

Abstracts of Dissertations

Institute of Information and
Communication Technologies

BULGARIAN ACADEMY OF
SCIENCES



7 / 2017



APPLICATION OF THE
PRINCIPLES OF THE
CONTROL THEORY TO
PROGRAM SYSTEMS

Stanislav Dimitrov

ПРИЛОЖЕНИЕ НА
ЗАКОНИТЕ НА ТЕОРИЯТА
НА УПРАВЛЕНИЕТО В
ПРОГРАМНИ СИСТЕМИ

Станислав Димитров

Автореферати на дисертации

Институт по информационни и
комуникационни технологии

БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ

ISSN: 1314-6351

Поредицата „Автореферати на дисертации на Института по информационни и комуникационни технологии при Българската академия на науките“ представя в електронен формат автореферати на дисертации за получаване на научната степен „Доктор на науките“ или на образователната и научната степен „Доктор“, защитени в Института по информационни и комуникационни технологии при Българската академия на науките. Представените трудове отразяват нови научни и научно-приложни приноси в редица области на информационните и комуникационните технологии като Компютърни мрежи и архитектури, Паралелни алгоритми, Научни пресмятания, Лингвистично моделиране, Математически методи за обработка на сензорна информация, Информационни технологии в сигурността, Технологии за управление и обработка на знания, Грид-технологии и приложения, Оптимизация и вземане на решения, Обработка на сигнали и разпознаване на образи, Интелигентни системи, Информационни процеси и системи, Вградени интелигентни технологии, Йерархични системи, Комуникационни системи и услуги и др.

Редактори

Генадий Агре

Институт по информационни и комуникационни технологии, Българска академия на науките
E-mail: agre@iinf.bas.bg

Райна Георгиева

Институт по информационни и комуникационни технологии, Българска академия на науките
E-mail: rayna@parallel.bas.bg

Даниела Борисова

Институт по информационни и комуникационни технологии, Българска академия на науките
E-mail: dborissova@iit.bas.bg

Настоящото издание е обект на авторско право. Всички права са запазени при превод, разпечатване, използване на илюстрации, цитирания, разпространение, възпроизвеждане на микрофилми или по други начини, както и съхранение в бази от данни на всички или част от материалите в настоящето издание. Копирането на изданието или на част от съдържанието му е разрешено само със съгласието на авторите и/или редакторите

*The series **Abstracts of Dissertations of the Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences** presents in an electronic format the abstracts of Doctor of Sciences and PhD dissertations defended in the Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences. The studies provide new original results in such areas of Information and Communication Technologies as Computer Networks and Architectures, Parallel Algorithms, Scientific Computations, Linguistic Modelling, Mathematical Methods for Sensor Data Processing, Information Technologies for Security, Technologies for Knowledge management and processing, Grid Technologies and Applications, Optimization and Decision Making, Signal Processing and Pattern Recognition, Information Processing and Systems, Intelligent Systems, Embedded Intelligent Technologies, Hierarchical Systems, Communication Systems and Services, etc.*

Editors

Gennady Agre

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
E-mail: agre@iinf.bas.bg

Rayna Georgieva

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
E-mail: rayna@parallel.bas.bg

Daniela Borissova

Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
E-mail: dborissova@iit.bas.bg

This work is subjected to copyright. All rights are reserved, whether the whole or part of the materials is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in other ways, and storage in data banks. Duplication of this work or part thereof is only permitted under the provisions of the authors and/or editor.



BULGARIAN ACADEMY OF SCIENCES

Abstract of PhD Thesis

APPLICATION OF THE PRINCIPLES OF THE CONTROL THEORY TO PROGRAM SYSTEMS

Stanislav Dimitrov

Supervisor: Prof. Todor Stoilov

Approved by Supervising Committee:

Prof. Grisha Spasov
Prof. Kosta Boshnakov
Prof. Idilia Bachkova
Assoc. Prof. Daniela Borisova
Prof. Todor Stoilov



**INSTITUTE OF INFORMATION AND
COMMUNICATION TECHNOLOGIES**

Department of Hierarchical Systems

Application of the principles of control theory to program systems

Stanislav Dimitrov

Topicality and significance of the dissertation

The intensive use of computer and communication systems causes lack of guaranteed time for serving the assigned tasks and makes static systems more and more inapplicable. This requires the designing and the implementation of techniques, which allow for dynamics and flexibility as to the changes in the workload and the resource management.

Approaches based on Automatic control theory for modelling, analyzing and designing of embedded computers and communication systems represents an interest for the specialists and a promising platform for controlling the changeability of large and complex real time systems. There are a lot of studies on feedback systems, which are used for adapting of calculations in real time, for guaranteeing the performance in cases of emerging disturbances and for resource planning. The models from Automatic control theory are being applied for describing the behavior of the systems, for controlling the response time, the optimality, the sustainability and the adaptation of the control of real systems.

The wide use of different computer configurations for access to various websites for the purposes of using web services generates highly burdened network traffic in the devices connecting the clients with the service providers. This in its turn loads the components of the machine that are responsible for maintaining the web pages. The server is a system in which a myriad of tasks are taking place simultaneously. Its administration constitutes a complex process of control. In view of Automatic control theory the link between the assigned tasks by the users and the usability of the hardware components represents an interesting research question, one that is being addressed in the current dissertation.

Aim and tasks of the research

The aim of the dissertation research is to develop and apply control models for the functioning of complex program systems, taking cue from control theory.

In order to achieve that goal, the study pursues the following tasks:

- To develop a functionality model of a computer system as to different input influences;
- To exert control of the studied computer system and to synthesize control through settings in the TCP protocol, the Apache web server and the file system;

- To apply system identification methods and to outline the dynamic features of the computer system in case of controlling the web server program;
- To study the effectiveness of the indicators for control quality of the system;
- To synthesize control controllers for the examined system and to assess the qualities of the open and the closed control system.

Methodology

In the pursuit of the above outlined tasks, the research uses main principles of Automatic control theory, and unfolds in the following steps:

- Generating input signals through user simulation of an information system with the aim of obtaining output signals for the work of the system (indicators of performance);
- Applying identification methods (approximation of transient response) with the aim of gathering data for the performance of the program system;
- Using statistical methods (regression analysis) to analyze the measured output data;
- Applying feedback control to the program system in order to explore the possibilities for improving the service provision for the users of the information system;
- Applying the method of control charts in measuring the quality of the performance indicators of the program system;
- Applying nonlinear model (Autoregressive with exogenous terms – ARX) to describe the behavior of the system during improvement of the customer service of the information system.

In the course of the experimental work there were used:

- Generators of input signals Jmeter и Apache Bench;
- SAR instrument for observation of the Linux operation system;
- MATLAB – for the statistical analyses of data and eliciting dynamic modes, for simulating nonlinear models and for setting the controllers of the computer system;
- The Linux’s PS program and AWK programming language for the feedback control;

The information system is realized through open source software products such as PHP, MySQL.

Resume of the PhD thesis

I. Program Systems and Automatic Control Theory

Chapter one of the PhD thesis deals with main assumptions of Automatic control theory (ACT), outlines the principles for building automation systems and pays attention to the regularities typical for the undergoing processes in them. ACT’s approaches are widely used in the modelling, analysing and designing of complex computer and communication systems. This first part of the text introduces the application of feedback control with the aim of adapting the

calculations, guaranteeing the performance and planning the resources. In order to implement ACT in software applications, which is actually the interest of the doctoral thesis, it is necessary to examine their properties and features. That is why this chapter also focuses on the components and the processes undergoing in the program system – the web server Apache and the operation system Linux.

The examination of the software system can be done in several ways: by performance assessment, by measuring the parameters of the system, by experimental work, simulations and analytical models. The performance of the system is determined most effectively by the means of two or more of the outlined approaches [BoGMT-05]. The modelling of the performance is a main realm of exploration when it comes to web servers. An accurate model represents a prognosis as to the performance indicators and represents a basis for planning of the necessary capacity of the performance resources with regard to different settings. In order to initiate more profound examination of the web servers and the computer systems, Queue networks (QN) are used as well as extended queue networks.

In the course of the overview in chapter one it was realized that using queue systems for modelling the performance of computer systems provides an excellent way of examining the various problems in the latter, by regarding them as network of queues and servers, and of effective modelling of the behaviour of the computer systems in a static condition. Nevertheless, in the attempts to model the dynamics of computer systems one encounters a variety of difficulties. This makes necessary the application of linear differential equations for the description of the computer systems' dynamics. They are also a good Automatic control theory tool for that particular purpose. Linear differential equations are extremely useful in building models through statistical or input-output data. This approach in exploring the systems is also at use in the System identification theory. In the framework of the dissertation research our understanding about their advantages and limitations will be formed by the means of generating input signals towards the system and by the means of software tools for measurement, observation and diagnostics of the computer system. This is a necessary step towards one of the main goals of the PhD thesis – applying the principles of Automatic control theory for obtaining adequate information for the performance of the program system.

II. Modelling a Computer System at Different Input Influences

Chapter two of the PhD thesis presents ways for analysing computer systems, which can be helpful in the building of models. It points out several factors that can influence the performance of the system and outlines the mere structure of the computer system that is used in the experimental work. The chapter describes various methodologies for examining the performance of the computer systems and explores possibilities for observation of the software and the hardware components of the latter. An analysis of the programs for generating input signals is made and a scenario for their application is provided. As a result of that, a functionality model for the performance of the information system is built for the purposes of the experimental work.

For building an automated control system it is necessary to conduct experimental work and create mathematical models of the object. Experiments of the computer system involve workload analysis and resource analysis with the aim of determining the model of the examined object – the Apache web server. The computer system at use in the experimental work functions with Linux operation system and the installed on it Apache web server. Together, they ensure the customer service of a web application for accessing issues of a scientific journal. Through measurements of the workload of the processes in Apache, control of the system is realized, one in which the settings of Apache are being preconfigured. Structural scheme of the examined system is presented below (fig. II. 1).



Fig. II.1. Structural scheme of the examined system

Initiating a research with various number of users aims to help in building a model of the parameters, measuring the performance of the system. In order to obtain that model of the information system it is necessary to make experiments, which load the server by simulating many users who are using its services simultaneously. In this way it could be modelled the functions performed by the system such as operations or processes, and it could be determined the reaction of the system manifested in response time, sustainability, reliability and scalability. The model ensures that the information system will perform successfully once being activated for exploitation by real users.

For the realization of the model the module UltimateThreadGroup of the Jmeter program has been used. It allows for planning of infinite number of records with users of the system during the experimental work. During the latter was simulated a usage of the server by 110 clients, which started sending requests at different time intervals. In the first 60 sec. the provided services are used by 10 clients, in the next 60 sec. are added 10 more clients and so on and so forth (every minute 10 more clients are added).

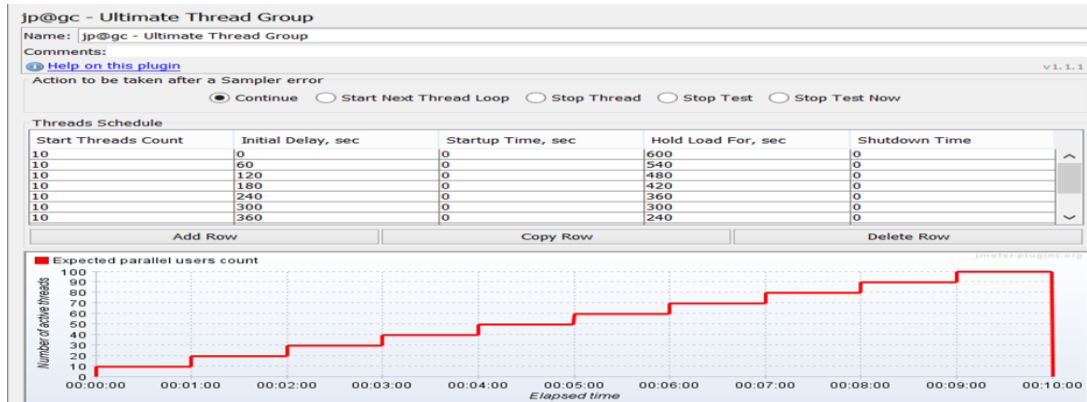


Fig.II.7. Setting users through UltimateThreadGroup

Main indicators of the performance of the system (Throughput, Response time and Latency) have been measured in order to model its capabilities. They are being measured through the *listener* elements of the Jmeter program, which keeps the results of the http request in files and presents a visual model of the results [ApaSF-16]. The results show that when the number of users increases the response time and latency increase as well.

The options of the Jmeter program allow for modelling of the functional capabilities of the system by examining the developed application and the hardware that supports it. This is a good way to analyse rapidly and easily a big quantity of data related to the generated workload towards the web server and to build a mathematical model for the changes in the parameters of the performance of the system. The functional model of the system gives room to explore changes in the parameters for performance in view of its sustainability and reliability.

III. Implementation of Control in the Studied Computer System

Chapter three of the PhD thesis presents the hardware and the software features of the computer system. The text focuses on the information system that has been used for examining the performance of the program system. It applies control through setting of the communication protocol TCP and of the program system – the Apache web server. The chapter describes the program for automatic setting of the Apache's configuration file and presents the effectiveness of the made improvements through the parameters for quality of the computer system (response time and throughput).

Implementing control in the studied computer system has two main goals: 1/ to demonstrate that the created information system for accessing scientific journals provides a modern, easy and functional way to reach materials distributed by the John Atanasoff Union of Automation and Informatics; 2/ to prove that the change in the parameters of the computer system, including the parameters of the Apache configuration file and the TCP communication protocol, will help in ensuring a simultaneous access to the system by more users.

The chapter starts with presenting some results as to the economic efficiency of the proposed solution by making a comparison of the costs related to paper-based and electronic-based distribution of journal “Automation and Informatics”. It is estimated that the latter will be 17 times less expensive than the former.

Then the text continues with presenting further results as to the performance of the program system. It analyses the workload that the server can handle and estimates the workload of the hardware resource. Examining the performance allows to measure the workload, done by the computer system and to reveal the capabilities of the developed software and the machine that supports it.

In the dissertation text the program system is presented as automated control system. The process of examining the performance involves the generation of input signals in the form of http requests by the Jmeter program towards the Apache web server, which supports the information system. Apache HTTP Server is presented as an automated control system, whose parameters of performance are being tested – these are throughput and response time in given maximum number of connections (determined by the parameter KeepAlive) between the server and the client while only certain number of clients are being allowed (through the MaxClient parameter). This method allows to regard the system as stabilizing at given permanent values of the configuration parameters of Apache KeepAlive и MaxClients.



Фиг. III.2. Apache HTTP Server, presented as an automated control system

The data from the conducted experiments shows that when it comes to the two input parameters – Response Time and Throughput, the automated control system is stable and reliable. They also show that when KeepAlive и MaxClients parameters are given constant values, the Apache web server regulates itself very well.

Then the chapter continues with some results of a study, through inputs towards the program system, conducted for video files. In this case the Apachebench generator of static pages is used. It allows generating and sending of requests with different sequence and size. In terms of ACT these represent single inputs with different amplitude. The aim of the experiment is to compare the effect of the default Apache web server configurations with the proposed in this dissertation work improved settings. This is done by examining the behaviour of the hardware components – the processor and the memory during stressing the server with various by size and mode of sending requests for a video file. The results also take into account switching on the parameter mod_deflate of Apache and its influence on the performance of the hardware components. The resulting data shows that that by using the proposed settings the usage of processor and memory are reduced with 2,37%, for processor and with 8,93% for memory respectively.

Then the text continues with presenting some results with regard to different types of files. The requests are sent with constant size and through an impulse in a particular moment of time while the behaviour of the parameters of performance are being monitored – response time and number of requests in the receiving queue of the web server. The results show that when the requests are being sent with the same size of the impulse towards the program system, the response time is constant, while sending a single impulse towards the server, the response time is significantly increasing. This is a reason to look for solutions for decreasing the response time of the system.

The computer system can be presented as a sequence of subsystems, working in synchrony. It includes the subsystem of the TCP protocol, the subsystem of the http protocol and the subsystem of the input/output data. The parameters of these subsystems need additional tuning for the specific hardware configuration, on the basis of which they work [MDVHR-01]. In the PhD thesis this additional change is done for the parameters of the Linux core and more specifically for the network settings and the TCP communication protocol. The communication between the end users and the server goes through three zones – the program system (Apache web server); the computer system (which includes the operation system and the TCP settings; and the network path. The doctoral study focuses on control of the program system, i.e. the web server as well as one in the second zone – the computer system, where the settings of the TCP protocol are improved and where installation of programs for influencing the performance of the operation system is taking place.

The effectiveness of the applied changes in the Apache configuration file is tested through the indicators for the performance of the system – throughput and response time in three cases: with default settings and two with reconfigured settings of the configuration file of Apache - (“bestSettings” and “highSettings”). It was revealed that the response time in the first case is longer in comparison with the other two cases. The suggested two variations for reconfiguring the settings of Apache lead to faster servicing of the clients’ requests. It was also found that the throughput is worse with the default settings rather than with the two cases with reconfigured settings. With “highSettings” the throughput is better when less than 20 clients are being serviced simultaneously.

In the measuring of the stress on the hardware components it was discovered that in sending an impulse we can observe bigger and more frequent fluctuations in the usage of the processor and memory no matter the file type. These disturbances show that it is necessary to introduce a change in the configuration parameters of the Apache web server and the TCP communication protocol in order to alleviate the work of the system.

Automatic control with the reconfiguring of Apache aims faster processing of the requests in view of the number of the clients. From the obtained results we could make the conclusion that the program for automatic reconfiguring of Apache needs to use the “highSettings” when the information system is used by up to 20 clients and “bestSettings” when the clients are more than

20. The Linux program PS and the language AWK are used in the program code for the reconfiguring of Apache.

The control program is of the following type:

```
#!/bin/bash
#!/bin/sed
while :
do
for i in $( ps aux | grep opt/lampp/bin/httpd |awk '{s+=$3} END {print s}' );
do
if [ $i \> 3.9 ]
then
cp httpd-mpm_2.conf httpd-mpm.conf
cp httpd-default_2.conf httpd-default.conf
sudo /opt/lampp/lampp reloadapache
echo "Load new settings!"
elif [ $i \< 1.00 ]
then
cp httpd-mpm_1.conf httpd-mpm.conf
cp httpd-default_1.conf httpd-default.conf
sudo /opt/lampp/lampp reloadapache
echo "Load settings by default!"
fi
done
sleep 20
done
```

One of the main tasks of the study is to examine the effectiveness of the implemented reconfiguration settings. In order to better present the results of the improvements in the system with regard to the performance of the computer system in servicing requests as to databases and video files, an experiment with a different from the original scenario was initiated. In the case of databases it becomes clear that the average value of the usage of the processor with the default settings is 3,53 %, and with the reconfigured settings of the system it is 5,68 %; in the case of video files – the average usage of the processor has decreased in the three examined time intervals respectively with 38,20%, 29,91,% and 41,27%.

The value of the proposed improvements of the applied settings of the TCP protocol and the Apache configuration file are confirmed also by the data on the response time – it decreases approximately 10 times with the newly introduced configuration, which is a significant contribution of the study.

By comparing the results from experiments with databases and video files it becomes clear that reconfiguring the TCP protocol and the Apache configuration file lead to improvement in the processor usage and to decrease in the response time of the system. The proposed in the dissertation work changes in the settings of the TCP protocol and the Apache configuration file represent an effective solution for the developed information system which stores databases with scientific journals and video files from seminars, conferences and workshops.

In conclusion, chapter three of the dissertation text demonstrates the applicability of ACT for improving the efficiency of the program systems manifested in optimized performance. The proposed solutions for changes in the TCP protocol and the Apache configuration file result in improvement in the indicators for quality of the performance – usage of the processor, usage of memory and response time.

IV. Application of identification methods from automated control theory for improving the performance of the computer system

The last chapter of the PhD study focuses on the application of the identification methods for a particular computer system. This aims at determining the dynamic characteristics of the system by the parameters of its performance. The presented features are being approximated and models of the system are being built. Their efficiency is examined by the method of the control charts for determining the applicability of the models. The implemented software improvements in the examined system are modelled by the means of nonlinear models, which are flexible with regard to complex phenomena and processes. The developed models are simulated in the simulation environment of Matlab with the aim of exploring the behaviour of the system when a proportional-integral controller for controlling the usage of the processor is on. With the use of a proportional controller in the simulation scheme is demonstrated the efficiency of the applied software control.

Chapter four of the text focuses on the applicability of nonlinear ARX models in order to present the implemented software improvement and to explore the possibility for additional control through proportional-integral controllers. The experimental work shows that in the case of video files it is more appropriate to initiate a proportional-integral control of the used system. In contradistinction to that, in the case of servicing clients requesting databases from the information system it turns out that when their number exceeds 5 the usage of proportional-integral control is not advisable.

Contributions of the PhD thesis

1. It develops a quantitative model by applying the formal apparatus of automated control theory for the control of complex program systems.
2. It implements control in the examined programmed system by setting the parameters of the TCP communication protocol, the Apache web server and the Linux operation system.

3. It determines experimentally the dynamic features of the program system during its work.
4. It determines the efficiency of the applied settings through the indicators of quality of the system.
5. It applies settings for proportional controllers in order to demonstrate the implemented improvements in the work of the program system and with the help of proportional-integral action to achieve better quality of control within the closed loop.

Literature

[ApaSF-16]	Apache Software Foundation, Apache Jmeter – User’s Manual, Building a web test plan, 2016, http://jmeter.Apache.org/usermanual/build-web-test-plan
[BoGMT-05]	G. Bolch, S. Greiner, H. de Meer and K. S. Trivedi, Queueing networks and Markov chains: Modeling and Performance Evaluation with Computer Science Applications, Wiley-Interscience, 2005.
[MDVHR-01]	R. D. Van der Mei, R. Hariharan and P. Reeser,” Web Server Performance Modeling, “Telecommunication Systems, Vol. 16, No. 3-4, 2001, pp. 361-378. doi:10.1023/A:1016667027983



АВТОРЕФЕРАТ НА ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен “доктор” по
научна специалност “Информатика“

ПРИЛОЖЕНИЕ НА ЗАКОНИТЕ НА ТЕОРИЯТА НА УПРАВЛЕНИЕТО В ПРОГРАМНИ СИСТЕМИ

Станислав Димитров Димитров

Ръководител: Проф. Тодор Стоилов

Научно жури:

Проф. Гриша Спасов.
Проф. Коста Бошнаков
Проф. Идилия Бачкова
Доц. Даниела Борисова
Проф. Тодор Стоилов



Дисертационният труд е обсъден и допуснат до защита на разширено заседание на секция „Йерархични системи“ на г. в Института по информационни и комуникационни технологии на БАН.

Дисертационният труд обхваща 159 страници, в него са включени 77 фигури, 9 таблици и са посочени 115 литературни източника.

Защитата на дисертацията ще се състои на от часа в зала на блок 2 на Института по информационни и комуникационни технологии на БАН на открито заседание на научно жури в състав:

1. проф. д-р Гриша Спасов
2. проф. д-р Коста Бошнаков
3. проф. д-р Идилия Бачкова
4. доц. д.т.н Даниела Борисова
5. проф. д.т.н Тодор Стоилов

Материалите за защитата на разположение в стая на Института по информационни и комуникационни технологии на БАН, ул „Акад Г. Бончев“ , бл 2.

Обща характеристика на дисертационния труд

Актуалност и значимост на изследването

Интензивното използване на компютърните и комуникационни системи причинява липса на гарантирано време за обслужване на поставените им задачи и прави статичните системи все по-неприложими. Това изисква да се проектират и прилагат техники, които позволяват динамика и гъвкавост по отношение на промени в натоварването и използването на ресурсите.

Подходи, основани на теория на автоматичното управление, за моделиране, анализиране и проектиране на вградени компютри и комуникационни системи представляват интерес за специалистите като обещаваща платформа за контролиране на променливостта на големи и сложни системи, работещи в реално време. Провеждат се редица изследвания на системи с обратна връзка, които се използват за адаптиране на изчисления в реално време, гарантиране на производителността при възникнали смущения и планиране на ресурсите. Моделите от теорията на автоматичното управление се прилагат за описание на поведението на системите, контролиране на времето за реакция, оптималността, устойчивостта и адаптацията на управлението на реални системи.

Широкото използване на различни компютърни конфигурации за достъп до разнообразни уеб сайтове, с цел използване на уеб услуги, генерира изключително натоварен мрежови трафик в устройствата, свързващи клиентите и доставчиците на услуги. Това на свой ред натоварва компонентите, изграждащи машината, на която се поддържат интернет-страниците. Сървърът е система, в която протичат много задачи едновременно. Неговото обслужване представлява сложен процес на управление. От гледна точка на теорията на автоматичното управление, връзката между зададените задачи от потребителите и използваемостта на хардуерните компоненти представлява интересен изследователски въпрос, разгледан в настоящата дисертация.

Цели и задачи на дисертационния труд

Целта на изследването в дисертационния труд е да се разработят и приложат модели за управление от теория на управлението при функционирането на сложни програмни системи.

За постигането на целта на дисертационната работа се дефинират и решават следните задачи:

- ✓ Да се разработи функционален модел на компютърната система при различни входни въздействия.
- ✓ Да се реализира управление в изследваната компютърна система и се синтезира управление чрез настройка на TCP протокола, Apache уеб сървър и файловата система.
- ✓ Да се приложат методите за идентификация на системи и да се определят динамичните характеристики на компютърната система при управление на програмата за уеб сървър.

- ✓ Да се изследва ефективността на показателите за качеството на управление на системата.
- ✓ Да се синтезират регулатори за управление на изследваната система и да се оценят качествата на отворена и затворена система за управление.

Методология на изследването

В изпълнение на горепоставените задачи са използвани основни положения от теория на автоматичното управление (ТАУ), реализирани в следните стъпки:

- Генериране на входящи въздействия чрез симулиране на потребители на информационна система с цел получаване на изходящи сигнали за работата на системата (показатели на производителността);
- Прилагане на методи за идентификация (апроксимиране на преходни характеристики) с цел събиране на данни за работата на програмната система;
- Използване на статистически методи (регресионен анализ) за анализ на измерените изходни данни;
- Прилагане на управление с обратна връзка (feedback control) на програмна система с цел проверка на възможностите за подобряване обслужването на потребители на информационната система;
- Прилагане метода на контролните карти при измерване качеството на показателите на производителността на програмната система;
- Прилагане на нелинеен модел (Autoregressive with exogenous terms – ARX) за описание поведението на системата при подобрене на обслужването на потребителите на информационната система.

В хода на експерименталната работа са използвани:

- Генератори на входни въздействия Jmeter и Apache Bench;
- Инструмент SAR за наблюдение на операционната система Linux;
- MATLAB – за статистически анализ на данни и извеждане на динамични модели, за реализиране на симулация на нелинейни модели и за настройка на регулаторите на компютърната система;
- Linux програмата PS и език за програмиране AWK за реализиране на управлението за обратна връзка.

Информационната система е реализирана чрез софтуерни продукти от тип “open source”, напр. програмен език PHP, MySQL.

Съдържание на дисертационния труд

Глава 1. Програмни системи и теория на автоматичното управление

Първа глава на дисертационния труд представя основни формализми от теория на автоматичното управление, които са използвани за основа на експерименталната работа и предлага обзор на програмни системи, при които е приложено управление с обратна връзка. Тази част от текста описва софтуерните обекти, използвани за извършване на проведените изследвания. Разгледани са и начините за моделиране на компютърни системи и са представени примери за измерване на производителността на уеб сървъри.

В първа част на главата се описва приложимостта на теорията на автоматичното управление, както и структурата и характеристиките на системите за автоматично управление. Представени са начините за управление в зависимост от събраната информация за системата и подходите за класифициране на системите за управление, които се основат на типа на модела, на характера на сигналите или на други признаци и свойства на системата. Обоснована е необходимостта от идентифициране и анализиране на изходните сигнали и от получаването на подходящо математическо описание – модел при прилагане на управление. Един от начините за получаване на математическо описание на системите (процеси, обекти) е чрез реализиране на експерименти, като по този начин се идентифицират характеристиките на поведението на изследваните системи.

Втората част на главата отбелязва примери за прилагане на принципа за управление с обратна връзка към избрания обект. Този вид управление се реализира и в различни софтуерни приложения за управление на бизнес функционалности, за управление на мрежови устройства, за управление на натоварването на клъстери от сървъри, за управлението на поточни медии и др. [DGHPT-02].

Третата част от главата представя софтуерните системи, които се използват за експерименталните цели на дисертацията – програмната система - уеб сървър Apache и операционната система Linux. Те позволяват да се осъществи качествен анализ за производителността на системата и да се изградят подходящи хипотези и изследвания за работещите процеси, планиране на нишките, ограничението на паметта и обработката на входно/изходните процеси. Производителността (Performance) е мярка за количеството работа, извършено от компютърната система. Ефективността на една компютърна система в зависимост от контекста, в който се използва, се определя от време за отговор, бързодействие, натоварване на хардуерните ресурси, достъпност до компютърната система или определено приложение, капацитет на използваната мрежата и др. [ArnoA-94]. Анализирането на дадена компютърна система се осъществява в услуга на различни цели, но предимно за анализиране на мащабируемостта, т.е. изменението на производителността при промяна в натоварването или изменението на използването на ресурса.

Изследването на софтуерната система може да се осъществи по няколко начина: чрез оценка на производителността, измерване на параметрите на системата, експериментални изследвания, симулации и аналитични модели. Производителността на системата най-

ефективно се определя чрез използване на два или повече от гореспоменатите подходи [BoGMT-05]. Моделирането на производителността е основна област на изследване при уеб сървърите. Точният модел представя прогноза за показатели на производителността и за планиране на необходимия капацитет на ресурсите за изпълнение при различни настройки. За изготвяне на по-задълбочени изследвания на уеб сървърите и компютърните системи се използват QN мрежа от опашки и разширени мрежи от опашки.

В обобщение, глава първа от дисертационния труд разглежда основни положения от теорията на автоматичното управление, спира се на принципите за изграждане на автоматичните системи, както и на закономерностите, характерни за протичащите в тях процеси. Нейните подходи намират все по-голямо приложение при моделирането, анализирането и проектирането на сложни компютърни и комуникационни системи. Тази част от текста представя приложението на управлението с обратна връзка с цел адаптиране на изчисленията, гарантиране на производителността и планиране на ресурсите. За прилагане на ТАУ в софтуерни приложения, какъвто е интересът на дисертацията, е необходимо да се познават техните свойства и характеристики. По тези причини са описани компонентите и процесите, протичащи в операционната система Linux и уеб сървъра Apache.

В глава първа бе установено, че използването на системи от опашки за моделиране на производителността на компютърни системи осигурява отличен начин за изследване на различни проблеми в последните, разглеждайки ги като мрежа от опашки и сървъри. Те ефективно моделират поведението на компютърните системи в установено състояние. Но трябва да се има предвид, че при опити за моделиране на динамиката на компютърните системи, тяхното използване създава редица затруднения. Това налага прилагането на линейни диференциални уравнения за описание динамиката на компютърните системи. Те са и метод за описание на системите, характерен за ТАУ. Линейните диференциални уравнения са изключително полезни при изграждането на модели чрез статистически данни или входно-изходни данни. Този подход за проучване на системите се разглежда и в теорията за идентификация на системи. В рамките на дисертационния труд разбирането на техните възможности и ограничения се търси чрез генериране на входни въздействия към системата и със софтуерни инструменти за измерване, наблюдение и диагностика на компютърната система. Това е необходима стъпка по посока на една от основните задачи на дисертацията – прилагане принципите на ТАУ с цел получаване на адекватна информация за работата на програмната система.

Глава 2. Моделиране на компютърна система при различни входни въздействия

Във втора глава на дисертационния труд са представени начини за анализ на компютърни системи, които да подпомогнат изграждането на модели. Отбелязани са някои фактори, които могат да влияят на производителността на системата. Тази част на текста представя структурата на използваната компютърна система за целите на експерименталната работа. Описани са методологиите за изследване производителността

на компютърните системи и са разгледани възможностите за наблюдение на софтуерните и хардуерни компоненти на последните. Направен е анализ на програмите за генериране на входни въздействия и е очертан сценарий на тяхното прилагане. В резултат на това е изграден функционален модел при използването на информационната система, който да послужи за целите на експерименталната част на настоящото дисертационно изследване.

Първата част на главата идентифицира предимствата от изграждането на модел на компютърната система – за анализ на хардуера и софтуера на използваната конфигурация, за планиране на капацитета и отстраняване на ограничения на системата, за анализиране и коригиране на мащабируемостта, и т.н. Получаването на модел е свързано с методи за анализ на производителността. Описани са тяхното приложение, действие и ефективност.

Във втората част на главата са представени някои фактори, които могат да влияят на производителността на компютърната система – планирани системни действия, други потребители или други натоварвания. Бе отбелязано, че е изключително трудно откриването на смущения в големи и сложни компютърни системи. Това е от значение за задачите на дисертацията, тъй като данните, които се използват за изграждане на модели, много често са зашумени.

В третата част на главата е представена използваната компютърна система. По своята същност тя е сложна система за управление, в която протичат много процеси едновременно. При нея определени процеси могат да се разглеждат като входни въздействия за системата, други – като смущения за системата, а трети – като изходни данни от нейната работа. Това позволява компютърната система да се представи като система за автоматично управление (САУ). Според теория на информацията САУ е информационна система, която преобразува съответната информация за постигането на определени цели като регулиране, следене и програмно управление [СапуГ-07].

За изграждането на система за автоматично управление е необходимо да се проведат експериментални изследвания и да се създаде математически модел на обекта. Експерименти за компютърната система може да се реализират посредством анализа на натоварването и анализа на ресурсите, за са се определи модела на изследвания обект – уеб сървър Apache. Използваната компютърна система за провеждане на експериментални изследвания в дисертационния труд работи под операционна система Linux и инсталиран на нея уеб сървър Apache. Те съвместно осигуряват обслужването на потребители на уеб приложение за достъп до научно списание. Чрез измерване на натоварването на процесите на Apache се реализира управление на системата, в което се преконфигурират настройките на Apache. Структурна схема на изследваната система е представена на фиг. П.4.



Фиг.П.4. Структурна схема на изследваната система

В четвъртата част на главата са представени методологиите за анализ на производителността. Те се разделят на няколко категории в зависимост от използваните продукти и подхода за наблюдение на показателите на производителността: липса на конкретни действия; задаване на въпроси; диагностичен цикъл; методи за изследване на ограниченията; USE метод; и др. [Gregg-13].

В петата част на главата са разгледани критериите за изграждане на програмни системи (време за отговор, производителност, цена) и основните настройки за ефективното им функциониране – настройки на софтуерно ниво (запис на файловата система, размер на блока), настройки на мрежово ниво (размер на буфера).

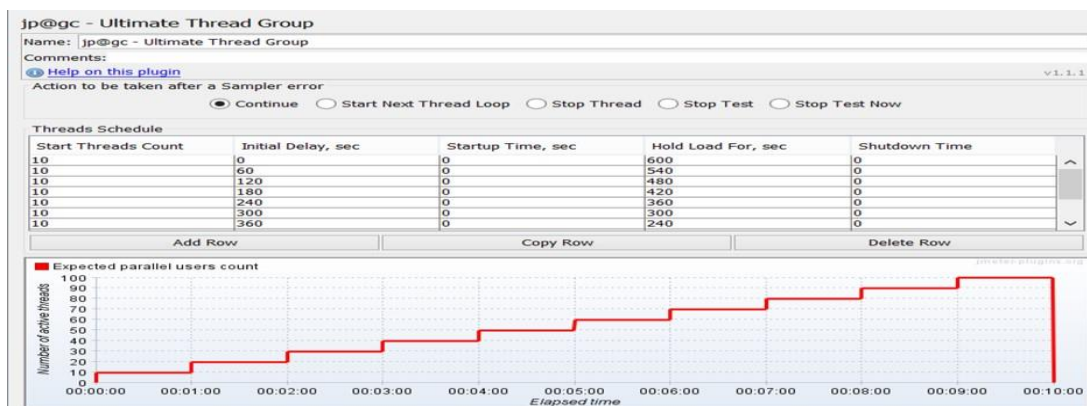
В шестата част от главата са описани начините за наблюдение на софтуерните и хардуерните компоненти под операционна система Linux – чрез броячи и чрез проследяване.

В седма част на глава първа са представени програми за генериране на входни въздействия. Чрез тях се събират данни за ефективността на компютърните системи. Те позволяват да се планират задачи за изпълнение от информационната система, да се приложат различни сценарии и конфигурации с цел събиране на данни от показатели на производителност и да се анализира тяхната надеждност. Този вид програми се използват от софтуерни дизайнери, администратори и обикновени потребители, но изискват познания за системата, която се изследва. Техните функционалности позволяват конфигуриране и генериране на различни входни въздействия, задълбочено наблюдение на поведението на услугите и предоставят показатели за сравняване и оценка на тяхното качество. Тези инструменти се използват, за да посочат недостатъците на системата при емулиране на стотици клиенти, позволяват бързо и ефективно реализиране на входни въздействия и на разпределение на заявките.

В тази част от текста се представя описание на сценария, по който са направени изследванията на информационната системата, използвана за експерименталните цели на дисертацията: клиентите зареждат сайта `hs22.iccs.bas.bg` и се насочват към страница `index.php`, след което потребители на информационната система разглеждат последователно страниците за новини (`news.php`), контакти (`contats.php`) и всички публикувани списания, които се разпространяват по електронен начин чрез посещаване на страницата за списанията (`all_papers.php`). Извършеният експеримент е с клиент, който вече е регистриран в системата, така че директно се използва страницата `login.php`, където се въвеждат потребителско име и парола. Проверката установява, че името и паролата съществуват в базата данни и това позволява на потребителя да извърши търсене в нея на целия текст на статията. Той търси две статии по два критерия. Първият е „име на автор“, а вторият е „година на публикуване“. След това потребителят прочита статиите в pdf. Всички тези действия, които се извършват от информационната система се използват, за да се симулират заявки от клиентите на системата в определен момент. По този начин се изследват възможностите на система за обслужване на потребителите на уеб базираното приложение за електронно разпространение на научно списание.

Осъществяване на проучване с различен брой потребители цели да се изгради модел на параметрите, измерващи производителността на системата. За получаването на модел на информационната система е необходимо да се извършат изследвания, при които се натоварва сървъра, като се симулират множество потребители, които потребяват неговите услуги едновременно. По този начин се моделират функциите, изпълнявани от системата като действия, операции или процеси, и се определят реакции на системата като бързодействие, устойчивост, надежност и мащабируемост. Моделът гарантира, че информационната система ще работи успешно при пускането ѝ в експлоатация за използване от реални клиенти.

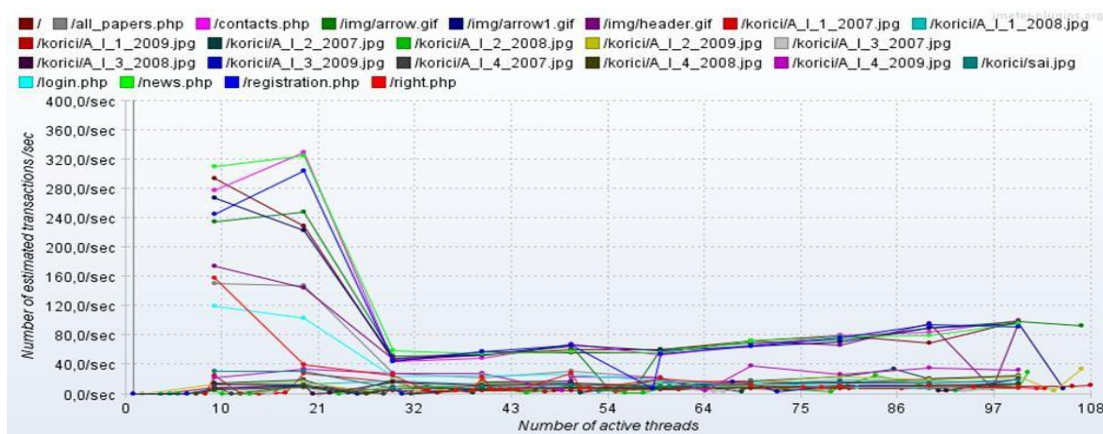
За реализиране на модел се използва модула UltimateThreadGroup на програмата Jmeter, който позволява да се планират безкраен брой записи с потребители на системата при провеждане на изследването. В експеримента се симулира използване на сървъра от 110 потребителя, които започват да изпращат заявки в различни времеви интервали. На фиг. II.7 е показано, че в първите 60 секунди услугите предлагани от него се използват от 10 потребителя в следващите 60 секунди се добавят нови 10 и така всяка минута се добавят по 10 потребителя.



Фиг. II.7. Настройване на потребителите чрез UltimateThreadGroup

Изследват се основни показатели за производителността на системата и се моделират нейните възможности. Те се измерват чрез listener елементите на програмата Jmeter, които съхраняват резултатите от http заявките във файлове и представят визуален модел на данните [АраSF-16].

Throughput измерва броя на операциите, извършени от системата повреме на изследването, и изразява капацитета на един сайт или една информационната система. Целта на неговото използване е да се определи броят на заявките за час/минута/секунда, с които може да се справи. Фигура II.8 показва статистически данни за максималния брой възможни транзакции (обслужени заявки от сървъра) на базата на броя на потребителите, имащи достъп до информационната система. Линиите на фигурата представят всяка страница на уеб приложението, която се изследва. Раздалечеността между линиите се получава, защото една част от страниците съдържат статични файлове - снимки, а друга част са файлове, които се генерират динамично чрез езика php. Фигурата представя ограничението на приложението, което е малък брой обслужени транзакции за статични файлове и необходимостта от промяна големината на снимките.



Фиг.П.8. Зависимост между броя транзакции и броя потребители

Следващият показател, който се моделира чрез програмата Jmeter, е времето за отговор – то показва средното време за обработка на заявките от сървъра в милисекунди. От направените експерименти става ясно, че времето за отговор се изменя в зависимост от изпратените заявки от потребителите, а за да се постигне определено време за отговор трябва се поставят ограничения в работата на системата.

Третият показател, който се моделира чрез Jmeter, е латентността на системата. Анализирано е времето, след което ще започнат да се обработват заявките в зависимост от броя потребители и е установено, че тя се увеличава при увеличаване на потребителските заявки. За да се обяснят промените в латентността и времето за отговор се измерват и хардуерните компоненти на системата – процесор, памет, хард диск и др. За тази цел се използва елементът PerfMonCollector, който позволява да се измерят приложените входни въздействия към сървъра и да се съберат статистически данни за процесора, паметта, swar и др. С негова помощ бе установено, че паметта се променя в зависимост от потребителите на приложението, а процесорът се натоварва константно през цялото време на изследването. Резултатите не показват пренатоварване на хардуерните ресурси и затова се изследват и други показатели на системата. На фиг.П.12 е показано натоварването на TCP комуникационния протокол и твърдия диск на системата. Тези измервания се извършват с цел да се проследи източника на увеличаването на времето за отговор и латентността.



Фиг. П.12 Натоварване на TCP комуникационния протокол и твърдия диск на системата

От представените резултати може да се направи заключение, че TCP комуникационния протокол е в пренатоварено състояние в последния времеви интервал на изследването на системата, което обяснява увеличаването на времето за отговор и

латентността ѝ. Става ясно, че неговото натоварване е пропорционално на броя потребители на системата. Синята линия на фигурата демонстрира, че натоварването на твърдия диск на системата е малко, което означава, че увеличаването на времето за отговор не е породено от него. Пиковите в нейното поведение се получават, когато процесите не могат да се обслужват от паметта и се предават към хард диска [GheGF-13].

Втора глава на дисертационния труд демонстрира, че опциите на програмата Jmeter позволят да се изгради модел на функционалните възможности на системата чрез изследване разработеното приложение и хардуера, който осигурява неговото поддържане. По този начин става възможно да се анализира бързо и лесно голямо количество данни от измервания на генерираното натоварване към уеб сървъра и да се изгради математически модел за промените в показателите за производителността на системата. Функционалният модел на системата дава възможност да се изследва изменението в показателите за производителност на системата по отношение на нейните устойчивост и надеждност.

Глава 3. Реализиране на управление в изследваната компютърна система

В трета глава на дисертационния труд са представени хардуерните и софтуерните характеристики на компютърната система. Текстът разглежда информационната система, която е използвана за изследване на производителността на програмната система.

Приложено е управление чрез настройка на комуникационния протокол TCP и на програмната система - уеб сървъра Apache. Описана е програмата за автоматично настройване на конфигурационния файл на Apache. Представена е ефективността на направените подобрения чрез показателите за качество на компютърната система (време за отговор и бързодействие).

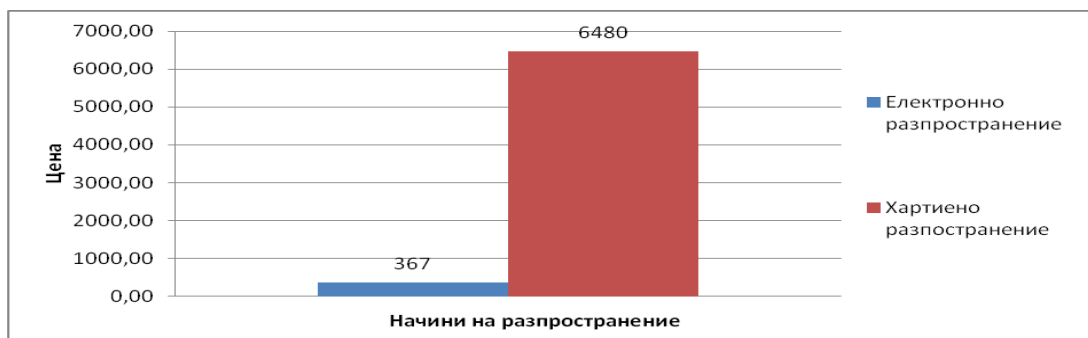
Реализирането на управление в изследваната компютърна система има две основни цели: 1/ да се покаже, че създадената информационна система за достъп до статиите на научно списание предоставя по-модерен, по-лесен и по-функционален начин за достъп и търсене в материалите, разпространявани от Съюза по автоматика и информатика; 2/ да докаже, че промяната на параметрите на компютърната система, в това число параметрите на конфигурационния файл на Apache и на комуникационния протокол TCP, ще подпомогне едновременния достъп на повече читатели до системата.

В първата част на главата са представени хардуерните и софтуерните характеристики на компютърната система, която се използва в дисертационния труд. Описана е информационната система в Интернет, която позволява селективно търсене, обработване и извличане на информация за статиите, публикувани в българско научно списание. Целта е да се създаде информационна услуга в Интернет за електронен абонамент на научно списание. Представено е конкретно решение, което използва трислоен модел на клиент/сървър. Създадената информационна система динамично генерира и предоставя информация [TricE-11]. Приложението е реализирано чрез няколко софтуерни модула, разработени посредством PHP (HypertextPreprocessor). Чрез тези модули се реализира SQL заявка (request) и SQL отговор (response), необходими за функционирането му. След

получаване на SQL заявката се претърсва базата данни MySQL, за да бъде извлечена желаната от потребителя информация. Данните от базата се обработват и структурират в HTML формат с цел отговаряне на SQL заявката като SQL отговор (response). Използваните продукти са от т.нар. свободен код (open source). Те са системно независими, поради което е възможно използването им при Windows-, Unix- и Linux- базирани машини.

Конкретната база данни, използвана в информационната система, съдържа четири отделни таблици. В тях има подробна информация за статиите, регистрираните потребители, потребителите с платен абонамент и направлението, в което са публикациите.

За да се демонстрира икономическата ефективност на решението е реализирано сравнение на електронното и хартиеното разпространение на списание „Автоматика и Информатика“. Разходите за електронно разпространение на списанието са изчислени въз основа на цената на електроенергията и консумацията на електроенергия от сървъра за една година. В разходите за хартиено разпространение са включени тираж на списанието за година (4 книжки годишно x 500 броя=2000 бр.) и цена на една книжка. Резултатите показват, че разпространението на списанието по електронен път е 17 пъти по-евтино от начина, използван в момента [Димит-13а].



Фиг. III.3. Сравнение на начините на разпространение на списание „Автоматика и Информатика“

Във втората част на главата се представят резултати от изследване производителността на програмната система. Анализирани са натоварванията, които уеб сървърът може да обслужи и е определено натоварването на хардуерния ресурс. Изследването на производителността позволява да се измери количеството работа, извършено от компютърната система, и да се разберат възможностите на разработения софтуер и на машината, която се използва за неговото поддържане.

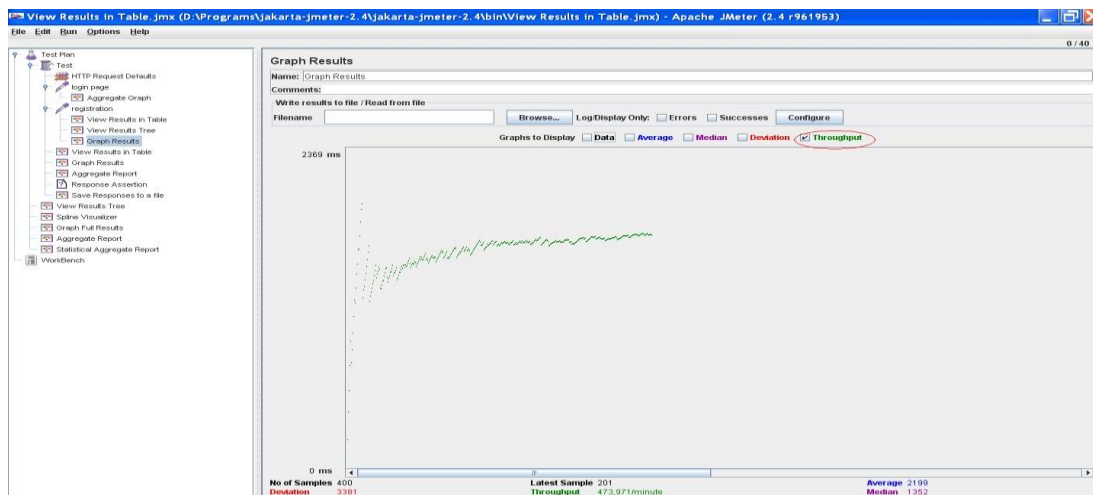
В дисертационния труд програмната система се представя като система за автоматично управление. Процесът по изследване на работоспособността се осъществява чрез генериране на входни въздействия във вид на http заявки от Jmeter към уеб сървъра Apache, който поддържа информационната система. Apache HTTP Server е представен като система за автоматично управление, на която са изследвани показателите на производителността – бързодействие и време за отговор, при зададени максимален брой свързвания (определяни от параметъра KeepAlive) между сървъра и потребител на системата, като се разрешава определен брой потребители да се свързват със сървъра (чрез параметъра MaxClients) (Фиг. III.4). По този начин се изследва системата като

стабилизираща при зададени постоянни стойности на конфигурационните параметри на Apache KeepAlive и MaxClients.



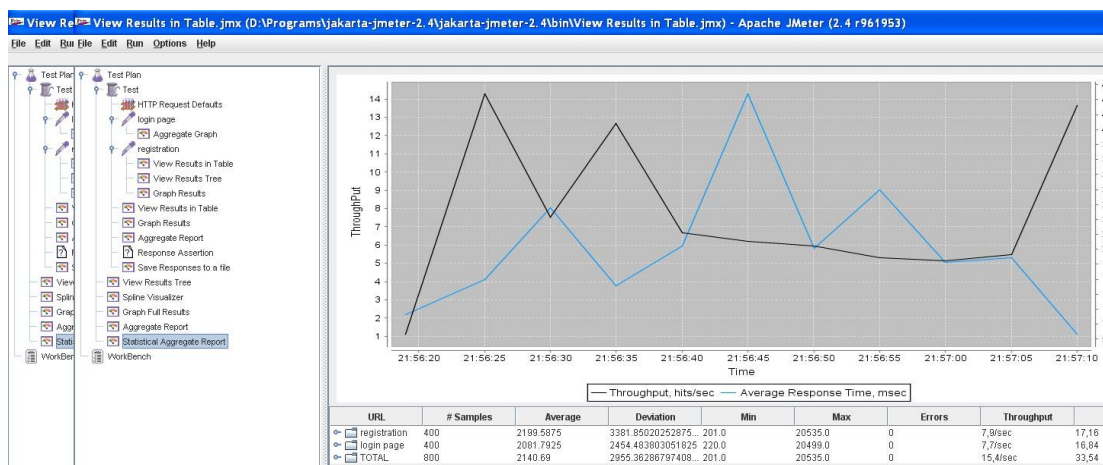
Фиг. III.4. Apache HTTP Server, представен като система за автоматично регулиране

В изследването се изпращат HTTP заявки към уеб сървъра Apache по метода Post чрез порт 80. Проследява се поведението на сървъра при свързване на 40 потребителя, които изпращат по 10 заявки. Експериментът е проведен при условие, че потребителите се регистрират (registration) и влязат в системата (login page). Както се вижда по-долу от резултатите за производителността на системата, тя е устойчива.



Фиг. III.9. Графични резултати

Обобщената статистика предоставя информация за двете изходни величини: време за отговор (ResponseTime) и бързодействие на сървъра (ThroughPut). Данните са с близки стойности и това означава, че системата за автоматично управление е стабилна и надеждна. От друга страна те показват, че при задаване на постоянни стойности за параметри KeepAlive и MaxClients, уеб сървъра Apache се саморегулира много добре [Димит-11].



Фиг. III.10. Обобщена статистика

Във втората част на главата се представят и резултатите от изследване чрез входни въздействия към програмната система, като се използват видео файлове. За целта системата за разпространение на научни статии е разширена чрез добавяне на модул за популяризиране на конференции, обсъждания и семинари чрез видео файлове. Те се характеризират с голям обем и това води до по-голямо натоварване на ресурсите на хардуерната конфигурация. Последното налага изследване подобряването на работата на програмната система чрез включване на допълнителен модул `mod_deflate` в нея, понеже той компресира съдържанието на файловете и намалява техния размер преди да се изпратят от сървъра към потребителя. Използва се и генератор за статични страници `Apachebench`, който позволява генериране и изпращане на заявки с различна последователност и големина. В експериментите се изпращат общо 250 заявки от `Apachebench`, които са разделени на групи с различна големина. Тези действия, погледнати от перспективата на ТАУ, представляват единични входни въздействия с различна амплитуда. Целта на експеримента е да се сравнят конфигурационните настройки по подразбиране на уеб сървъра Apache с предложените в този дисертационен труд подобрени настройки. Изследва се поведението на хардуерните компоненти - процесора и паметта при натоварване на сървъра чрез различни по големина и начин на изпращане заявки към видео файл. Отчита се включването на параметъра на `mod_deflate` на Apache и неговото влияние върху използването на хардуерните компоненти.

Получените резултати от изследванията с `Apachebench` дават информация за вида на изпратените заявки до уеб сървъра и статистиката за тяхното изпълнение. Става ясно, че прилагането на подобрени настройки увеличава производителността на сървъра.

<pre> root@hs12 ~# ab -n 250 -c 250 http://hs12.iccs.bas.bg:80/venko.mp4 This is ApacheBench, Version 2.3 <\$Revision: 655654 > Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/ Licensed to The Apache Software Foundation, http://www.apache.org/ Benchmarking hs12.iccs.bas.bg (be patient) Completed 100 requests Completed 200 requests Finished 250 requests Server Software: Apache/2.2.15 Server Hostname: hs12.iccs.bas.bg Server Port: 80 Document Path: /venko.mp4 Document Length: 77695623 bytes Concurrency Level: 250 Time taken for tests: 36.547 seconds Complete requests: 250 Failed requests: 0 Write errors: 0 Total transferred: 19423971750bytes HTML transferred: 19423905750 bytes Requests per second: 6.84 [#sec] (mean) Time per request: 36546.521 [ms] (mean) Time per request: 146.186 [ms] (mean, across all concurrent requests) Transfer rate: 519029.50[Kbytes/sec]received Connection Times (ms) min mean[+/-sd] median max Connect: 12 15 1.5 16 17 Processing: 896 28446 11862.2 34455 36521 Waiting: 8 6877 3099.0 6761 12673 Total: 914 29481 11861.4 34471 36534 Percentage of the requests served within a certain time (ms) 50% 34471 66% 35501 75% 36147 80% 36150 90% 36389 95% 36533 98% 36534 99% 36534 100% 36534 (longest request) </pre>	<pre> root@hs12 ~# ab -n 250 -c 250 http://hs12.iccs.bas.bg:80/venko.mp4 This is ApacheBench, Version 2.3 <\$Revision: 655654 > Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/ Licensed to The Apache Software Foundation, http://www.apache.org/ Benchmarking hs12.iccs.bas.bg (be patient) Completed 100 requests Completed 200 requests Finished 250 requests Server Software: Apache/2.2.15 Server Hostname: hs12.iccs.bas.bg Server Port: 80 Document Path: /venko.mp4 Document Length: 77695623 bytes Concurrency Level: 250 Time taken for tests: 38.199 seconds Complete requests: 250 Failed requests: 0 Write errors: 0 Total transferred: 19423971750bytes HTML transferred: 19423905750bytes Requests per second: 6.54[#sec] (mean) Time per request: 38198.849[ms] (mean) Time per request: 152.795 [ms] (mean, across all concurrent requests) Transfer rate: 486576.38 [Kbytes/sec]received Connection Times (ms) min mean[+/-sd] median max Connect: 13 16 1.6 16 18 Processing: 974 30516 11947.9 36139 38172 Waiting: 6 6374 3019.0 6304 12674 Total: 991 30532 11948.9 36156 38186 Percentage of the requests served within a certain time (ms) 50% 36156 66% 37262 75% 37794 80% 37798 90% 38029 95% 38185 98% 38185 99% 38185 100% 38186 (longest request) </pre>
---	---

Фиг. III.11. Резултати от експеримента – сравнение на настройките по подразбиране с предложените подобрени настройки

По време на изпращане на заявките чрез Apachebench, с програмата SAR е проследено поведението на хардуерните компоненти процесор и памет, отнасящо се до промените в конфигурационния файл на Apache. SAR е команда на операционната система Linux, която брои и показва дейностите, изпълнявани от последната. По този начин се изследва влиянието на подобрените конфигурационни настройки към основните хардуерни компоненти. От направените експерименти може да се види, че натоварването на процесора и паметта е намаляло, съответно на процесора с 2,37%, а на паметта – с 8,93%.



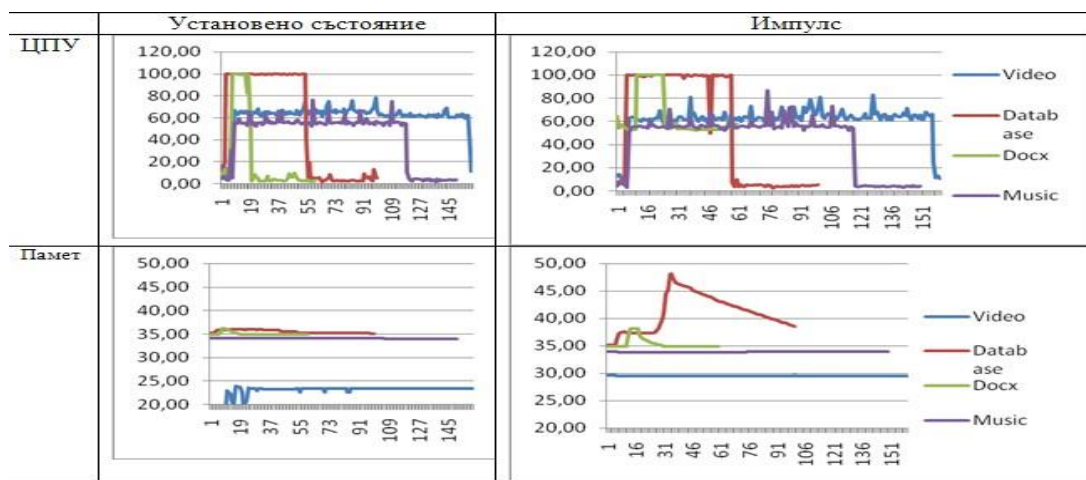
Фиг. III.12. Натоварване на процесора и паметта при различни настройки на конфигурационния файл

При изследване влиянието на модула `mod_deflate` върху времето за обслужване на заявки от сървъра са използвани три варианта за настройка на уеб сървъра Apache. Първият е настройки по подразбиране на Apache, вторият е оптимизирани настройки на `prefork` и изключен модул `mod_deflate`, а третият е подобрени настройки на `prefork` и включен модул `mod_deflate`. Става ясно, че включването на модула `mod_deflate` не влияе на времето за обслужване на заявките. Неговото самостоятелно включване не е ефективно решение за подобряване на производителността на системата за обработване на заявки към видео файлове [DimiT-13].

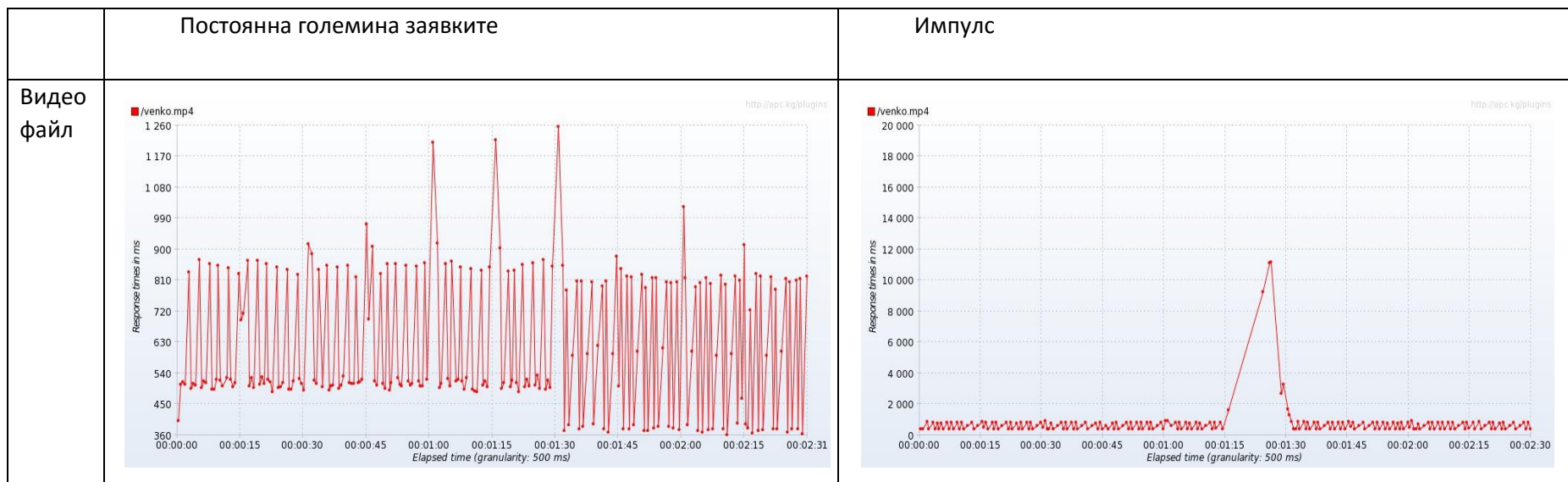
Във втората част на главата са представени и резултати от реализиране на входни въздействия към програмната система чрез различни видове файлове. Обектът на

управление – Apache сървър е изследван при нови входни въздействия, които са музикални, видео, текстови файлове и база данни. Използва се програмата Jmeter за изпращане на входни въздействия във вид на http заявки и се проследяват промените в натоварването на най-важните хардуерни компоненти – процесора и паметта. Заявките са изпращани с постоянна големина и чрез импулс в определен момент от време, като се наблюдава поведението на показателите за производителност – време за отговор и брой на заявките в приемащата опашка на уеб сървъра. Резултатите показват (фиг.19 а-г), че когато заявките се изпращат с еднаква големина на импулса към програмната система, времето за отговор е постоянно, а при изпращане на единичен импулс към сървъра, времето за отговор се увеличава изключително много. Това е предпоставка да се търси решение за намаляване на времето за отговор от системата. Ако използваме терминологията на теория на автоматично управление, изпращането по този начин на заявки към системата може да се представи като единично входно въздействие и като единична импулсна функция.

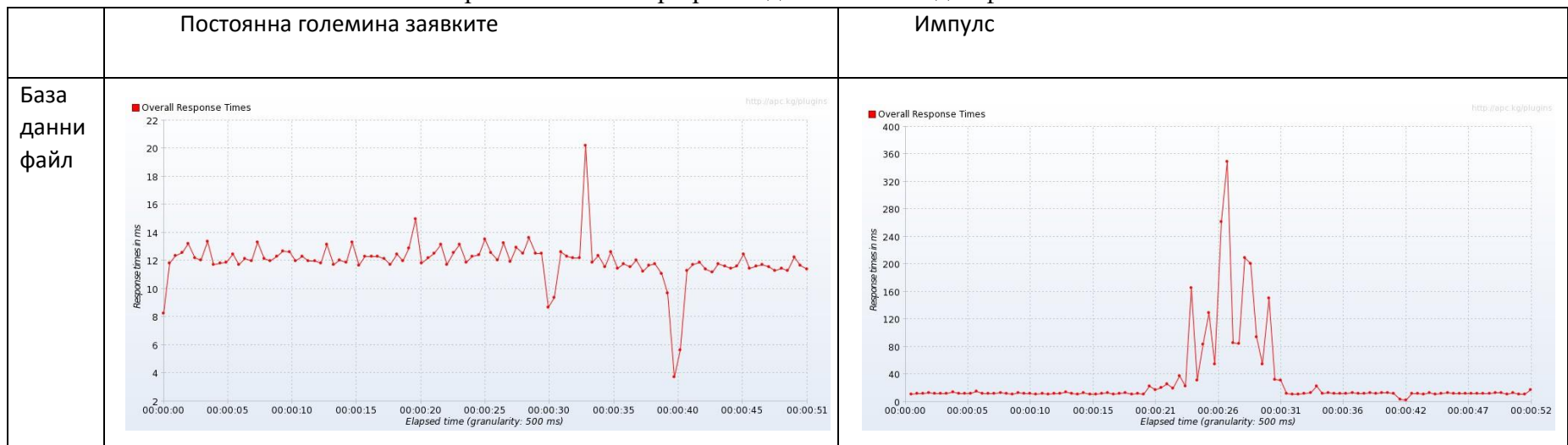
При измерване натоварването на хардуерните компоненти бе констатирано, че при изпращане на импулс се наблюдават по-големи и по-чести флукутации в използването на процесора и паметта при всички видове файлове. Тези смущения показват, че е необходима промяна в конфигурационните параметри на уеб сървъра Apache и на TCP комуникационния протокол. Така ще се облекчи работата на системата и ще се намалят смущенията в нея [Димит-13].



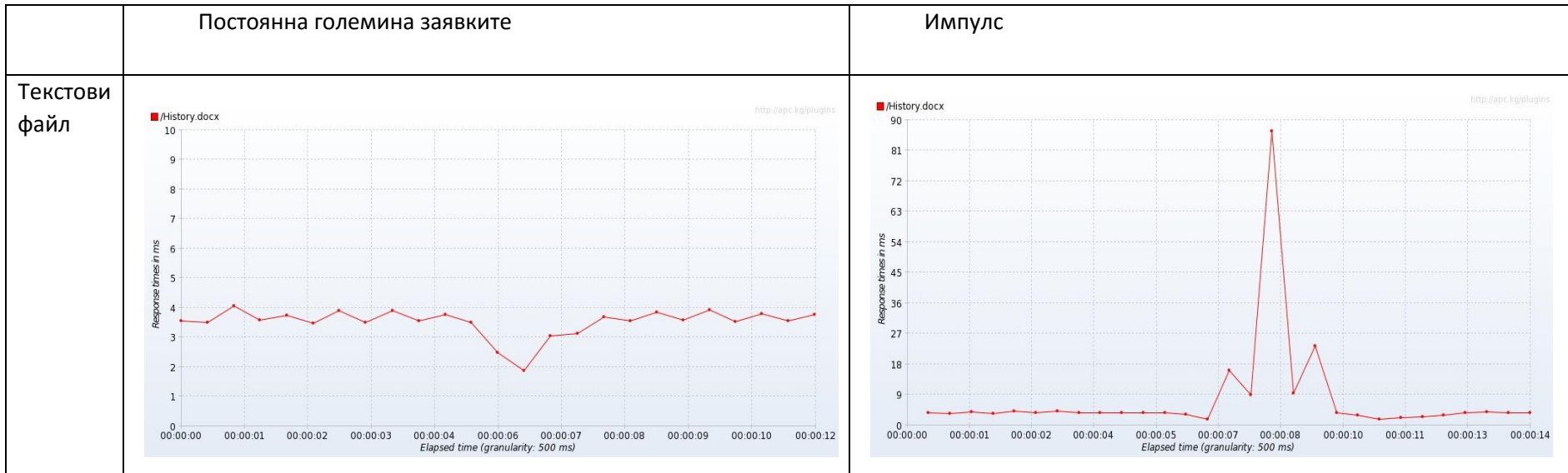
Фиг. III.15. Реакция на изходните сигнали



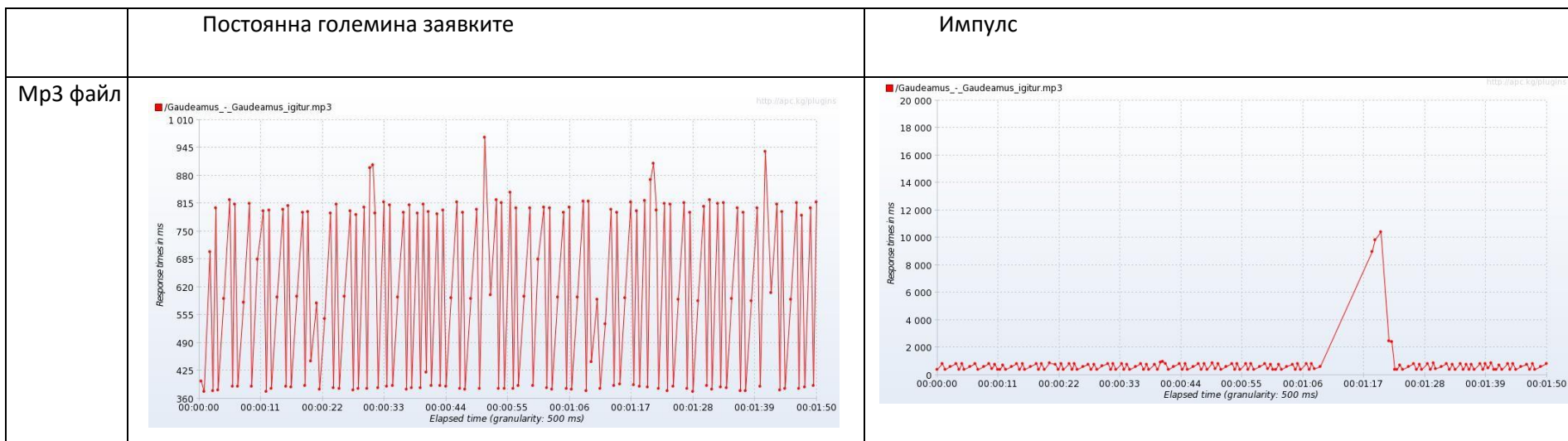
Фиг. III.14-а. Измененията на времето за отговор при входен сигнал- видео файл



Фиг. III.14-б. Измененията на времето за отговор при входен сигнал – файл база данни



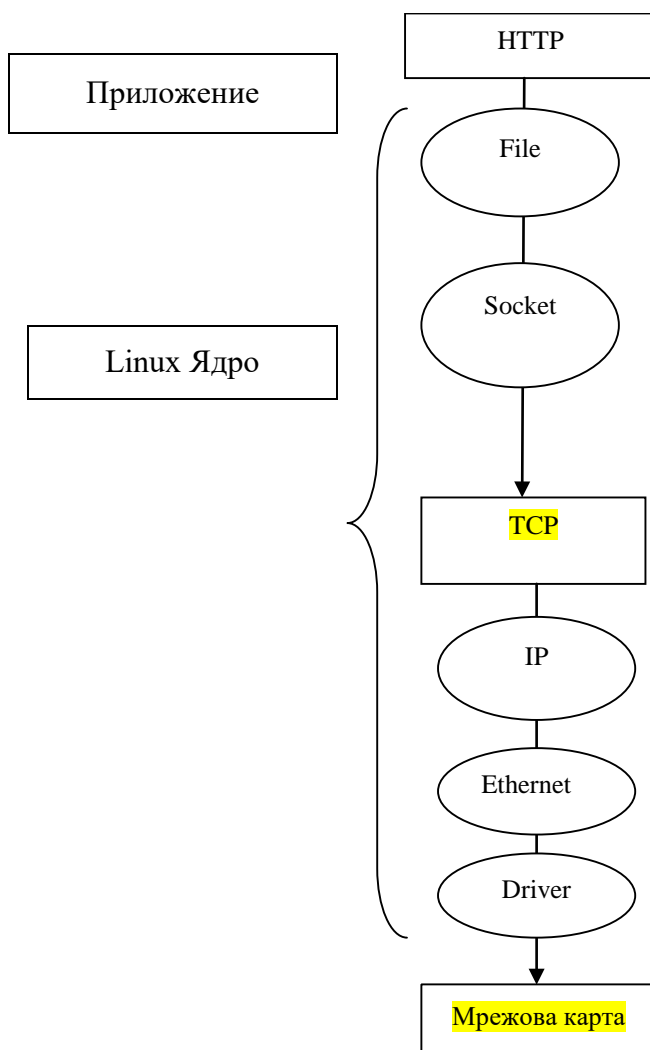
Фиг. III.14-в. Измененията на времето за отговор при входен сигнал – текстови файл



Фиг. III.14-г. Измененията на времето за отговор при входен сигнал – mp3 файл

В третата част на главата се представят резултати от експерименти за подобряване производителността на изследваната програмна система чрез промяна на нейни параметри с цел подпомогане на едновременния достъп на повече читатели до информационната система. Въвеждането на допълнителни настройки на различни нива в компютърната система увеличава производителността на програмната система [Gregg-13]. Това се потвърждава и от факта, че при разработване на информационни системи, усилията на екипите са насочени към бързо внедряване на софтуера и се отделя малко време за измерване на производителността и оптимизирането им. Лошо са организирани бази данните и не са реализирани подходящите настройки на устройствата за съхранение на данните. Тези недостатъци са предпоставка за търсене на начини за подобряване на използваните информационна система, уеб сървър и операционна система.

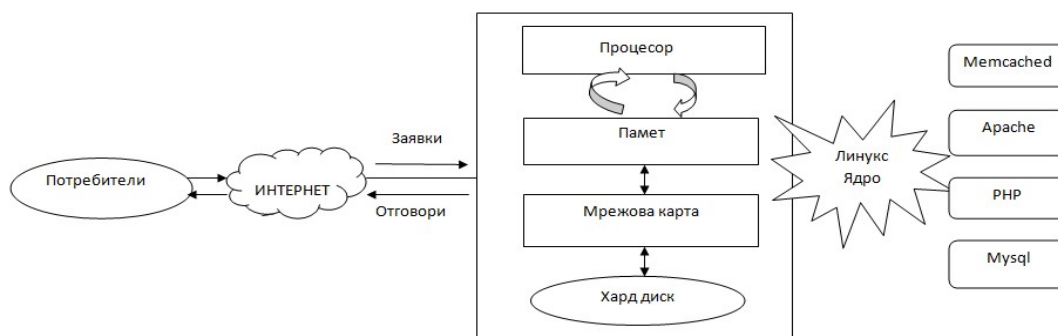
Компютърната система може да се представи като последователност от подсистеми, работещи в синхрон. Тя включва подсистемата на TCP протокола, подсистемата на http протокола и подсистемата на входно/изходните данни. Параметрите на тези подсистеми имат нужда от допълнителна настройка за специфичната хардуерна конфигурация, на която работят [MDVHR-01].



В дисертационния труд тази допълнителна настройка е извършена по отношение на параметрите в Linux ядрото и по-специално в мрежовите настройки и TCP комуникационния протокол. Комуникацията между крайните потребители преминава през три основни зони – първата зона е самата програмна система – уеб сървъра Apache; втората зона е компютърната система, която включва операционната система и TCP настройките; третата зона е пътят по мрежата.

В дисертацията се извършва управление в програмната система, т.е в уеб сървъра, както и във втората зона (компютърната система), където се анализират и подобряват настройките на TCP протокола и се инсталират програми за подобряване производителността на операционната система.

Фиг. III.16 Последователни подсистеми



Фиг. III.17. Комуникацията между потребителите и сървъра

Производителността на компютърните мрежи на ниво TCP протокол се ограничава от два прозореца: прозорец на задръстванията (congestion window) и прозорец на получаване (receive window). За постигане на по-качествено управление се изисква да се направят допълнителни настройки на параметрите на TCP протокола. Те се разделят в няколко категории според тяхното предназначение [WigeN-13],[OracC-00],[SoftP-15] о За подобряване на производителността – чрез регулиране размера на опашките, изключване на селективното потвърждение, повишаване на прозореца на задръстванията и др. Операционната система разполага с настройки за регулиране размера на опашката между мрежовата подсистема на ядрото и драйвера на мрежовата карта. Размерът на опашката се настройва така, че да няма загуби на данни, поради препълване на локалния буфер. При използване на неподходящи настройки на TCP протокола се попада в състояние на управление на задръстването, което ограничава големината на изпратените пакети и причинява загуби на UDP пакетите.

о За обслужване на повече клиенти – за сървъри, които обслужват голям брой сесии едновременно се изисква включване на няколко допълнителни опции. Чрез тях се определя диапазонът на използваните портове от UDP и TCP протоколите, оптимизира се производителността при голям брой клиенти, управляват се свързвания по време на използване на TCP протокола и се променят настройките на времето за затваряне на свързването (close connection) и времето за убиване на остарял процес [Moura-00]. Използва се и опция за увеличаване на максималния брой свързвания, които не са получили потвърждение [Garre-09].

о За подпомагане работата на web server – при инсталиран веб сървър, който обслужва много кратки TCP свързвания, е подходящо да се настроят следните опции: включване на параметъра TCP_tw_recycle, който изчиства времето за изчакване (time_wait) в сокетите; включване на параметъра TCP_reuse, който позволява повторно използване на сокета и намалява натоварването на сървъра; настройване на параметъра

TCP_synack_retries, който определя броя синхронизиращи пакети, изпратени преди ядрото да откаже свързването. Друга опция, подпомагаща работата на уеб сървъра е /Net/core/somemconn. Тя се използва за определяне на броя приети заявки и позволява да се обработят голям брой заявки едновременно [HTCAG-10].

Обикновено уеб сървърът Apache позволява лесно изграждане на среда за поддържане на уеб сайтове, с минимална промяна в конфигурациите и малко усилия от страна на администратора. Но при използването на настройките му по подразбиране се наблюдава голямо натоварване на хардуерния ресурс на компютърната конфигурация и особено на паметта. Това поставя нелеката задача за оптимизиране производителността на системата, тъй като липсват лесни и бързи правила за нейното решение. За да се постигне максимална производителност се изискват много експерименти с различни опции и условия [Avoya-11], които включват вид на заявките, изпълнявани от сървъра, оптимизиране на база данните, кеширане на php скрипта. В хода на дисертационния труд са разгледани и подобрили всички тези условия.

Управлението на програмната система е реализирано чрез оптимизиране на модулите, използвани от Apache, и чрез синтез на директивите на Apache. Извършен е анализ на използваната памет на компютърната система от програмната система и е изчислена стойността на параметъра на конфигурационния файл на Apache MaxClients по два начина: чрез програмни кодове, написани на езика Perl; чрез команди, написани на езика awk и се стартира Linux програмата Free. При втория начин се използва по-малко хардуер ресурс за установяване на показателите на системата, понеже не е необходимо да се стартират и компилират допълнителни програми. Посредством тези два начина на изчисление се получава информация за паметта, използвана от един процес на Apache и от всички други приложения, работещи с операционната система.

На базата на изчисленията за използване на паметта на хардуерната конфигурация се предлагат два варианта за промяна на настройките на Prefork модула на Apache. При първия се намаляват стойностите на параметрите на prefork модула. Целта е да се разгледат възможностите за реакция на системата и да се реализира управление, при което при по-малък брой потребители на информационната система, ресурсите на системата да се използват от другите програми, работещи с операционната система. Във втория се увеличават стойностите на параметрите на prefork модула – по този начин се изисква повече хардуерен ресурс за уеб сървъра Apache и се стартират допълнителни дъщерни процеси при по-голям брой потребители на системата. Това се осъществява чрез промяна на параметрите (bestSettings и highSettings), показани в таблица III.2. Това са основни директиви за настройка на уеб сървъра Apache, които се използват за определяне на процесите, изпълнявани от него.

Таблица III.2. Параметри на директивите на Apache

Prefork	bestSettings	highSettings
StartServers	3	10
MinSpareServers	3	10
MaxSpareServers	10	15
MaxRequestWorkers	50	84
MaxConnectionPerChild	1000	1000
Timeout	40	45
MaxKeepAliveRequests	14	14
KeepAliveTimeout	2	2

В резултат от направените промени в настройките на директивите за конфигуриране на уеб сървъра Apache се променя и натоварването на основните компоненти на системата т.е процесора и паметта. Използването на процесора при настройките highSetting е представено в колона “cpu_high”, а при bestSettings – в колона “cpu_best”. Използването на паметта при настройките highSetting е показано в колона “mem_high”, а при bestSettings – в колона “mem_best”. От резултатите става ясно, че когато постъпват по-малко от 20 клиента, настройките highSetting са по-ефективни за обслужването на клиентите, понеже се използва по-голям ресурс на процесора и по-малък ресурс на паметта. За по-голям брой клиенти използването на конфигурационните настройки bestSettings са по-ефективни, тъй като се използва по-малко натоварване на паметта и по-голямо на процесора. Чрез прилагането на два варианта на настройки за конфигурационния файл се постига по-бързо обслужване на клиентите и по-ефективно използване на наличните хардуерни ресурси.

Таблица III.3. Промени (в %) на използваните памет и процесор

Клиенти	cpu_best, %	mem_best, %	cpu_high, %	mem_high, %
2	34,123	95,129	43,4142	88,845
10	39,974	94,84	43,324	89,271
15	37,989	91,729	36,188	93,521
20	39,934	94,284	42,788	93,209
25	40,072	92,164	37,689	84,995
30	40,319	91,923	40,313	85,19
35	40,559	90,401	44,864	84,963
40	46,41	82,648	42,476	89,137

Ефективността на направените промени в конфигурационния файл на Apache се изследва и чрез показателите за производителността на системата – бързодействие и време за отговор. Резултатите представят измененията в тези два показателя при три случая: с

настройки по подразбиране (default) и два варианта за преконфигурирани настройки на конфигурационния файл на Apache (bestSettings и highSettings). Установява се, че за настройки по подразбиране времето за отговор (RT_defaultold) е по-голямо, отколкото за двата случая с преконфигурирани настройки. Когато сървърът работи с настройки по подразбиране, времето за обслужване на потребителските заявки е по-голямо. Предложените два варианта за преконфигурирани настройки на параметрите на директивите на Apache предоставят по-бързо обслужване на потребителските задачи.



Фиг. III.20. Изменение на времето за отговор при различни настройки на конфигурационния файл

Също така бе установено, че бързодействието с настройките по подразбиране е помалко в сравнение с двата случая на преконфигурираните настройки. Бързодействието с настройки highSettings е по-добро, когато се обслужват по-малко от 20 клиента едновременно.



Фиг. III.21. Бързодействие при различни настройки на конфигурационния файл.

Създаването на управление за автоматично преконфигуриране на Apache цели побързо обработване на заявките в зависимост от броя на потребителите. От получените резултати може да се направи извода, че програмата за автоматично преконфигуриране на Apache трябва да използва настройките, наречени `highSettings`, когато информационната система се използва до 20 потребителя и настройките `bestSettings`, когато потребителите са повече от 20. В програмния код за преконфигуриране на Apache се използват Linux програмата `PS` и езика `awk`.

Преконфигурирането на уеб сървъра Apache се реализира чрез измерване на параметъра `%cpu`. В програмата за управление той измерва колко процесорно време се използва от процеса и въз основа на това се преконфигурира конфигурационния файл на Apache. Този параметър се проверява в цикъла `while`, след което се избира кои конфигурационни файлове да се използват с оглед неговата стойност. Представеният програмен код осъществява автоматично преконфигуриране на уеб сървъра Apache на потребителско ниво в зависимост от броя потребители на информационната система. По този начин се прилага по-ефективно обслужване на потребители на разработеното приложение.

Управляващата програма е от типа:

```
#!/bin/bash
#!/bin/sed
while :
do
for i in $( ps aux | grep opt/lampp/bin/httpd
|awk '{s+=$3} END {print s}' ); do
if [ $i \> 3.9 ]
then
cp httpd-mpm_2.conf httpd-mpm.conf
cp httpd-default_2.conf httpd-default.conf
sudo /opt/lampp/lampp reloadapache
echo "Load new settings!" elif
[ $i \< 1.00 ]
then cp httpd-mpm_1.conf httpd-
mpm.conf cp httpd-default_1.conf httpd-
default.conf sudo /opt/lampp/lampp
reloadapache echo "Load settings by
default!" fi
done
sleep 20
done
```

В дисертационния труд са представени и основни параметри за подобрене управлението на файловата система – `swappiness` и `file-max`. Разгледани са и допълнителни механизми, като например обновяване на хардуерната конфигурация, подобряване настройките на браузъра и кеширане на данни от страна на браузъра.

Една от основните задачи на дисертационния труд е да сравнени ефективността на реализираните преконфигурационни настройки. За да се представят по-добре и по-ясно резултатите от подобренията за системата се изследва работа на компютърната система при обслужване на заявки към база данни с нов сценарий на изпращане. Входният сигнал е във вид на импулс, генериран от програмата за симулиране на реални клиенти.

От изследването става ясно, че осреднените стойности за използване на процесора, с настройки по подразбиране е 3,53 %, а с преконфигурирани настройки на системата то е 5,68 %. По този начин се определя началната точка на работа на процесора, при която започва изпращането на заявки. В този момент чрез генератора на входен сигнал се симулира едновременно изпращане на заявки от три клиента към системата. В този период на изпращане на заявки средното използване на процесора е 92,44% при настройки по подразбиране и 88,54% при преконфигурирани настройки. Изпращането на заявки с такава големина трае 20 секунди, след което се стартира симулиране на изпращане на заявки от пет клиента едновременно, като в този период средното използване на процесора при настройки по подразбиране е 99,28%, а с подобрени настройки е 96,39%. Изследван е и трети интервал, в който отново се изпращат заявки от три клиента едновременно, като при него използването на процесора с настройките по подразбиране е 99,50% и 91,24% с подобрените настройки. Използването на процесора в трите интервала намалява съответно с 3,90%, 2,89% и 8,26%.



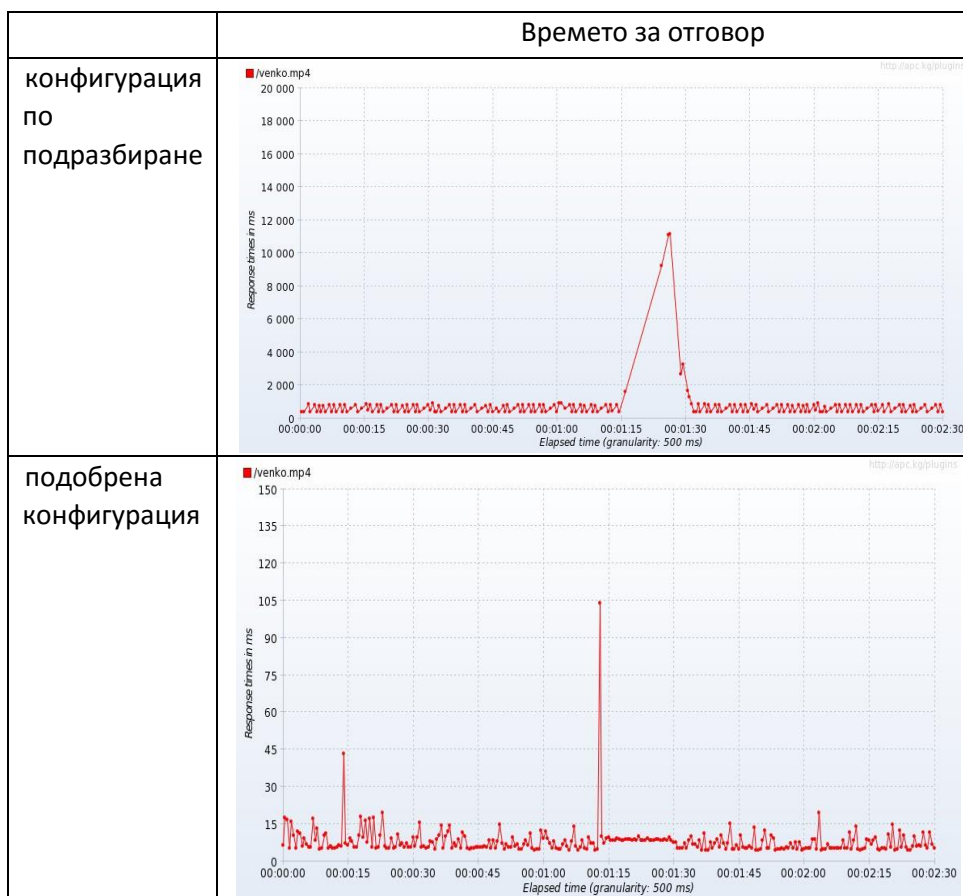
Фиг. III.23. Сравнение на използването на процесора при работа с база данни

Направените настройки на конфигурационния файл са приложени и за сравнение с резултатите за видео файл. Средното използване на процесора при начални условия, когато изследването се извършва с настройки по подразбиране е 12,35%, а при преконфигурирани настройки е 44,82%. Изпращането на заявки се осъществява по следния сценарий: в начало се изпращат заявки от един клиент, след което се увеличава генерирането на заявки от 10 клиента едновременно, и последващо изпращане на заявки отново от един клиент. При такъв сценарий на изпращане на заявките може да се направи заключението, че средното използване на процесора в първия интервал на изпращане на заявки е намаляло с 38,20%, във втория – с 29,91%, а при следващото изпращане на заявка от един клиент е намаляло с 41,27%.



Фиг. III.24. Сравнение на използването на процесора при работа с видео файл

Подобренията от приложените настройки на конфигурационния файл на Apache и TCP протокола се потвърждават и от графиките за времето за отговор. То намалява около 10 пъти при нововъведената конфигурация, което представлява значителен принос на изследването.



Фиг. III.25. Време за отговор при изследване на видео файл

От двете представени сравнения на резултатите – за база данни и за видео файл става ясно, че преконфигурирането на TCP протокола и на конфигурационния файл на Apache водят до подобрене в използването на процесора и до намаляване на времето за отговор на системата. По този начин се доказва, че предложените в дисертацията промени в настройките на уеб сървъра Apache и комуникационния протокол TCP са ефективно решение за създадената информационна система, която поддържа база данни с научни статии и видео файлове от семинари, конференции и обсъждания.

В заключение, глава трета на дисертационния труд демонстрира приложимостта на Теория на автоматичното управление за подобряване ефективността на програмни системи, изразена в оптимизиране на производителността. Експерименталната работа показва, че предложените в хода на дисертационния труд промени в TCP комуникационния протокол и в конфигурационния файл на Apache, водят до подобрене в показателите за качество на производителността – натоварване на процесора, натоварване на паметта, време за отговор.

Глава 4. Използване на методите за идентификация от ТАУ за подобряване на производителността на компютърна система

В последната глава на дисертацията е представено прилагането на методите за идентификация за конкретна компютърна система. По този начин се цели да се определят динамичните характеристики на системата чрез показатели на производителността ѝ.

Представените характеристики се апроксимират и се изграждат модели на системата. Тяхната ефективност се изследва чрез метода на контролните карти, за да се определи приложимостта им. Реализираните софтуерни подобрения в изследваната система са моделирани чрез нелинейни модели, поради тяхната гъвкавост относно сложни явления и процеси. Получените модели са симулирани в симулационна среда на Matlab, за да се изследва поведението на системата при включване на пропорционално-интегрален (ПИ) регулатор за управление на натоварването на процесора. С използването на пропорционален (П) регулатор в симулационната схема се демонстрират ефективността на приложеното софтуерно управление чрез формализъм от ТАУ.

В първата част на главата са разгледани методите за идентификация, които позволяват да се изградят модели на динамични системи чрез експериментални изследвания. Идентификация на системата е научна област, изследваща изграждането на математически модели на динамични системи чрез наблюденията на входно-изходни данни. Един от начините за идентифициране на обекта за управление е чрез подаване на единично стъпаловидно входно въздействие към него при нулеви начални условия, а реакцията на обекта е неговата преходна характеристика (функция). Получаването на преходната характеристика се осъществява чрез провеждане на експеримент. Целта е събиране на недостъпна и неясна информация за обекта и въз основа на статистически методи да се

извърши моделирането и оптимизирането му. При снемане на преходни характеристики експериментално, т.е. реализиране на наблюдение, данните много често са зашумени от влиянието на неконтролируеми въздействия, което изисква осъществяването на експериментално снемане на данните няколко пъти (4-6 пъти).

При моделиране на система за автоматично управление чрез експериментални изследвания за получаване на техните преходни характеристики се използват таблици за записване на получените данни. Целта на тяхното прилагане е нагледност на резултатите и улесняване на последващата обработка на данните.

В таблица IV.1 са показани първите 15 измерени данни за натоварването на процесора (Cpu%₁₋₄), изчислените преходни характеристики на натоварването на процесора (Y_{1-4}) и осреднената преходната характеристика при подаване на единично въздействие от 10 потребителя. Изискването, което трябва да се съобрази, е обектът да е в установено състояние. При обекти, при които началният момент е различен от нула, като нашия, се използва следната зависимост [Гарип-07]

$$y_{ij}^0 = y_{ij} - y_{j0}, \quad i=0, 1..(N-1) \quad j=1, 2..n.$$

(3)

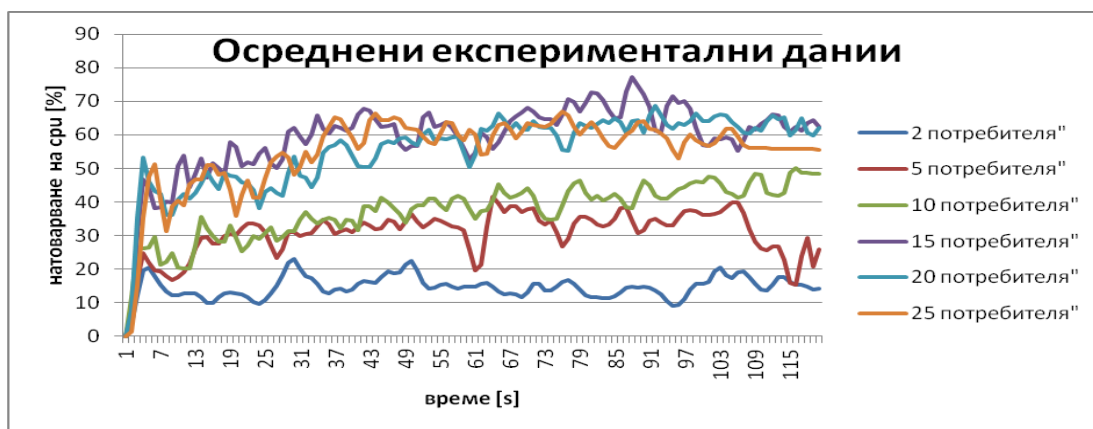
Така преходната функция се премества в началото на координатната система и започва от 0, резултатите от прилагането на тези действия са показани в табл. IV.1 в колони Y_{1-4} . В последната колона се изчислява средната стойност от четирите експериментално проведени опита. Този начин на обработване на експерименталните данни се използва за получаване на осреднени преходни характеристики на натоварването на процесора и за бързодействието (получените данни от системата) при обслужването на различен брой клиенти.

Табл.IV.1 Получаване на осреднена експериментална преходна характеристика

T[s]	Измерени данни				При нулеви начални условия				\bar{Y}
	Cpu% ₁	Cpu% ₂	Cpu% ₃	Cpu% ₄	Y ₁	Y ₂	Y ₃	Y ₄	
1	19,6420	56,6896	62,4040	67,9263	0,0000	0,0000	0,0000	0,0000	0,0000
2	21,8174	59,9757	61,0201	67,6415	2,1754	5,0404	1,6464	1,2349	2,5243
3	27,2572	56,1521	61,0108	67,7781	7,6152	14,6565	5,9758	6,8977	8,7863
4	33,4232	48,7413	62,8116	67,7385	13,7812	24,8439	8,1972	12,5882	14,8526
5	38,8170	43,8372	63,5706	66,5161	19,1750	32,9478	9,7298	15,5006	19,3383
6	43,8703	42,8724	61,2725	62,6580	24,2283	36,2339	13,6123	19,3758	23,3625
7	49,0946	43,6702	57,4372	57,4163	29,4526	32,4102	21,0318	27,9603	27,7137
8	54,6025	44,2807	55,6166	55,6947	34,9605	24,9995	29,9253	37,4323	31,8294
9	59,9135	44,6167	58,2741	60,0223	40,2715	20,0954	35,9234	42,2437	34,6335
10	62,7727	45,7113	64,4806	66,5499	43,1307	19,1306	36,9471	42,7508	35,4898
11	61,8238	47,0324	70,0408	69,6327	42,1818	19,9284	35,5633	42,4660	35,0348
12	59,2710	47,4592	71,4321	68,3844	39,6290	20,5389	35,5539	42,6026	34,5811

13	56,8640	48,4573	69,3436	66,4653	37,2220	20,8748	37,3547	42,5630	34,5036
14	53,4603	52,3111	67,1831	66,7838	33,8184	21,9694	38,1138	41,3406	33,8105
15	48,6432	58,0175	66,7838	68,1800	29,0012	23,2906	35,8156	37,4825	31,3975

От представени осреднени преходни характеристики на фиг.IV.2 (по оста x са представени данните от измерванията във времето с момент на дискретизация 1 секунда, а по оста y са представени данните от натоварването на процесора в [%]) се вижда, че те се разделят на две групи, когато изследваната системата обслужва по-малко или повече от 15 потребителя. За да се изведе обобщена преходна характеристика е необходимо получените данни да имат близки стойности. Въз основа на това изискване се изгражда модел, който ще определи зависимостта между стойностите по оста x и тези по оста y. Полученият модел за натоварването на процесора на изследваната система се използва за прилагане на допълнителни настройки, чрез които да се постигне по-добро обслужване на потребителите на системата. Натоварването на система в голяма степен зависи от броя потребители на системата, затова разделянето на две нива е подходящо решение – когато потребителите са по-малко или повече от 15. Изборът на нивата се основава на близостта на представените криви. Изчисляват се модели за повече и по-малко от 15 потребителя. От друга страна, честата промяната на настройките на системата изисква допълнителни изчисления и хадруерен ресурс, а чрез изграждане на модел се ограничават изчисленията и се намалява използвания ресурс за настройка.



Фиг. IV.2 Осреднени експериментални данни за натоварването на процесора

Тенденцията в данните най-добре се описва с алгебричен полином от трета степен. Редът на полинома, който апроксимира данните се определя от броя на флукуациите в данните или от минимумите и максимумите на кривата. Той описва модела с висок коефициент детерминация и е по-лесно реализируем от полиноми от по-висока степен. Осъществената апроксимация на експерименталните данни с полином от трета степен за по-малък брой потребители е:

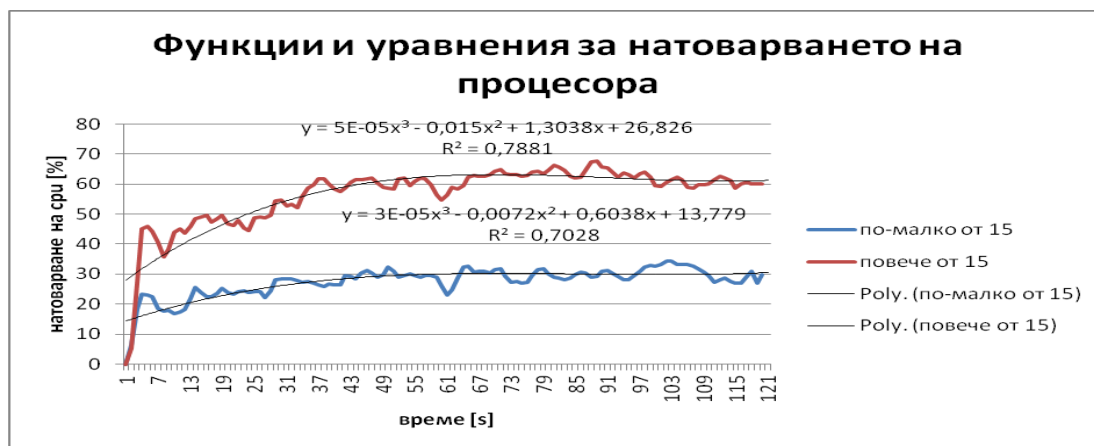
$$y=0,0003x^3-0,0072x^2+0,6038x+13,779 \tag{8}$$

Коефициентът на детерминация на получения модел е $R^2=0,7028$. Това означава, че 70,28% от данните за натоварването на процесора се изменят във времето според установения полином от трети ред при по-малко от 15 потребителя на разработената информационна система.

Апроксимация на експерименталните данни с полином от трета степен за повече от 15 потребителя се описва с:

$$y=0,00005x^3-0,015x^2+0,6038x+26,826 \quad (9)$$

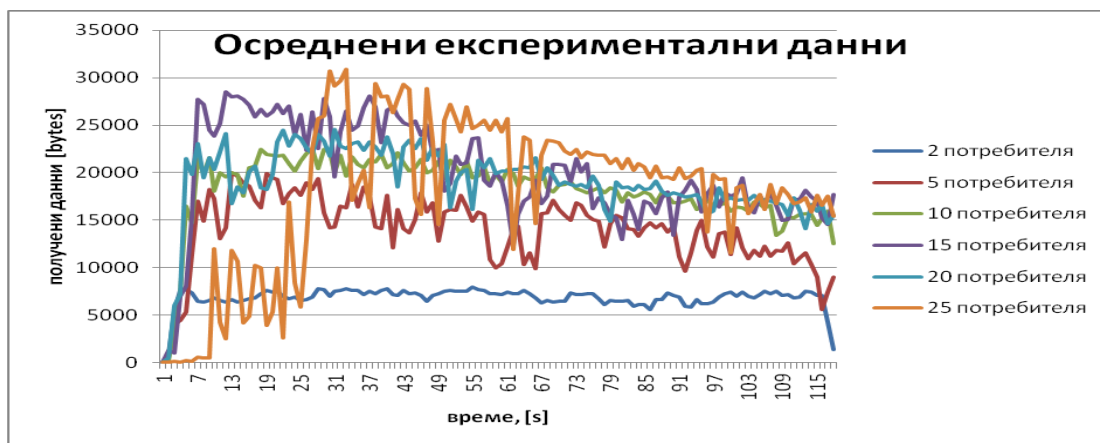
Коефициентът на детерминация е $R^2=0,7881$. Това означава, че 78,81% от данните за натоварването на процесора се изменят във времето според установения полином от трети ред при повече от 15 потребителя на разработеното приложение.



Фиг. IV.3. Функции и уравнения за натоварването на процесора

Системата се изследва допълнително чрез анализиране на данните от втори параметър, за да се установи зависимостта на количеството получени и изпратени данни от системата и броя потребители на системата. Целта е да се изгради модел на неговото поведение в зависимост от броя потребители и да се получи по-ясна представа за бързодействието на системата.

Осреднените преходни характеристики за този параметър се наблюдават с по-голяма колебливост и това явление най-ясно проличава при експериментите с 25 потребителя. Представените експериментални преходни характеристики за бързодействието на системата имат по-близки стойности, когато потребителите са повече от 15. При по-малък брой потребители стойностите им се отдалечават. При определянето на обобщена преходна характеристика е необходимо получените експериментални данни да имат близки стойности. Въз основа на резултатите за получените осреднени експериментални данни от програмата Jmeter и посоченото условие за близки стойности, отново се изгражда модел на две нива.



Фиг. IV.4. Осреднени експериментални данни за бързодействието на системата

Моделите на бързодействието на системата се описват с алгебричен полином от четвърти ред поради по-големия брой на флукуациите в данните. Полиномът, който описва модела за получените данни за повече от 15 потребителя е:

$$y = -0,0012x^4 + 0,362x^3 - 38,702x^2 + 1599,9x + 1539,6 \tag{10}$$

Коефициентът на детерминация е $R^2 = 0,8225$, което означава, че 82,25% от данните за бързодействието на системата се изменят във времето според установения полином от четвърти ред при повече от 15 потребителя на разработеното приложение. Полиномът, който описва модела за получените за по-малко от 15 потребителя е: $y = -0,0012x^4 + 0,3317x^3 - 30,362x^2 + 1048,9x + 4112,5$ (11)

Коефициентът на детерминация е $R^2 = 0,733$, което означава, че 73,3% от данните за бързодействието на системата се изменят във времето според установения полином от четвърти ред за по-малко от 15 потребителя на разработеното приложение.



Фиг.IV.5. Функции и уравнения на бързодействието на системата

При измерване показателите на натоварване на процесора и бързодействие при подаване на единично входно въздействие във вид на брой потребители на уеб сайт, бе установено, че реакцията на изследвания обект се променя при увеличаване на броя на

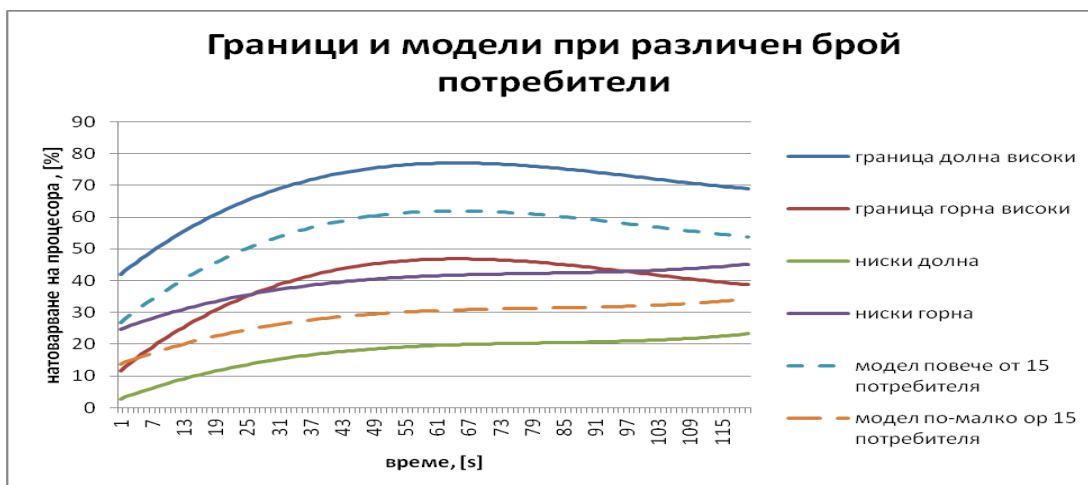
потребители на софтуерната система. За описанието на подобни обекти, които си изменят преходните характеристики при промяна на входните въздействия, се използват системи с твърда адаптация.

Поради тази причина във втората част на четвърта глава от дисертационния труд са разгледани системите с твърда адаптация. При тях изменението на параметрите на регулатора се осъществява въз основа на модела на обекта. Чрез измерване на работните натоварвания на променливите на обекта на процеса или въздействащите смущения се установява промяната на модела. По този начин може много бързо системата да реагира на променените натоварване на процесора и бързодействие на системата. Системите с твърда адаптация могат много добре да се приложат и за компютърни системи, понеже при изменение на броя потребители се променят и характеристиките на обекта – в случая уеб сървър Apache. Идентификацията се осъществява чрез измерване натоварването на процесора, а настройките на регулатора се реализират чрез промяна на конфигурационния файл на Apache. Използваната операционна система Linux и уеб сървъра Apache имат много добре направени настройки за саморегулиране по подразбиране. По този начин може много бързо системата да реагира при променено натоварване на процесора и бързодействие на системата. За прилагане на системи с твърда адаптация е необходимо да се изчислят показателите за качеството на измерваните параметри с цел определяне на състоянията, при които тя трябва да бъде прилагана.

За изчисляване показателите за качество в третата част на главата се използва методът на контролните карти. Чрез него се наблюдават текущите стойности на качествените показатели на даден продукт. Той позволя бързо вземане на решение с цел коригиране и избягване на отклонения от желаните стойности. При метода на контролните карти се използва правилото на трите сигми, т.е стойности на показателя на качеството x имат нормален закон за разпределение и попадат в интервала $m_x \pm 3\sigma$ с вероятност 0,9973, където m_x е математическото очакване на показателя, а σ стандартното му отклонение.

Прието, е че показателите на изследваната система са натоварването на процесора и постъпилите данни от потребителите, т.е бързодействието на обслужване. Техните данни се изчисляват в интервал $\pm\sigma$, поради близостта на получените данни и възможността те да бъдат обработвани с по-малка вероятност. Изчислени са стандартни отклонения за двата модела при повече и при по-малко от 15 потребителя. Тези стойности се използват за определяне на граници, в които да се приложат изчислените модели.

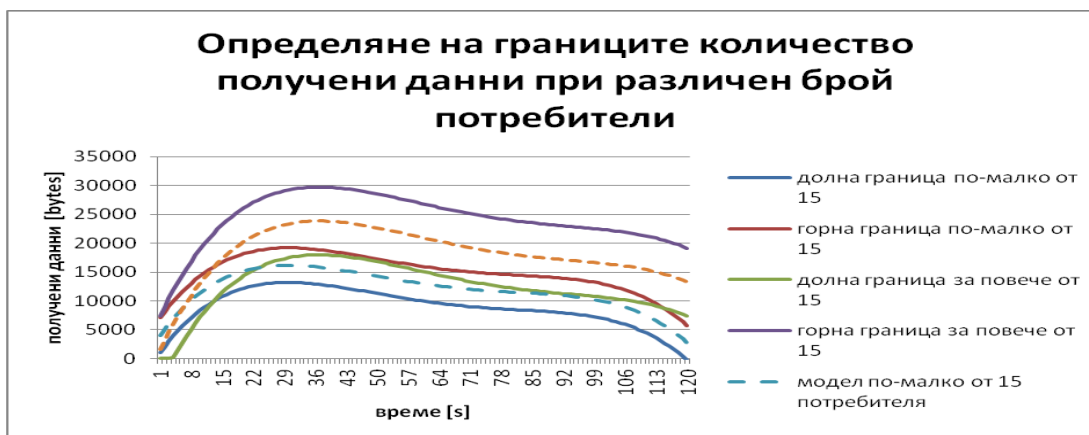
На фиг. IV.7. са показани графично функциите, изчислени за моделите за натоварването на процесора и за бързодействието на система, съвместно с изчислените стандартни отклонения на експерименталните данни. Това позволява да се определят границите на действията на подобренията на конфигурационните настройки на системата.



Фиг.IV.7. Функции на моделите и границите на показателя на натоварване на процесора.

Вижда се, че долната граница на модела за повече от 15 потребителя в два момента пресича горната граница на модела за по-малко от 15 потребителя. Освен това се наблюдава един малък интервал от пространството, който не се покрива от нито един от двата модела. За да се отстранят тези недостатъци на моделите може да се приложат допълнителни изчисления като например осредняване стойностите на двете граници.

Относно показателя бързодействие на системата бе установено, че определената долна граница за повече от 15 потребителя е по-малка от горната граница за по-малък брой потребители. Затова е необходимо или да се намали областта, в която да се проследява количеството постъпили данни от потребителите на системата, или да се използва само горната граница за потребители по-малко от 15. Трето възможно решение е да се изчисли средна стойност за двете граници.



Фиг.IV.8 Функции на моделите и границите на бързодействие на системата.

В последната част на главата е представен нелинеен анализ на приложеното управление на програмната система. За да се представи реализираното софтуерно управление чрез прилагане принципите за автоматично управление се използва инструмента на Matlab – System identification tools, който позволява чрез Matlab функции и симулационни блокове да се изградят математически модели. Той е инструмент, който има

широко приложение за идентифицирането на времеви и честотни входно-изходни данни, непрекъснати и дискретни предавателни функции и др. Освен това позволява да се представи динамиката на нелинейни системи чрез изчисляване на Хамерщайн-Винер модели или нелинейни ARX модели с вълновидни мрежи, дървовидни части и сигмоидни мрежови нелинейности. Те предоставят по-голяма гъвкавост за установяване на сложни явления.

Поради факта, че изключително голям брой процеси в компютърните системи са нелинейни, използването на нелинейна ARX структура за тяхното моделиране е изключително лесно, понеже се определят единствено редовете на полиномите А, В и времезакъснението pk за системата. В ARX нелинейната структура използването на нелинеен оценител с вълновидна мрежа често предоставя задоволителни резултати и се използва като модел за бърза оценка. Редът и закъснението на нелинейните ARX модели са целочислени и положителни стойности, въвеждани чрез параметрите pa , pb , pk , където pa определя броя изходни стойности за предвиждане на текущата стойност на модела, pb определя броя на входните стойности, а pk се определя закъснението от стойности.

Нелинейните ARX модели трябва да се разглеждат като разширение на линейните ARX модели на системи с един вход и един изход (SISO ARX).

Нелинейният ARX модел използва по-гъвкава нелинейна функция за описание на изследваната система

$$y_p(t) = f(y(t-1), y(t-2), y(t-3), \dots, u(t), u(t-1), u(t-2), \dots) \quad (15)$$

където f е нелинейна функция. Входните данни за f са регресорите на модела. В структурата на нелинейните модели се избира нелинейна функция за нелинейния оценител. При нелинейните ARX модели регресорите могат да бъдат входните и изходните закъснения или по-сложни техни зависимости.

Нелинейните ARX модели се представят чрез регресори и нелинеен оценител, който има линейна и нелинейна функция.

Нелинейните оценители представляват нелинейна функция като сума от последователни нелинейни елементи. Оценяващият алгоритъм във вълновидната мрежа определя броя на използваните елементите автоматично. Той използва следната формула за определяне на функцията $g(x)$:

$$g(x) = \sum_{k=1}^n \alpha_k k(\beta_k (x - y_k)) \quad (17)$$

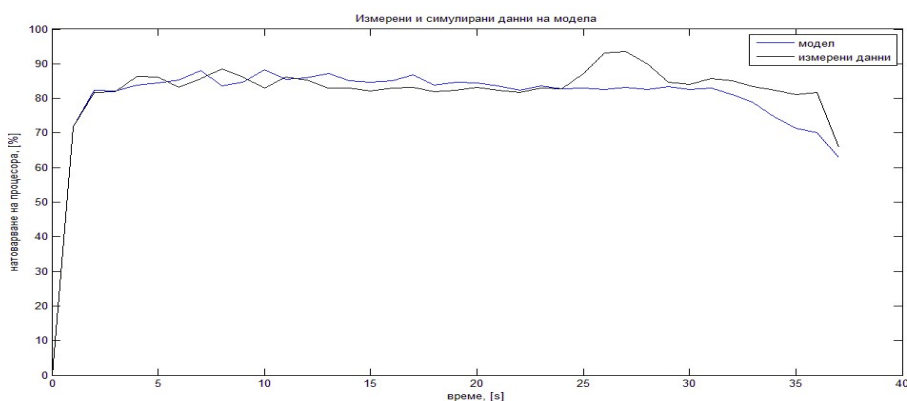
където $k(s)$ е вълновидна функция. Определянето на регресорите на модела се осъществява чрез задаване на степенния ред на полинома и закъснението на модела, които създават поредицата от стандартни регресори на модела.

В дисертационния труд използването на нелинейните ARX модели цели да се представят модели на подобренията, които са реализирани чрез следните софтуерни

приложения: TSP комуникационния протокол, конфигурационния файл на уеб сървъра Apache и програмата за автоматичното им преконфигуриране – всички работещи на операционната система Linux.

В изследваните модели за входни стойности са използвани данните от натоварването на процесора на системата преди да се извършат промените в гореспоменатите приложения, а изходните – след извършените промени. Данните, използвани за изграждане на моделите, са получени при различен брой потребители на информационна система за електронно разпространение на научно списание, когато те използват видео файл или база данни.

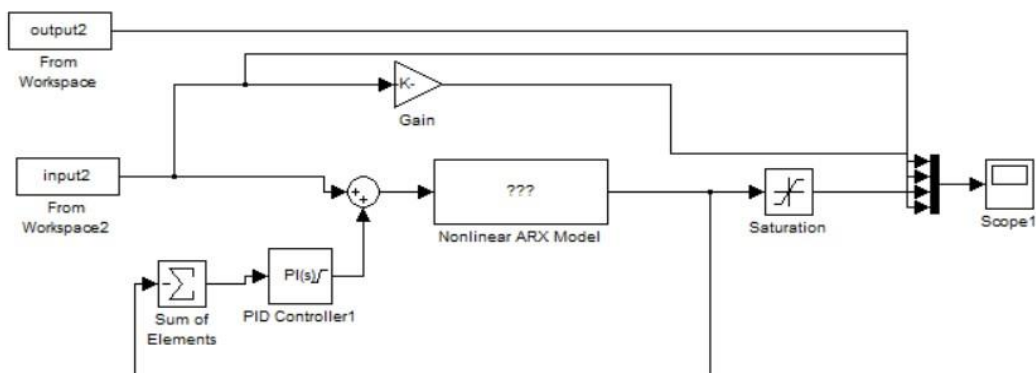
Първо е изследвано е поведението на използваната системата с нелинеен модел за трима потребители на база данните на разработеното софтуерно приложение за натоварване на процесора при обслужване им. Моделът е изграден чрез инструмента за идентификация Identification tool на Matlab. Достоверността му е 68,16%. Избраните степенни редове на полиномите са $n_a=2$, $n_b=2$, $n_k=0$.



Фиг.IV.10. Нелинеен ARX модел на изследваната система при три потребителя на база данни

Полученият модел е изследван в симулационна среда на Matlab. Проучени са резултатите от приложеното софтуерно управление и е изследвана приложимостта на линейни регулатори в използваната компютърна система. Те се прилагат, когато се изисква постигане на експлоатационни качества на процесите или при решаването на сложни задачи за управление. Задачата, която решава регулаторът, е да поддържа максимално близко управляваната променлива ($input_2$) до зададената променлива ($output_2$). Това се постига чрез сравняване на двете стойности, а на изхода на регулатора се подава манипулираната стойност. В зависимост от желаните изходни стойности, които се изисква да се постигнат след действието на регулатора, се използват различни начини за управление. За да се симулира управление в изследваната система се използват П и ПИ регулатори. Пропорционалният регулатор позволява да се генерира сигнал, пропорционален на отклонението от зададената стойност. Чрез него се постига груба (не нулева) връзка между отклонението и манипулираната променлива. Пропорционалноинтегралният регулатор позволява бърза реакция и компенсирание на смущенията в установен режим.

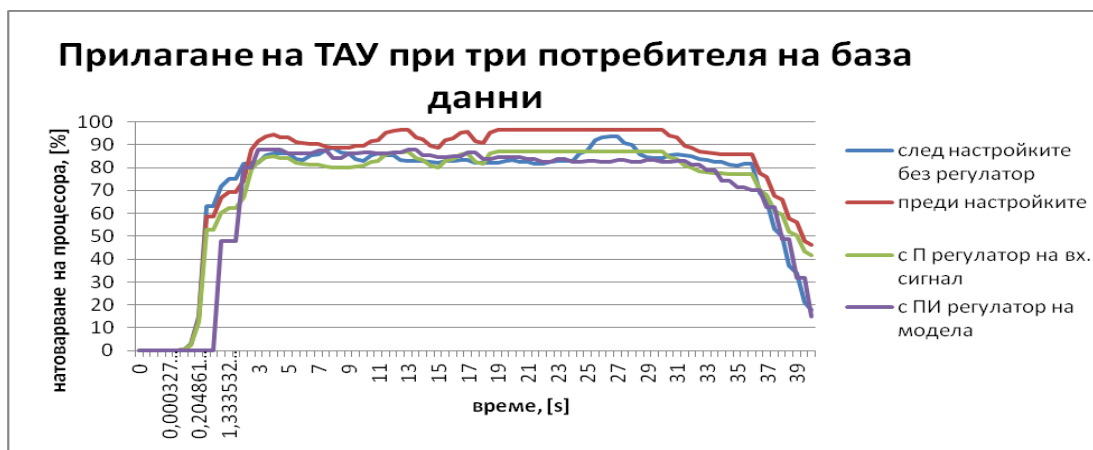
На симулационната схема (фиг.IV.11) входният (input2) и изходният (output2) сигнал са експериментално получените данни от натоврането на процесора преди и след извършените настройки в използваните програми за управление на системата. Симулирано е управление в отворена система за автоматично управление с пропорционален регулатор и е представено намаляването на натоварването на процесора. Изследва се управление в затворена система за автоматично управление с ПИ регулатор на експериментално получените данни.



Фиг.IV.11. Симулационен модел на системата

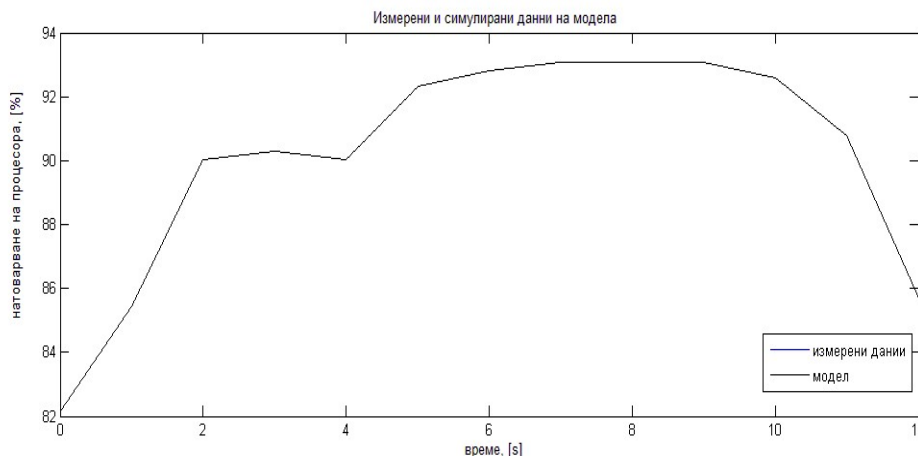
Настройката на ПИД регулатора е процес, при който се определят стойностите на П, И и Д компонентите на регулатора за постигане на желаното качество на процеса. ПИД регулаторите се настройват ръчно или въз основа на определени правила за настройка. Ръчните методи са интерактивни и изискват много време за постигане на желания резултат. Определените правила също имат ограничения – не се поддържат от неустойчиви системи, от системи от по-висок ред или от системи без времезакъснение. Затова се използва auto tuning PID controller – за да се постигне оптимално проектиране на регулатора и последният да отговаря на изискванията, когато правилата за настройка и ръчно настройване не могат да се приложат. Инструментът auto tuning PID controller подпомага интерактивното настройване на регулатора за постигане на желаните настройки. Той позволява лесно да се определят стойностите на компонентите на регулатора за постигане на желаното качество на изходния сигнал на системата.

На фиг.IV.12 са показани получените резултати от симулационния модел. Стойностите на ПИ регулатора са получени с инструмента за настройване на регулатори autotunePID. Чрез него се постига по-бърза реакция на системата и допълнително изглаждане на получения експериментално изходен сигнал.



Фиг. IV.12. Получени резултати от симулационната схема за база данни при три потребителя

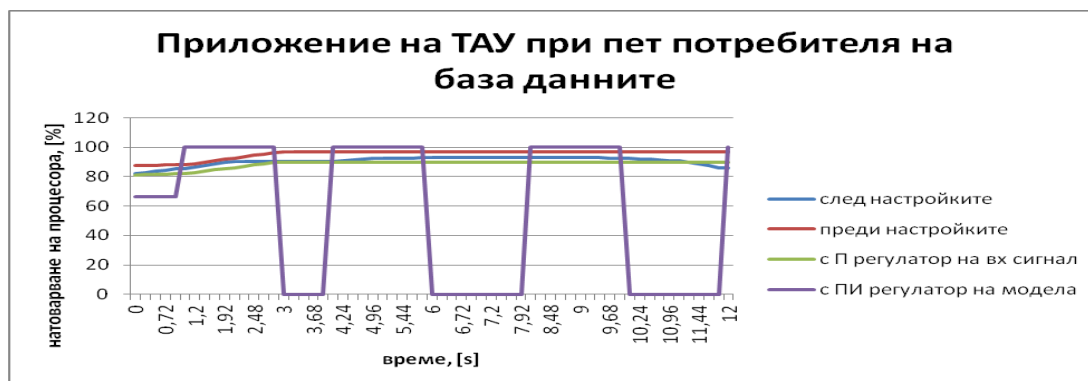
В дисертационния труд е изграден и модел при промяна броя потребители (от три на пет) на база данните на софтуерното приложение. За получаването на модела се използва нелинеен ARX модел с вълновидна мрежа за оценяване на нелийността, при промяна на броя потребители на системата. Той е изграден чрез инструмента за идентификация Identification tool на Matlab и описва промяната в системата с достоверност 100%. Избраните степенни редове на полиномите са $na=2$, $nb=2$, $nk=1$. Полученият модел е симулиран чрез симулационната схема, показана на фиг.IV.11.



Фиг. IV.13. Нелинеен ARX модел на изследваната система при пет потребителя на база данните

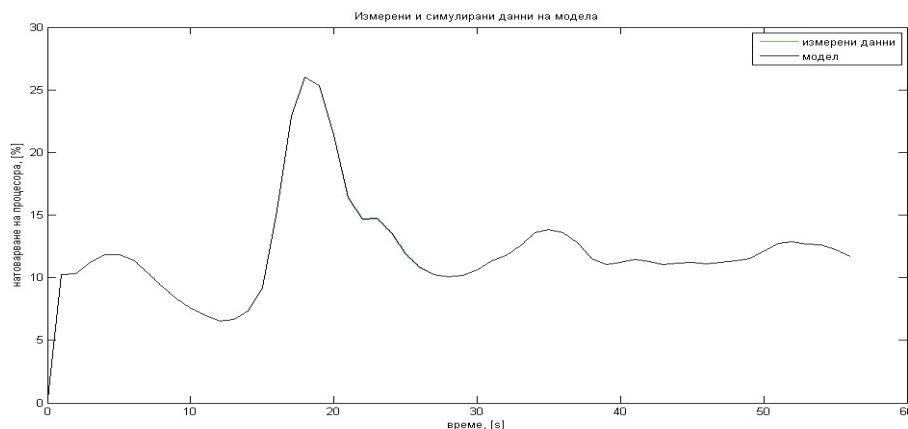
При тестване на ARX модела чрез симулационната схема става ясно, че приложението на регулатор за управление с обратна връзка не е уместно, понеже системата е много бърза и това ще доведе до затруднение при обслужването на клиенти. Друг извод, който може да се направи от фигурата по-долу, е че системата ще работи в пренатоварен режим и това ще забави обслужването на потребителските заявки. Последното показва, че при обслужване

на голям брой потребители на база данните трябва да се търси друго решение за тяхното обслужване.



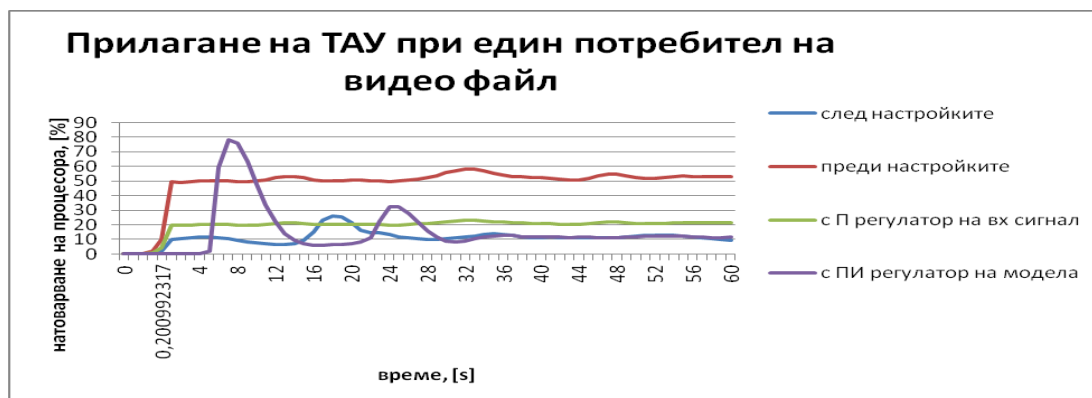
Фиг.IV.14 Получените резултати от симулационния модел при пет потребителя на база данните

Ивършени са и аналогични експерименти, но с използване на видео файл от потребители на софтуерното приложение. Изследвани са реализираните подобрения в системата чрез нелинеен ARX модел, изграден с помощта на инструмента за идентификация Identification tool на Matlab. Той описва натоварването на процесора на система с достоверност 99,66%. Избраните степенни редове на полиномите са $na=2$, $nb=2$, $nk=0$. Измерените изходни данни и симулирания нелинеен модел, получен чрез Matlab, са показани по-долу.



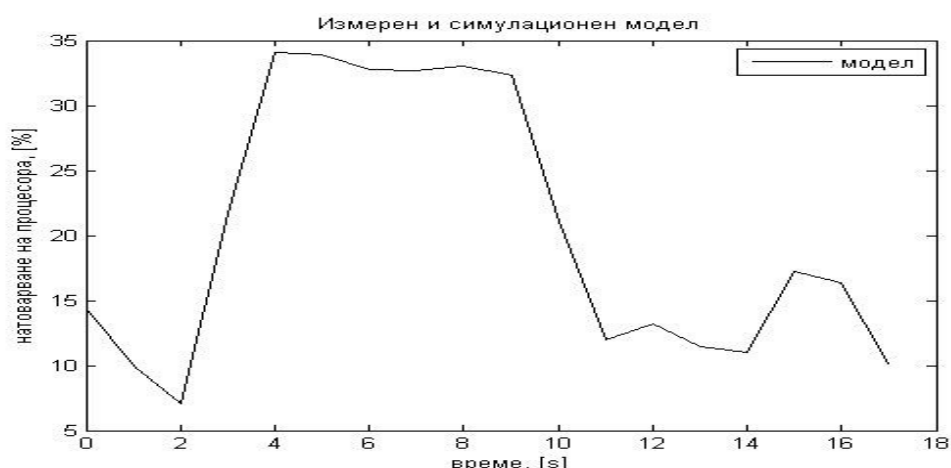
Фиг.IV.15. Нелинеен ARX модел при един потребител на видео файл

При по-нататъшно тестване чрез симулационната схема става ясно, че прилагането на ПИ регулатор към системата ще позволи по-бързото обслужване на потребителските заявки и ще допринесе за допълнително изглаждане на експериментално получения изходен сигнал.



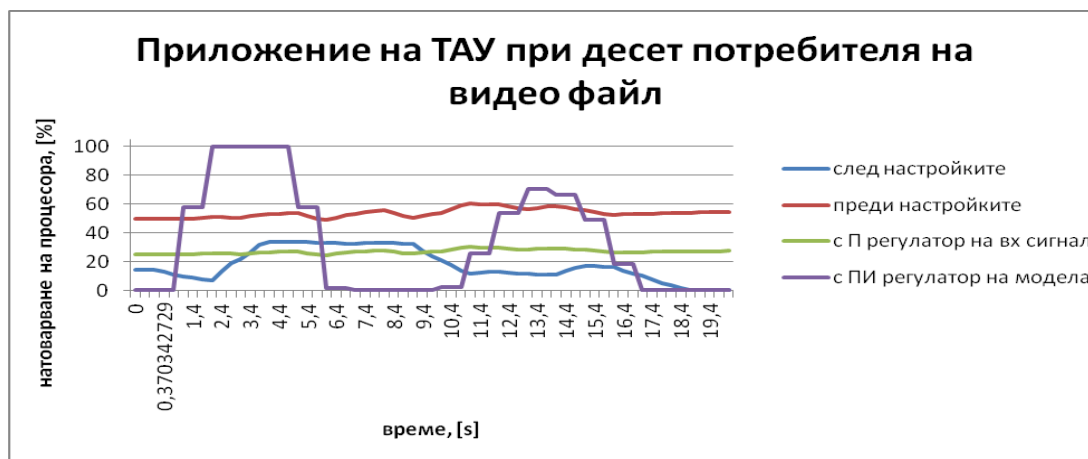
Фиг.IV.16. Получени резултати от симулационната схема за видео файл при един потребител.

Последните експерименти анализират натоварването на процесора при промяна на броя потребители на разработената информационна система (от един на десет), използващи видео файл, предоставен от нея. Нелинеен ARX модел, изграден чрез инструмента за идентификация Identification tool на Matlab, ще послужи за описание поведението на системата. Той описва натоварването на процесора на система с достоверност 100%. Избраните степенни редове на полиномите са $na=2$, $nb=2$, $nk=0$.



Фиг.IV.17. Нелинеен ARX-модел при десет потребителя на видео файл

Полученият модел е изследван чрез представената по-горе в текста симулационна схема. Резултатите от симулациите на изследваната система при десет потребителя на видео файл показват, че прилагането на ПИ регулатор към системата ще позволи по-бързото обслужване на потребителските заявки.



Фиг.IV.18. Получени резултати от симулационната схема за видео файл при десет потребителя

В обобщение, четвърта глава на дисертационния труд демонстрира приложимостта на нелинейни ARX модели, за да представи реализираното софтуерно подобрене и да разгледа възможността за допълнително управление чрез ПИ регулатори. От експерименталната работа може да се заключи, че при потребление на видео файлове е подходящо да се реализира ПИ управление на използваната система. За разлика от това, при обслужване на потребители на бази данни на информационната система се оказва, че при надвишаване на техния брой над пет, използването на ПИ управление не е уместно.

Приноси на дисертационния труд

1. Разработен е количествен модел чрез прилагане на формалния апарат на теорията на автоматичното управление за управление на сложни програмни системи.
2. Реализира се управление в изследваната програмна система като се извършва настройка на параметрите на TCP комуникационния протокол, уеб сървъра Apache и операционната система.
3. Експериментално се определят динамичните характеристики на програмната система при нейната работа.
4. Определя се ефективността на приложените настройки чрез показателите на качеството на системата.
5. Настройват се регулатори с пропорционално действие, за да се представят реализираните подобрения в работата на програмната система и с помощта на пропорционално-интегрално действие да се постигне по-добро качество на управление в затворения контур.

Апробация на резултатите

Резултати, получени в хода на работата по дисертационния труд, са оповестени и дискутирани по време на следните конференции, семинари и работни срещи:

- Работна среща (Workshop) по проект BG051PO001-3.3.06.0048 "Изграждане и развитие на млади висококвалифицирани изследователи за ефективно прилагане на биомедицински изследвания за подобряване качеството на живот" 26.04.2013г., ИЕМПАМ-БАН, бл. 25.
- International Conference Computer Systems and Technologies, CompSysTech'13, 28-29, June 2013, Rouse, Bulgaria
- Първа национална тематична школа и борса за научни идеи в областта на информационните и комуникационните технологии, 27-28 юни, 2013, гр. Русе.
- Международна конференция „Автоматика и информатика“, 03-07 октомври 2013, гр. София
- VII Национална студентска научно-техническа конференция, 23-27 септември 2011г., гр. Созопол

Резултати, включени в дисертацията са обсъждани и на работни семинари на секция „Йерархични системи“ към ИИКТ-БАН и катедра „Информатика“ при МГУ „Св. Иван Рилски“.

Списък на публикациите по дисертацията

1	[DimiT-13]	Dimitrov S., T.Stoilov. Loading test of the Apache HTTP Server by video file and usage measurements of the hardware components. Proceedings of the International Conference Computer Systems and Technologies, CompSysTech'13, 28-29 June 2013, Rouse, ACM Conference Proceeding Series, pp.59-66.
2	[Димит-13]	Димитров, Ст., Тест за натоварване на Apache HTTP Server чрез различни видове файлове и измерване използването на хардуерните компоненти, Първа национална тематична школа и борса за научни идеи в областта на информационните и комуникационните технологии, 27-28 юни, 2013, Русе, ISSN 1314-9024, стр. 212-217
3	[Димит-13а]	Димитров, Ст., Информационна система за електронен абонамент на научно списание, Международна конференция „Автоматика и информатика“ 2013 3-7 Октомври, София, ISSN 1313-1850 2013 стр. I-185 - I-188
4	[Димит-11]	Димитров, Ст., Разглеждане на Apache HTTP Server като система за автоматично регулиране и извършване на тестове с Jmeter , VII Национална студентска научно-техническа конференция 23-27.09.2011г., гр. Созопол, ISSN 1314-0442, 2011, стр. 263-269.

Литература

[ApaSF-16]	Apache Software Foundation, Apache Jmeter – User’s Manual, Building a web test plan, 2016, http://jmeter.apache.org/usermanual/build-web-testplan
[ArnoA-94]	Arnold O. A., Computer Performance Analysis with Mathematica, Academic Press, 1994. <i>\$1.1 Introduction, pg 1.</i>
[Avoya-11]	Hovhannes Avoyan, (2011), 25 Apache Performance Tuning Tips, http://www.monitis.com/blog/2011/07/05/25-Apache-performance-tuningtips/
[BoGMT-05]	G. Bolch, S. Greiner, H. de Meer and K. S. Trivedi, Queueing networks and Markov chains: Modeling and Performance Evaluation with Computer Science Applications, Wiley-Interscience, 2005.
[DGHPT-02]	Diao Y., Gandhi N., Hellerstein J.L., Parekh S., Tilbury D., Using MIMO feedback control to enforce policies for interrelated metrics with application to the Apache Web server. IEEE/IFIP Network Operations and Management, April 2002
[Garre-09]	Russ Garrett, (2009), Linux Kernel Tuning, http://russ.garrett.co.uk/2009/01/01/linux-kernel-tuning/
[GheGF-13]	Gheorghe G. F., Web Application Performance Testing using Apache JMeter2 March 2013, http://grosan.co.uk/web-application-performancetesting-using-Apache-jmeter-part-2-monitor-server-resources/
[Gregg-13]	Gregg B., Systems Performance: Enterprise and the Cloud 1st Edition, Prentice Hall, 2013, ISBN-10: 0133390098
[HTCAG-10]	High Throughput Computing Administration Guide, (2010) https://computing.llnl.gov/linux/slurm/high_throughput.html
[MDVHR-01]	R. D. Van der Mei, R. Hariharan and P. Reeser,” Web Server Performance Modeling, “Telecommunication Systems, Vol. 16, No. 3-4, 2001, pp. 361378. doi:10.1023/A:1016667027983
[Moura-00]	G. Mourani, (2000), Securing and Optimizing Linux RedHat Edition -A Hands on Guide, chapter 6 Linux General Optimzation, http://tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Editionv1.3/chap6sec75.html
[OracC-00]	Oracle Company, Oracle9i Application Server Oracle HTTP Server powered by Apache Performance Guide, 2000, https://docs.oracle.com/cd/A95427_01/httpperf/listener.htm
[SoftP-15]	SoftPanorama, Linux TCP Performance Tuning, 20 october 2015, http://www.softpanorama.org/Commercial_linuxes/Performance_tuning/TCP_performance_tuning.shtml
[TricE-11]	Trichkova, E., “PHPbasedsitetemplatefordistancelearninginmedicaldisciplines”, ProceedingsofELearning'11 conference - E-LearningandtheKnowledgeSociety, 25-26 August 2011, Bucharest, Romania, pp. 87-92, ISBN 978-606-505-459-2
[WigeN-13]	Wiger N, Linux Network Tuning for 2013, 6 Apr 2013, http://www.nateware.com/linux-network-tuning-for-2013.html

	[Гарип-07]	Гарипов, Е., Идентификация на системи: I част (Идентификация чрез непрекъснати модели) (трето допълнено и преработено издание).ТУ - София, 2007, 133 стр
	[СапуГ-07]	(Георги Сапунджиев, Информационни системи в индустрията ISBN 954-438-481-2)

Abstracts of Dissertations

Number 7, 2017

INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGIES
BULGARIAN ACADEMY OF SCIENCES

БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО ИНФОРМАЦИОННИ И КОМУНИКАЦИОННИ ТЕХНОЛОГИИ

Брой 7, 2017

Автореферати на дисертации