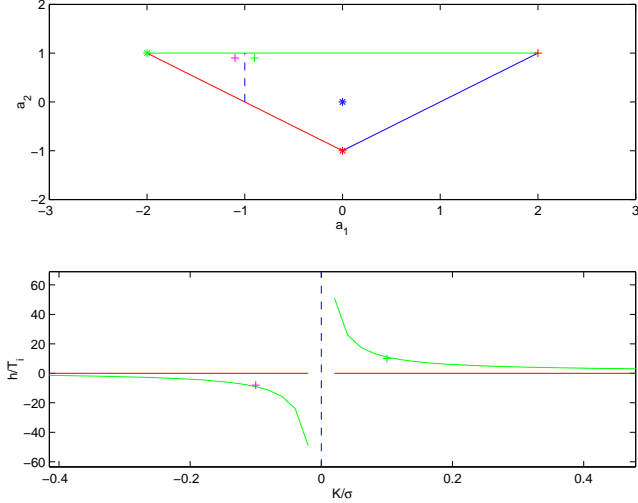


Fig. 4. (Upper:) The internal of the triangle corresponds to the stability area for the characteristic polynomial  $z \cdot (z^2 + a_1 z + a_2)$  of  $G_z$ . (Lower:) Corresponding stabilizing control parameters  $\{K/\sigma, h/T_1\}$ .

$$\text{Re}[(1 + \eta(1 - z^{-1}))G_z(z)] + \frac{1}{k} \geq 0 \text{ for } z = e^{i\omega}, \omega \geq 0 \quad (12)$$

where the nonlinearity  $\phi$  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.(2). By choosing coefficients for the characteristic polynomial (2) in the upper left triangle (A1) we will get controller parameters  $\{K, T_i\}$  which also will give a stable transfer function  $G_z$ . The corresponding poles are plotted in the lower diagram of Fig. 5. Fig. 6 shows a graphical representation of the Tsytkin condition (12) for this set of control parameters. The green (dashed) non-intersecting line in Fig. 6 corresponds to the existence of a positive parameter  $\eta$  satisfying Eq.(12). Thus, absolute stability for the nonlinear system also is guaranteed for this choice of parameters.

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

#### IV. EXPERIMENTS

The admission control mechanism was implemented in the Apache [17] web server. Apache is made up of a core package and several modules that handle different operations, such as Common Gateway Interface (CGI) execution, logging, caching etc. A new module was created that contains the admission control mechanisms. The new module was then hooked into the core of Apache, so that it was called every time a request was made to the web server. The module could then either reject or admit the request according to the control mechanism. The admission control mechanism was written in C and tested on a Windows platform. A discrete-event simulation program implemented

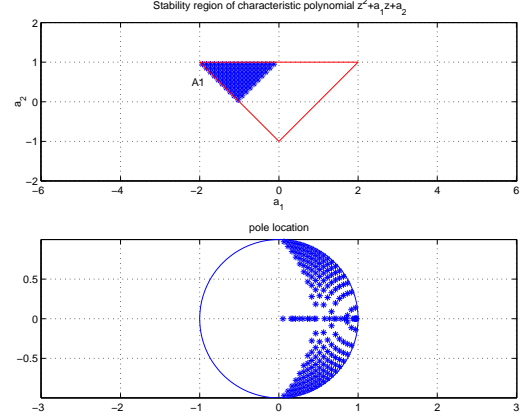


Fig. 5. (Upper:) The large triangle is the stability area a linear model would predict. However, from this parameter set, nonlinear analysis guarantees only stability for parameters  $\{a_1, a_2\}$  in the upper left triangle (A1,  $*$ ). (Lower:) Pole location corresponding to  $\{a_1, a_2\} \in A1$ .

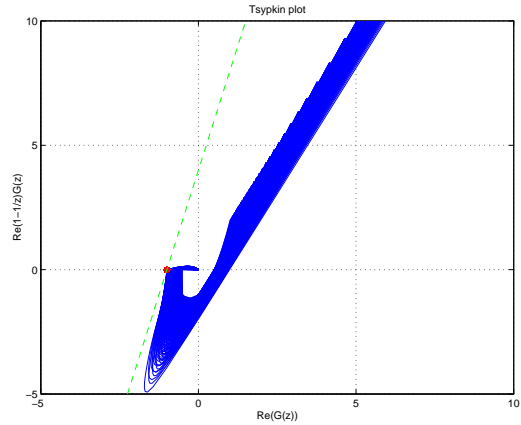


Fig. 6. Set of Tsytkin plots which all satisfy the frequency condition (12) for  $G_z = G_z(K, T_i)$ , where  $(K, T_i)$  correspond to pole locations in Fig. 5

in C, and the control theoretic models were implemented with the Matlab/Simulink package. The traffic generators in the discrete-time model were built as Matlab programs. They generate arrivals and departures according to the given statistical distributions. We tested the system by sending the generated traffic to the server controlled by the suggested admission controllers, and collecting performance metrics such as the server utilization distribution and step responses.

The results from the experiments on the Apache server were compared with the results from the discrete-event simulations.

#### A. Setup

Our measurements used one server computer and one computer representing the clients connected through a 100 Mbits/s Ethernet switch. The server was a PC Pentium III 1700 MHz with 512 MB RAM running Windows 2000 as operating system. The computer representing the clients was a PC Pentium II 400 MHz with 256 MB RAM running

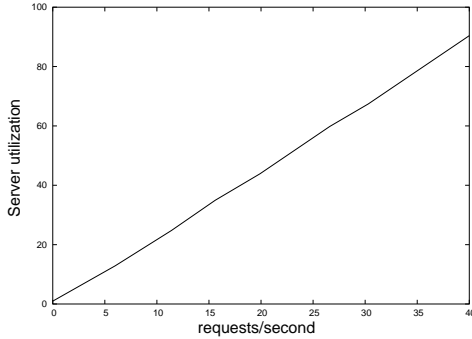


Fig. 7. Average server utilization for the open-loop system.

RedHat Linux 7.3. Apache 2.0.45 was installed on the server. We used the default configuration of Apache. The client computer was installed with a HTTP load generator, which was a modified version of S-Client [18]. The S-Client is able to generate high request rates even with few client computers by aborting TCP connection attempts that take too long time. The original version of S-Client uses deterministic waiting times between requests. We modified the code to use Poissonian arrivals instead. The client program was programmed to request dynamically generated HTML files from the server. The CGI script was written in Perl. It generates a random number of random numbers, adds them together and returns the summation. The average request rate was set to 100 requests per second in all experiments except for the measurements in Fig. 7. The modified Apache version was installed on the server. In all experiments, the control interval was set to one second.

### B. Validation of the Model

We have validated that the open-loop system, that is without control feedback, is accurate in terms of average server utilization. The average server utilization for varying arrival rates are shown in Fig. 7. For a single-server queue, the server utilization is proportional to the arrival rate, and the slope of the server utilization curve is given by the average service time. The measurements in Fig. 7 gives an estimation of the average service time in the web server,  $1/\mu=0.0255$ .

### C. Controller parameters

Control parameters for the PI-controller are chosen from the stability area A1 in Fig. 5. In the simulations and experiments below we use  $\{K, T_i\}=\{20, 2.8\}$ .

### D. Performance metrics

An admission control mechanism have two control objectives. First, it should keep the control variable at a reference value, i.e., the error,  $e = \rho_{ref} - \rho$ , 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 tested the mechanism in two ways. First, we show the steady-state distribution of the output variable, by

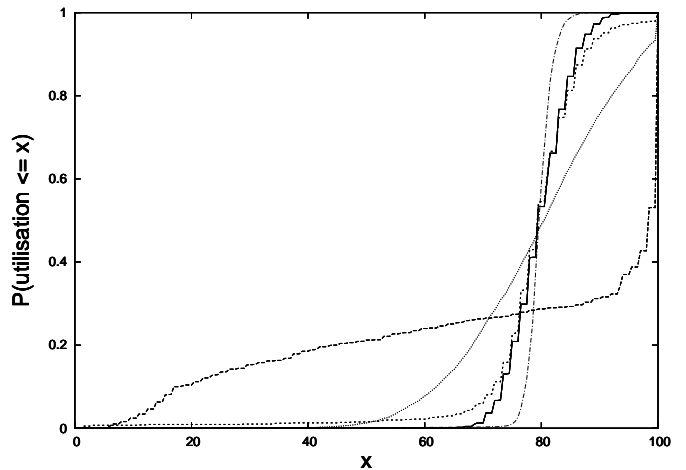


Fig. 8. Server utilization distribution of measurements from the experimental system,  $\{K, T_i\}=\{20, 2.8\}$  (full) and  $\{K, T_i\}=\{20, 0.1\}$  (dashed). Simulations of a controlled M/M/1-system (dotted) and a controlled M/D/1-system (dash-dotted) are also shown for comparison.

plotting the estimated distribution function. The distribution function is estimated from measurements during 1000 seconds with the specific parameter setting. The distribution function shows how well the control mechanism meets the first control objective. Second, we plot the step response during 60 seconds when starting with an empty system. The step response shows the settling time for the control mechanism.

### E. Distribution function

Fig. 8 shows the estimated distribution function when using the PI-controller. Both good and bad parameter settings were used to outline the differences in behavior and performance. An ideal admission control mechanism would show a distribution function that is zero until the desired load, and is one thereafter. In this case, the load was kept at 80%, and the parameter setting,  $\{K, T_i\}=\{20, 2.8\}$ , results in a controller that behaves very well in this sense. The parameter setting,  $\{K, T_i\}=\{20, 0.1\}$ , as can be seen, perform worse. Also, as comparison, results from simulations of a M/D/1-system and a M/M/1-system are given in Fig. 8, when using  $\{K, T_i\}=\{20, 2.8\}$ . The simulation results from these systems align well with the experimental results of the Apache server.

### F. Step response

Fig. 9 shows the behavior of the web server during the transient period. The measurements were made on an empty system that was exposed to 100 requests per second. The parameter setting,  $\{K, T_i\}=\{20, 2.8\}$ , exhibits a short settling time with a relatively steady server utilization. Comparisons to M/D/1 and M/M/1 simulations, also in Fig. 9, show that the model is accurate.

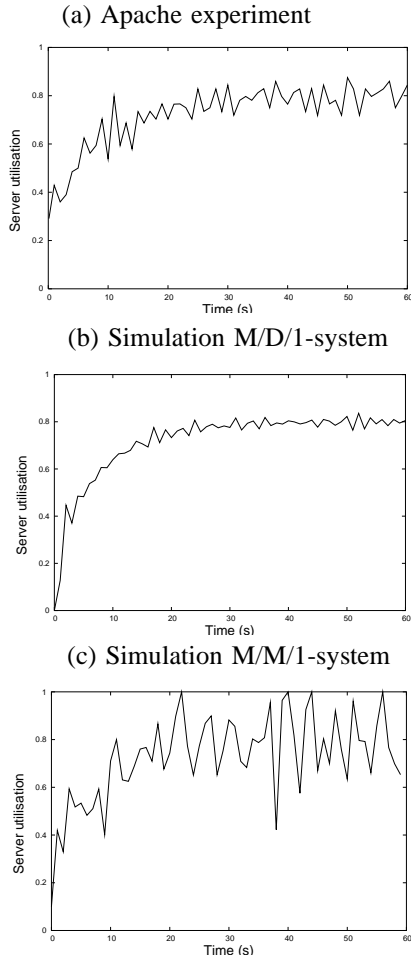


Fig. 9. (a) Step response experiment on the Apache server, (b) Simulation of M/D/1-system (c) Simulation of M/M/1-system  $\{K, T_i\} = \{20, 2.8\}$ ,  $\rho_{ref} = 0.8$ .

## V. DISCUSSION

The analysis in Section III-C 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. However, we can conclude that

- Pole-placement based on a linear model is OK in a restricted area (region A1 in Fig. 5).
- There are choices of parameters that gives stable closed loop poles, but where the linear analysis would indicate an unstable closed loop systems.

During simulation studies the dominant nonlinear effect has come from the queue nonlinearity  $\phi$ . The saturation due to limited arrival rate can be handled with a standard implementation of an anti-reset windup scheme, see [9].

## VI. CONCLUSIONS

Traditionally, queuing theory has been used when investigating server systems. However, within queuing theory

there are few mathematical tools for design and stability analysis of, for instance, admission control mechanisms. Therefore, these mechanisms have mostly been developed with empirical methods. In this paper, we have designed load control mechanisms for a web-server system with control theoretic methods and analyzed its stability properties. The controller structure considered is a PI-controller and a region for stabilizing control parameters is presented.

The designs have been experimentally verified with simulations and experiments on an Apache web-server system.

## ACKNOWLEDGMENTS

The work in this paper has partially been supported by the Swedish Research Council through the Multi Project Grant 621-2001-3020 and contract 621-2001-3053.

## REFERENCES

- [1] U. Körner and C. Nyberg, "Overload control in communication networks," in *Proc. of Globecom 91*, 1991, pp. 1331–1335.
- [2] T. Abdelzaher and C. Lu, "Modeling and performance control of Internet servers," in *Proc. of the 39th IEEE Conference on Decision and Control*, 2000, pp. 2234–2239.
- [3] T. Abdelzaher, K. Shin, and N. Bhatti, "Performance guarantees for web server end-systems: A control theoretic approach," *IEEE Transactions on Parallel and Distributed System*, vol. 13, no. 1, pp. 80–96, Jan 2002.
- [4] C. Lu, T. Abdelzaher, J. Stankovic, and S. Son, "A feedback control approach for guaranteeing relative delays in web servers," in *Proc. of the 7th IEEE Real-Time Technology and Applications Symposium*, 2001, pp. 51–62.
- [5] J. Kuri and A. Kumar, "Optimal control of arrivals to queues with delayed queue length information," *IEEE Transactions on Automatic Control*, vol. 40, no. 8, pp. 1444–1450, Aug 1995.
- [6] L. Sha, X. Liu, Y. Lu, and T. Abdelzaher, "Queueing model based network server performance control," in *Proc. of 23rd Real-Time Systems Symposium (RTSS 2002)*, 81–90.
- [7] N. Gandhi, D. Tilbury, Y. Diao, J. Hellerstein, and S. Parekh, "MIMO control of an Apache web server: modeling and controller design," in *Proc. of American Control Conference 2002*, vol. 6, 2002.
- [8] M. Kihl, A. Robertsson, and B. Wittenmark, "Analysis of admission control mechanisms using non-linear control theory," in *Proc. of IEEE International Symposium on Computer Communications*, 2003.
- [9] A. Robertsson, B. Wittenmark, and M. Kihl, "Analysis and design of admission control in web-server systems," in *Proceedings of ACC'03*, 2003.
- [10] C. Agnew, "Dynamic modeling and control of congestion-prone systems," *Operations Research*, vol. 24, no. 3, pp. 400–419, 1976.
- [11] M. Kihl, A. Robertsson, and B. Wittenmark, "Performance modelling and control of server systems using non-linear control theory." Berlin, Germany: 18th International Teletraffic Congress, Sept. 2003.
- [12] Y. Z. Tsytkin, "Frequency criteria for the absolute stability of nonlinear sampled-data systems," *Automation and Remote Control*, vol. 25, no. 3, pp. 261–267, 1964.
- [13] —, "Absolute stability of a class of nonlinear automatic sampled data systems," *Automation and Remote Control*, vol. 25, no. 7, pp. 918–923, 1964.
- [14] E. J. EI and B. Lee, "On the stability of a class of nonlinear sampled-data systems," *IEEE Transactions on Automatic Control*, vol. 9, no. 51, pp. 51–61, 1964.
- [15] K. Åström and B. Wittenmark, *Computer-controlled systems, theory and design*, 2nd ed. Prentice Hall International Editions, 1990.
- [16] M. Larsen and P. V. Kokotovic, "A brief look at the Tsytkin criterion: from analysis to design," *Int J. of Adaptive Control and Signal Processing*, vol. 15, no. 2, pp. 121–128, 2001.
- [17] "Apache web server," <http://www.apache.org>.
- [18] G. Banga and P. Druschel, "Measuring the capacity of a web server," in *USENIX Symposium on Internet Technologies and Systems*, December 1997, pp. 61–71.