

A power quality monitoring system based on MATLAB server pages

Bülent VURAL, Ali KIZIL, Mehmet UZUNOĞLU

*Department of Electrical Engineering, Yıldız Technical University,
34349 Beşiktaş, İstanbul-TURKEY*

e-mail: bvural@yildiz.edu.tr, akizil@yildiz.edu.tr, uzunoglu@yildiz.edu.tr

Abstract

The power quality (PQ) requirement is one of the most important issues for power companies and their customers. Continuously monitoring the PQ from remote and distributed centers will help to improve the PQ. In this study, a remote power quality monitoring system for low voltage sub-networks is developed using MATLAB Server Pages (MSP). MATLAB Server Pages, which is an open source technical web programming language, combines MATLAB with integrated J2EE specifications. The proposed PQ monitoring system uses the MATLAB HTML tag library and the data captured by the DAQ hardware to evaluate the Power Quality parameters. The proposed system can be modified as technology and evaluation methods improve.

Key Words: Analyzer; data acquisition; harmonics; microcontroller; PQ monitoring; MSP; power quality.

1. Introduction

Power quality (PQ) has acquired much interest in recent years because of the increasing concern for supplying clean electrical energy to the consumers in the existence of non-sinusoidal waveforms. Since modern electrical systems are largely based on control units with microcontrollers, disturbances in PQ become more important due to the effects of these disturbances on the safe operation of control units [1-3]. Anything unfavorable (e.g. corruptive effects of power electronics and non-linear loads) in the power system can be defined as power quality disturbances [3-5]. The purpose of power companies is to provide solutions to enhance their power quality, in specific, supply continuity and voltage quality [6,7]. A proper understanding of the power quality can help companies to determine an appropriate approach to improve PQ [8].

Power quality monitoring is a process that is based on gathering voltage and current data, transporting them to remote centers, and converting them into decision-making information [9-10].

PQ monitoring systems continuously measure and analyze the power quality, and can carry out additional functions, such as communication with an internet network, statistical analysis through uninterrupted measure-

ments, and web-based deployment. Generally, the PQ monitoring systems are classified in three generations, which vary according to functions performed [1]. In the first generation, the functions are displaying electrical parameters (voltage, current, and power, etc.) and evaluating power quality (option). In the second generation, the functions are event detecting, PQ parameter calculation, serial communication and feature display. Finally, the functions in the third generation are PQ parameter calculation, communication and web-browsing features.

There are already some studies related to third generation PQ monitoring systems in the literature. Leou et al. [11] presented a database management system to store power quality data that utilized a user-friendly interface. Active Server Pages (ASP) were used to implement the web-to-database application and enable users to monitor the PQ of the system through the browser. Chen, et. al. [12] examined various issues of the network for PQ monitoring. They described the design of a multi-channel monitoring system, and the various requirements that a monitoring instrument must satisfy to achieve these objectives. However, as defined in [13-15], the comparable systems in Ref. [11, 12] have disadvantageous characteristics, such as high network bandwidths, extremely large data storage requirements, and long computation times. In Ref. [16], a concept for a power quality monitoring system based on an intelligent, adaptive, and reconfigurable multi-agent system (MAS) was reported. Although advances in network-based PQ monitoring systems provide system-wide insights on power quality monitoring, problems such as transmitting raw data to remote centers and large memory requirements to store that data are still major setbacks. Also, the need of high computing power to process large amount of raw data causes difficulty in PQ monitoring.

In this paper, Matlab server pages (MSP) are developed as the core of the system proposed in this study, and are used for web-browsing, database management and monitoring the power quality from a remote center through internet networks. Since the proposed system only transports the results of raw data (voltage and current) analysis, high network bandwidth is not required. Similarly, a large amount of data storage is also not required. Furthermore, the use of an embedded PC, rather than the usual DSP-based PQ analyzer, shortens the computation time.

Section 2 discusses the technical aspects of proposed system from the point of view of both methodology and measuring equipment. In section 3, experimental results are discussed. Finally, conclusions are given in section 4.

2. System description

As shown in Figure 1, the proposed PQ monitoring system consists of many parts: transducers, data acquisition (DAQ) hardware, embedded PC, and MSP software for remote access, monitoring and data base management. The technical details of each part will be discussed in the following sub-sections.

The system first samples and collects current and voltage data from a measurement point by means of transducers and DAQ hardware. After the DAQ hardware gathers the raw data, the data is sent to the embedded PC through a communication protocol, either hardwired or wireless. Then, the embedded PC regulates and analyzes the data with a MATLAB-file and a MATLAB & Simulink model. After all processes are completed, the results of analysis are sent to a database within the MSP. After this point, end users can reach those results over the internet network.

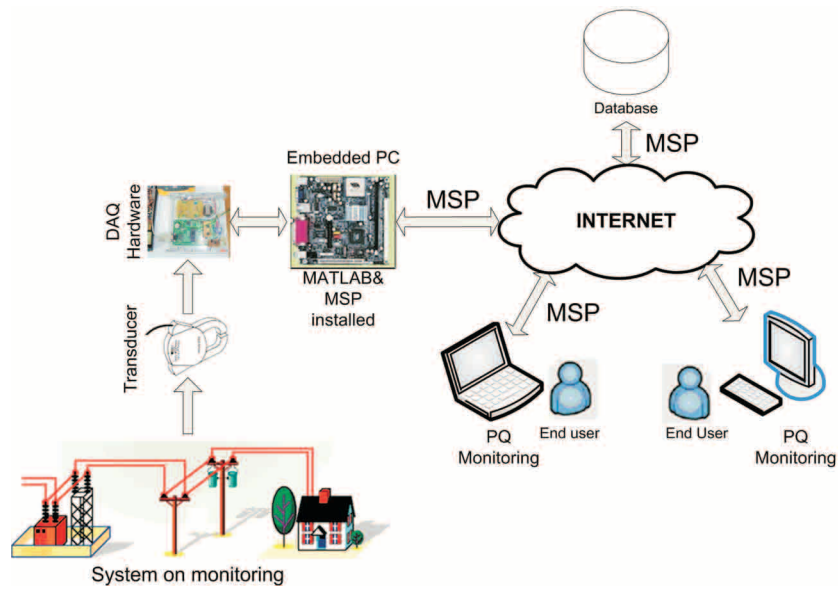


Figure 1. Block diagram of the overall PQ monitoring system.

2.1. MATLAB server pages (MSP)

MATLAB is a high-level technical computing language. Its user-friendly syntax, rapid prototyping ability, well-written documentation and all-in-one platform features has made it preferred by many users. Moreover, users can take advantage of built-in compilers to generate stand-alone applications. However, nowadays companies demand N-Tier applications, which mean that users want to drive MATLAB as a computation engine from other popular programming languages, i.e., Java, C#, C++. For example, consider user interfaces, such as a web browser or a stand-alone file to run processes, which send retrieved user commands to the application server, where MATLAB runs. However, if any databases or other routines are used, these interfaces must also call procedures on other servers, such as a “Database Server” or “Business Logic Server”.

MATLAB has documentation relating to external interfaces [17], which describe how to call MATLAB from C/C++/Fortran/Java & C#. Developers can easily access MATLAB through Component Object Model (COM) [18] and Dynamic Data Exchange (DDE) [19] support at Microsoft Windows based platforms. Also, there are built-in C routines to call MATLAB from other languages. These routines are accessible for both Microsoft Windows and all POSIX-based (Linux, UNIX like) operating systems. C and C++ languages are not well formed, and therefore not a rapid choice for developing N-Tier applications (e.g., Remote Method Invocations and Web Services), such as C# [20]. As long C# on .NET Framework is seen as a suitable solution, C# can only implement C or C++ routines in unsafe blocks, which cause stack overflow [21]. As a satisfactory solution, Java Native Interface (JNI) [22] technology can wrap C routines in Java.

On the other hand, MATLAB has a product called a “MATLAB Compiler” to generate runtime-based [23] DLLs, C# Assemblies or Java classes to use MATLAB code in C, C++, C# or Java. The main purpose is to reduce development time and facilitate easy code management. However, the MATLAB Compiler does have limitations and restrictions [24]. The developer may face challenges in the solution. There are some approaches that use C routines of MATLAB to overcome this issue. One is called the MATLAB Web Server toolbox

[25]. MATLAB Web Servers use a Common Gateway Interface (CGI) [26] to transfer data from the web to MATLAB. Because CGI is a legacy technology, there are several deficiencies in this solution, such as no database connectivity support, no web service architecture, no interaction with other languages like JavaScript [27], and hard coding. Ohio State University has developed “A Java Based Web Interface to MATLAB” [28] solution, which uses C routines in Java. Although it seems to be a good candidate having Kerberos authentication [29], and audio/video playback on HTML and dynamic zoom for images, it has a hard syntax to code because of HTML Embedded Servlets, and that it only supports POSIX based-systems with no built in database support, and has no interoperability and web service functionalities. Without an N-Tier architecture approach serving rapid application development environments, it cannot level demands.

To overcome the above mentioned limitations, we present a solution that uses JNI [30] and supports N-Tier architecture, which will be suitable for developer’s demands. In this paper, MATLAB Server Pages are developed as a Java based, open source, platform independent application to satisfy developers’ expectations. MSP offers a web-based application development platform, including database connectivity, Remote Method Invocation, Distributed Computing, and Web Services. Not only can MATLAB users use these products, web developers and application engineers can now prefer MATLAB for N-Tier architectures. In short, MATLAB Server Pages is technical web programming language that uses MATLAB in the background via JNI.

2.1.1. System technical architecture

MSP is an open source technical web programming language that uses MATLAB in background. MSP supports “3-Tier Architecture”: Web Tier, Business Tier and Database Tier. Also, MSP consists of “distributed computing” and “parallel processing” via remote procedure calls and web services.

MSP uses Java Server Page (JSP) [31] technology to define Custom Tag Libraries [32], called MATLAB Tag Library Definitions (TLDs) & MATLAB Remote Method Invocation (RMI) TLDs, to handle rapid application development (RAD) environments for web developers. Underlying Java RMI support acts as a gateway for distributed computing and parallel processing. Thus, MSP can use remote MATLAB just like using MATLAB on a local computer. The Expression Language Mechanism [33] of the JSP Standard Tag Library [34] and MSP Custom Tag Libraries work together to realize database connectivity. MSP implements JavaServer Faces [35] for defining business logic and business rules, including advanced UI components. MSP includes Web Service Definition Language (WSDL) [36] files that provide an opportunity for using MATLAB as a Web Service. MSP is compatible with Business Process Execution Language (BPEL) [37] and Web Services Interoperability (WS-I) [38] standards.

As shown in Figure 2, the MSP architecture can be summarized as follows:

- MATLAB Server Pages connect to MATLAB with a Java Wrapper Class called JMatLink developed by Stefan Mueller [17]. MSP Java classes drive JMatLink for JavaServer Pages by adding new methods for MATLAB features, JSP page output, imaging on browsers, remoting, and web services.
- MSP implements JSP and JSTL to define the MATLAB Custom Tag Library, which includes support for expression language and scriptlets. This enables developers to call MATLAB from a JSP page using HTML tags. MSP main classes are declared as remote classes. It is possible to register MSP classes and bind them from remote classes. Marshalling serializes the methods and properties of MSP classes to a remote site, while unmarshalling deserializes them at remote site. MSP includes FTP client classes to

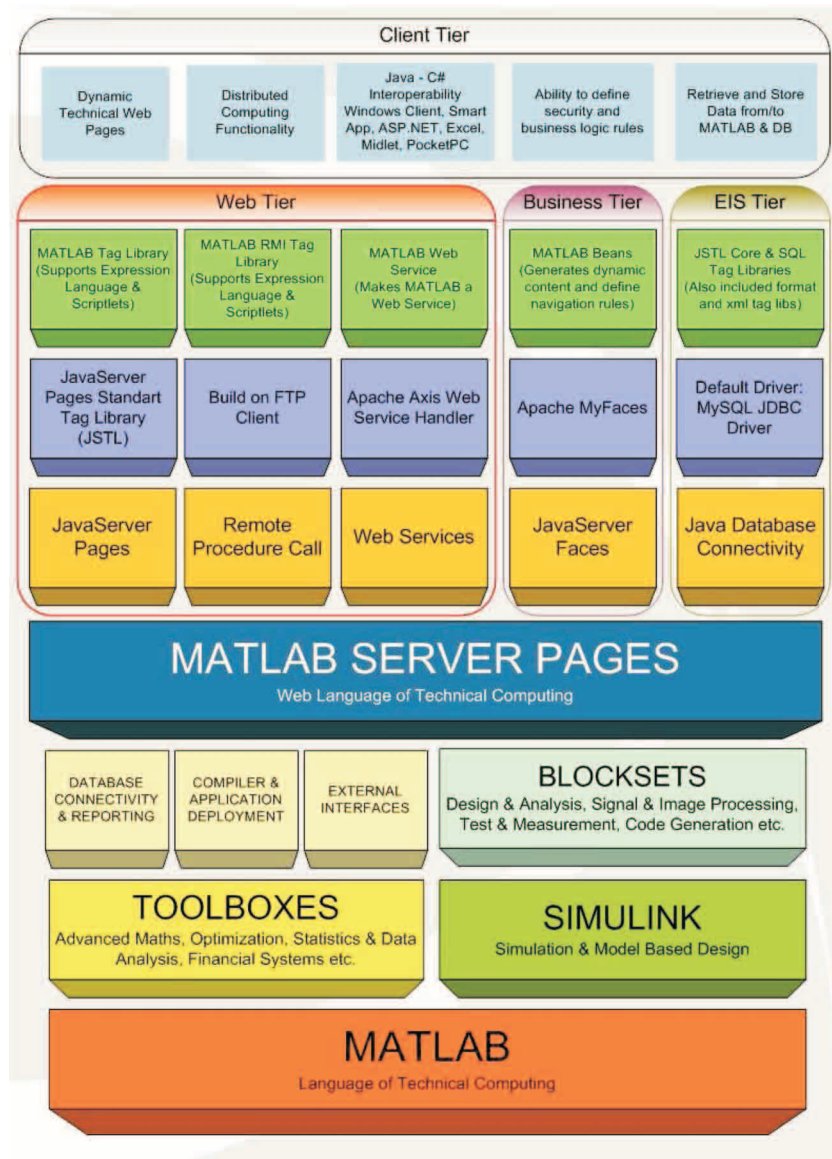


Figure 2. Contribution of all tiers creates MSP Architecture.

pass and retrieve files to and from a remote computer. This is designed for transferring images, MATLAB files and outputs without serialization. Both sides run a web server. If needed, the MSP FTP client puts necessary MATLAB files on the server and executes commands remotely. The generated output, besides images and HTML pages, are transferred back to the local computer. Images and HTML Pages are displayed on the local computer using a HTTP protocol. This eliminates serialization problems. End users and developers do not need to know about these background processes. There are ready custom tags that handle these operations. This custom tag library is called the MATLAB RMI Tag Library. Apache Software Foundation's Axis project has some tools for Java 2 WSDL generations. MSP contains Axis libraries to generate Web Services for its MSP classes. There are two types of WSDLs in MSP. Nowadays,

the WS-I community tries to minimize the interoperability problems of web services. MSP has WSDLs for both Java and C# proxy class generators. All of these implementations create a Web Tier. MSP Web Tier can stand-alone from the application server using MSP RMI features.

- By default, MSP comes with MySQL JDBC driver. The JSP Standard Tag Library has a SQL tag library, which can run SQL expressions on the database server. The Expression Language Mechanism handles MSP MATLAB Tag Library and JSTL SQL Library interactions. From MSP, developers can connect to a database server. They can retrieve data from MATLAB and store it to a database, and vice versa. This is the Database Tier – EIS Tier of MSP.
- JavaServer Faces is a technology that has advanced user interface components for JSP and can define navigation rules for business processes by running beans in the background. MSP uses the Apache Software Foundation MyFaces implementation. The pages that use JSF have a MLSPX extension in the MSP. The default extension is MLSP for MSP. MLSPXs can call beans and do operations defined in them. Developers can add their beans to MSP. MATLAB Beans works like Plug-in Logic. If developers use remoting in MATLAB Beans, they can separate business logic and business rules to another server. This puts “Business Tier” forward.

MSP has several demos and examples on how to use the MSP Engine. Client systems can be dynamic web pages, or part of distributed computing and web service operations. These types of applications are termed the “Client Tier”. The communication between all the tiers (Web Tier, IES Tier, Business Tier and Client Tier) is illustrated as shown in Figure 3.

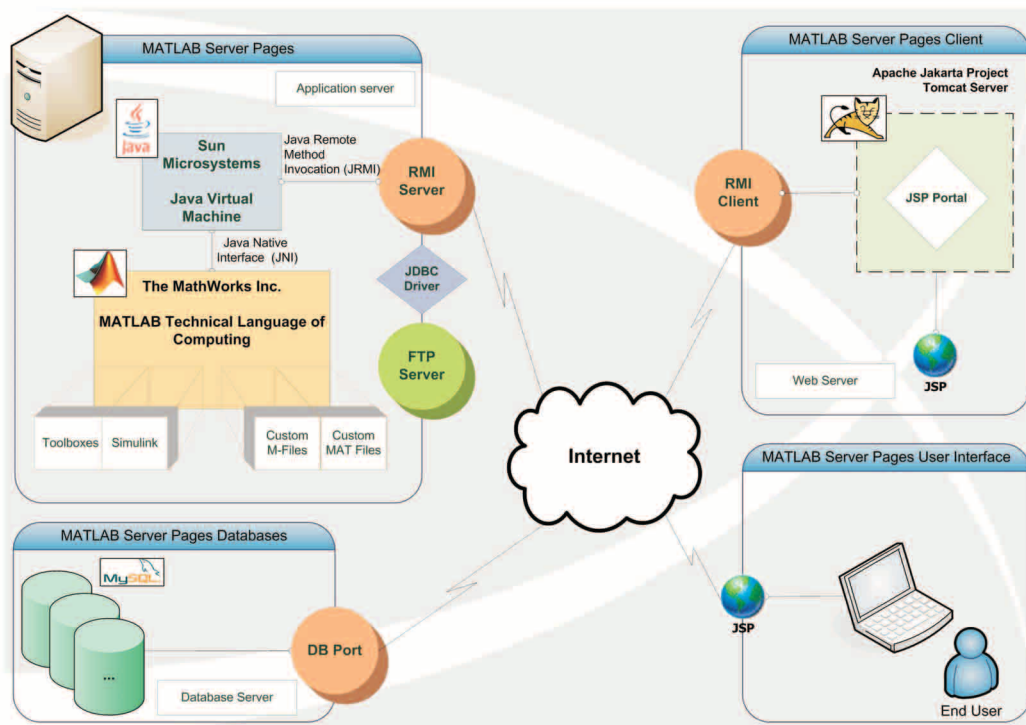


Figure 3. Overview of MSP 3-Tier Architecture.

MATLAB is installed on an Application Server Layer. There are MSP Main Classes on the PC to handle data exchange over RMI with the Web Server. Main Classes have several methods to handle custom JSP Tags at the Web Server. For example:

```
<matlab:Engine> tag on Web Server runs;
public void engineOpen() {
    // Reads MSP settings from Bundle.
    ResourceBundle rb = ResourceBundle.getBundle("MSP");
    // Code Dir is the base for MATLAB M-Files, Simulink Models and
    // Matlab Server Files. All runnable codes and models must be
    // there.
    Code_Dir = rb.getString("Code_directory");
    // Image Dir is the base for MATLAB created images.
    Image_Dir = rb.getString("Image_directory");
    // System out out for Java debugger.
    System.out.println("MATLAB Server Pages Engine");
    javaLink = new JMatLink();
    javaLink.engOpenSingleUse();
    // Running MSP at Code Dir.
    javaLink.engEvalString("cd '" + Code_Dir + "'");
}
```

This class handles all RMI operations, database connectivity, distributed computing, and web services.

Web Server Layer acts as a RMI Client to connect to the Application Server. This Web Server has a Remote Method Invocation Interface to MSP Main Classes located at the Application Server. This interface includes all methods in these classes. So applications on this server can use all functionality of MSP as though they were running locally. Programmers can deploy applications on this server with an easy syntax:

Test.mjsp

```
-----
<html>
<head>
< %@ include file="/Scripts/header.inc" %>
<title> Hello World! </title>
</head>
<body>
<matlab:Engine>
<matlab:Command cmd="rand(1,10)"/>
<matlab:WriteData name="ans"/>
</matlab:Engine>
</body>
</html>
```

The Web Server connects to the Database Server Layer to exchange data using JSTL SQL Library and

MSP Tag Libraries. Also end users, who involve the Client Tier Layer, can connect to the Web Server Layer via a web browser or using a web service.

2.2. DAQ hardware

The DAQ hardware (Figure 4) consists of three parts: a power supply, transducers and main board. The power supply provides voltage levels of -5, -12, 0, +5, and +12 volts for the main board. Transducers capture the current and voltage values, and scale them down. The main board has signal amplifiers, signal offset circuits, zero crossing detectors, communication and microcontroller units. The amplifiers and offset circuits put the signals into suitable forms. Especially in low current measurements, the signal from the Hall-effect current transducer (input 0-200A, output 0-5V) should be amplified for correct measurement. Zero crossing detectors capture the zero crossing of signals to determine the starting time of sampling. The serial communication unit, which has a RS232 protocol and 19200 baud rate, provides data flow between the microcontroller and embedded PC. In addition, an 8-bit microcontroller, which has 368 bytes of RAM and a 20 MHz operating frequency is used in order to first convert the analog signals to digital signals, and then to send the data to the embedded PC via the communication unit. The analog-to-digital converter (ADC) module on the microcontroller has 10-bit resolution and a 1 MHz sampling rate.

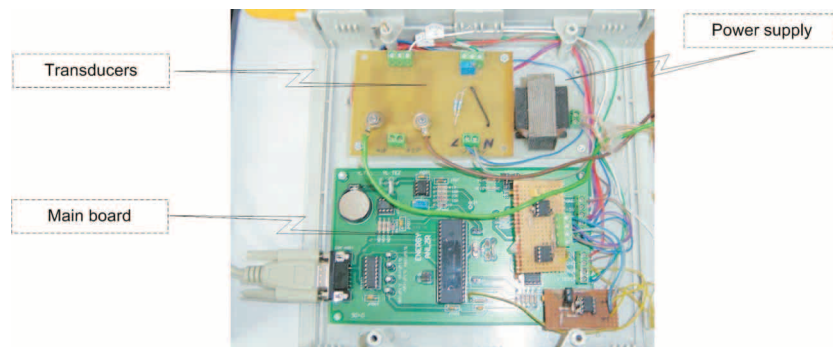


Figure 4. The DAQ hardware.

The DAQ hardware captures and samples the voltage and current signals. Then, these signals are sent to the embedded PC after a request from the embedded PC.

2.3. Analysis of the PQ with embedded PC

The embedded PC processes and analyzes the data with a MATLAB & Simulink model, as shown in Figure 5. The results of the analysis are stored in the PC memory. The model evaluates the percentage of total harmonic distortion (THD%), the DC magnitude, the percentage of harmonic orders and harmonic amplitudes, active power, reactive power, power factor, crest factor and displays the transients. Users can reach these results with the date of measurement through the internet by using MSP at any time.

The data taken from the DAQ hardware is transferred to the MATLAB workspace for PQ analysis. As seen in Figure 5, the main block includes subsystems for each PQ parameter. Existing blocks in MATLAB & Simulink are used for PQ analysis; for example, the MATLAB & Simulink FFT block is used for analyzing harmonics. After the data is analyzed, the results are represented and recorded in the database. Finally, MSP is used to deploy the analyzed results to end-users through the internet.

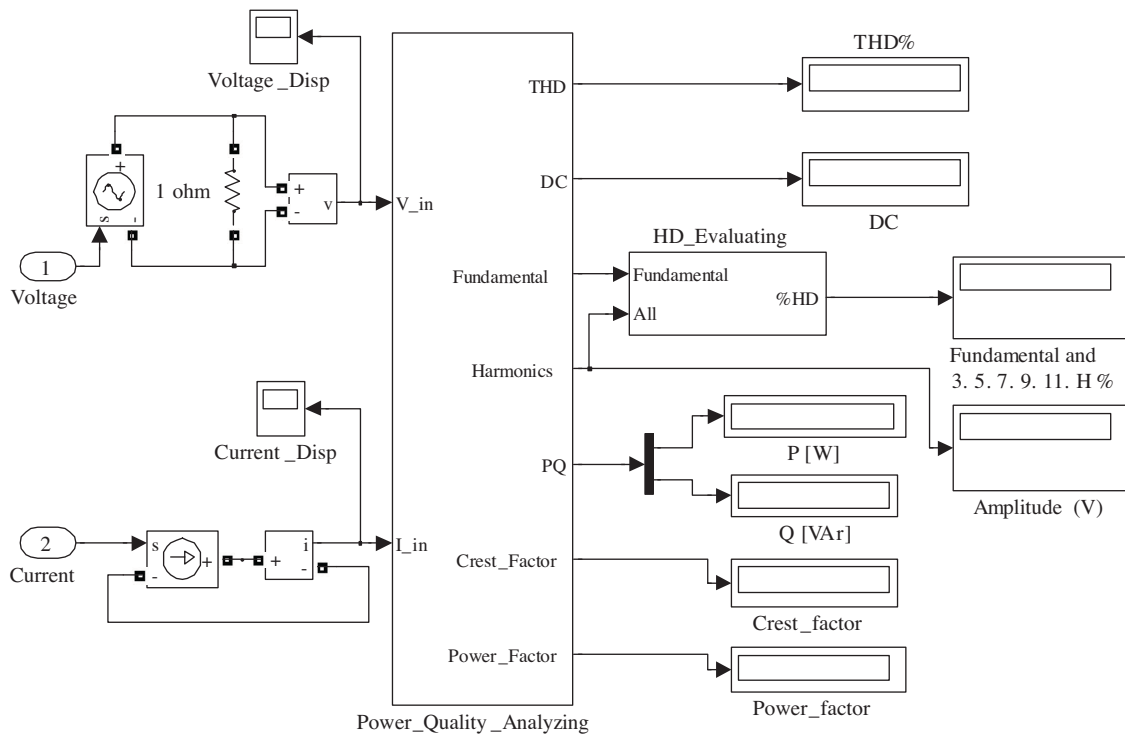


Figure 5. MATLAB & Simulink model for analyzing Power Quality.

3. Experimental results and discussions

The experimental setup of the proposed system is shown in Figure 6. Several parameters can be used to define the PQ. However, voltage quality is the most significant parameter of PQ [5]. Thus, we focused on voltage quality. The analyzed results of the proposed system are compared with actual measurements obtained using FLUKE 434[®]. Both FLUKE 434 and the proposed PQ monitoring measurements are performed simultaneously for single-phase. The PQ monitoring system and FLUKE 434 were controlled from same computer in such a way that both measurements were obtained at the same time, as required for exact comparison.

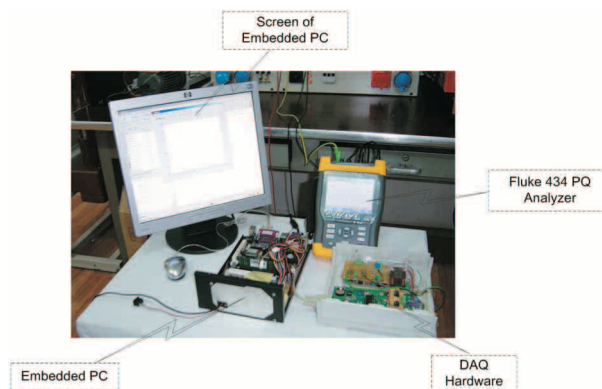


Figure 6. Test experimental setup.

In Figure 7, voltage waveforms from the PQ monitoring system and FLUKE 434 are given. First, voltage waveforms from the proposed PQ monitoring system and FLUKE 434 are compared. As can be seen from Figure 6, the waveforms are similar. Here, we sampled one period of the voltage waveform with the proposed system because of the low memory and low sampling resolution of the microcontroller on the DAQ hardware.

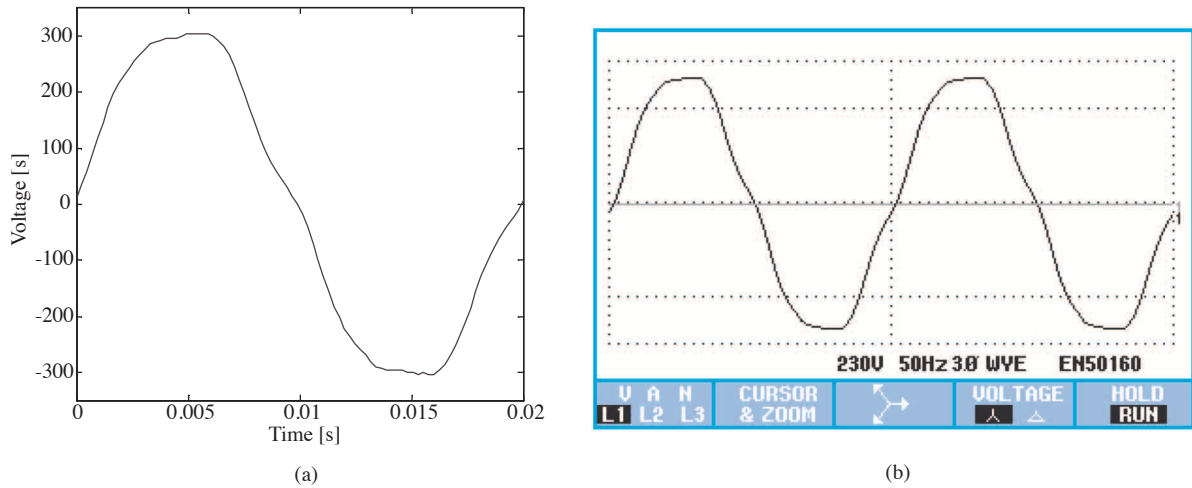


Figure 7. Voltage waveforms from (a) the proposed PQ monitoring system and (b) FLUKE 434.

In Table 1, analysis results for some PQ parameters are given. Here, harmonic analysis is performed up to the 11th harmonics. The resolution of sampling is not enough for higher harmonic levels. But as explained before, the system has a flexible structure and the DAQ hardware can be easily configured with a high-level microcontroller, microprocessor or DSP. Additionally, large memory can be used for obtaining a large quantity of measurements. The analyzed PQ parameters could be increased with high-resolution sampling rates and high performance microprocessors with larger memory spaces.

Table 1. Comparison of PQ measurements.

Power	Proposed PQ System	FLUKE 434	Total Harmonic Distortion (%) and Harmonic Orders	Proposed PQ System	FLUKE 434	Parameters	Proposed PQ System	FLUKE 434
Active Power	6.41 [kW]	6.41 [kW]	THD	5.97	5.9	DC Magnitude of Voltage	0.39	0.3
Reactive Power	0.05 [kVAr]	0.05 [kVAr]	3 rd h- order	0.51	0.5	Crest Factor of Voltage	1.40	1.4
			5 th h- order	5.56	5.5			
			7 th h-order	2.01	2.0			
			9 th h-order	0.44	0.4			
PF	0.99	0.99	11 th h-order	0.13	0.1			

In Figure 8, the results are given as a web page of the proposed PQ monitoring system. Using the web page, results can be stored or sent with an e-mail, and the analysis model can be viewed.

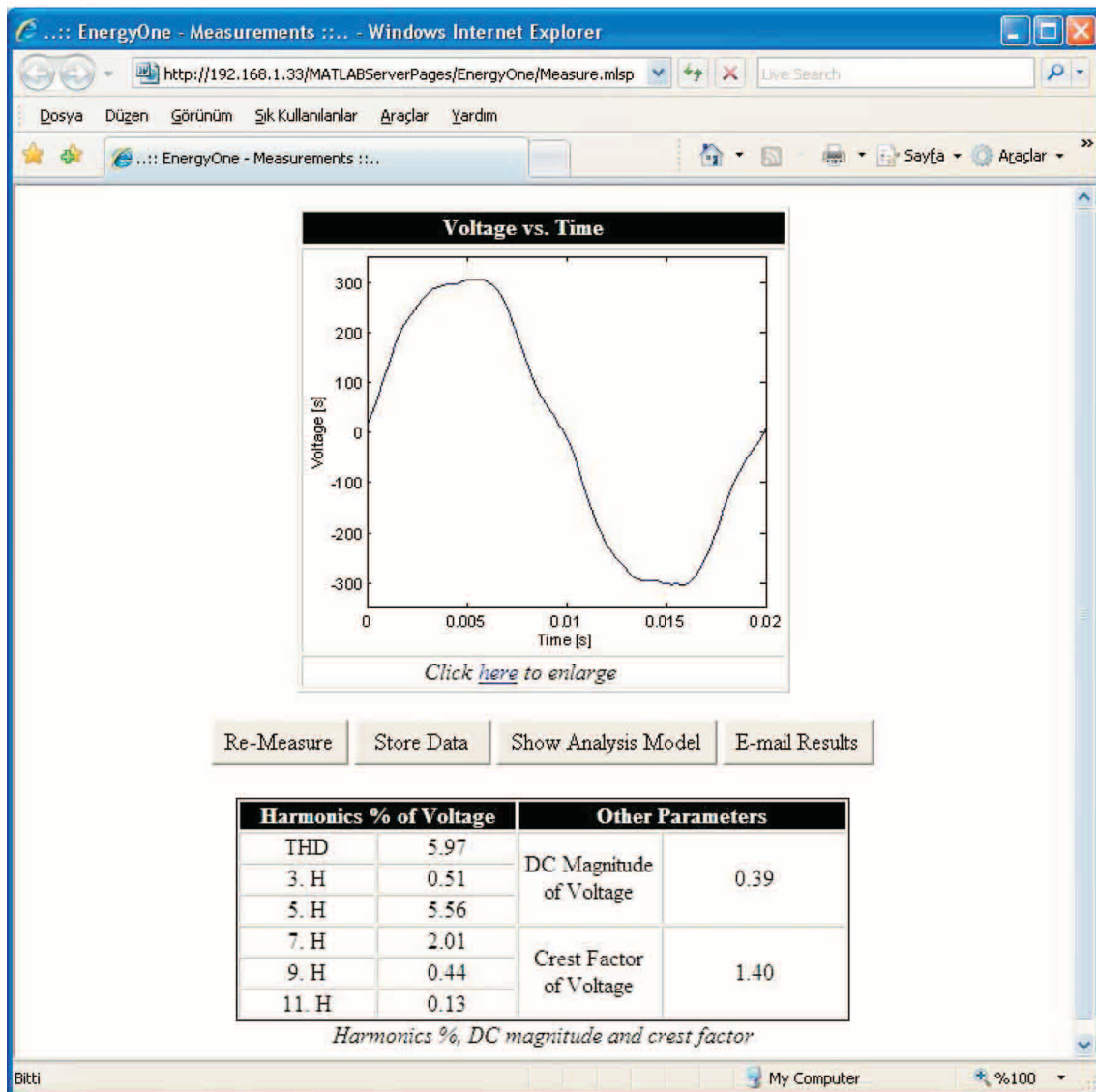


Figure 8. Web page seen by the end user.

4. Conclusions

In this study, a power quality monitoring system based on a MSP web-to-database application is implemented and good performance is achieved. Since MATLAB Server Pages (MSP) are developed as a Java based, open source, platform independent application, the proposed system can be easily adapted to new changes and improvements. Furthermore, tools developed for MATLAB can easily be integrated into the proposed system. MSP offers a web-based application development platform, including database connectivity, remote method invocation, distributed computing, and web services. Therefore, the end users can monitor the status of PQ through the internet by typing the HTML address of the system without any additional programs or equipment. In future studies, we plan to further develop the MATLAB & Simulink model and the DAQ hardware using a

high-performance processor to improve the PQ monitoring system. The proposed PQ monitoring system can also be extended for use with other technologies, such as GSM networks and cellular phones.

Acknowledgements

This work was supported in part by the research funds of Yildiz Technical University under Grant 24-04-02-01.

References

- [1] D.-J. Won, Il-Y. Chung, J.-M. Kim, S.-Il Moon, J.-C. Se, J.-W. Cho, "Development of power quality monitoring system with central processing scheme", in: IEEE Power Engineering Society Summer Meeting, Vol. 2, pp. 915 – 919, 2002.
- [2] L. Angrisani, P. Daponte, M. D'Apuzzo, A. Pietrosanto, "A VXI power quality analyser implementing a wavelet transform-based measurement procedure", *Measurement*, Vol. 26 (2), pp. 91-102, 1999.
- [3] R. Sasdelli, A. Ferrero, A. Menchetti, L. Peretto, "Electric-power quality measurements under distorted conditions: Are they utopia or reality?", *Measurement*, Vol. 23 (4), pp. 257-264, 1998.
- [4] S. Chen, XL. Wang, "Embracing advance information technology in power quality monitoring", in: IEEE Power Engineering Society Winter Meeting, Vol. 2, pp. 1466 – 1471, 2002.
- [5] M. H. Bollen, "What is power quality?", *Electric Power Systems Research*, Vol. 45 (3), pp. 5-14, 2004.
- [6] A. McEachern, "Roles of intelligent systems in power quality monitoring: past, present, and future", in: IEEE Power Engineering Society Summer Meeting, Vol. 2, pp. 1103 – 1105, 2002.
- [7] R. Lima, D. Quiroga, C. Reineri,, F. Magnago, "Hardware and software architecture for power quality analysis." *Computers & Electrical Engineering*, Vol. 34 (6), pp. 520-530, 2008
- [8] P. Di Daponte, M. Penta, G. Mercurio, "TransientMeter: a distributed measurement system for power quality Monitoring." *IEEE Transactions on Power Delivery*, Vol. 19 (2), pp.456–463, 2004.
- [9] R-C. Leou, W-R. Tsai, Y-N. Chang, "A power quality monitoring system based on J2EE architecture." In: IEEE Region 10 Conference, Vol. 3, pp.291– 294, 2004.
- [10] D. J. Won, Il-Y. Chung, J-M. Kim, S-II Moon, J-C. Seo, J-W. Choe, "Development of a network-based power quality diagnosis system." *Electric Power Systems Research*, Vol. 77 (8), pp.1086-1094, 2007.
- [11] R-C. Leou, Y-C. Chang, J-H Teng, "A Web-based power quality monitoring system." In: IEEE Power Engineering Society Summer Meeting, Vol. 3, pp.1504–1508, 2001.
- [12] S. Chen, C. L. Zhang, Y. Z. Liu, "A multi-channel monitoring system for system-wide power quality measurements." In: International Conference of Power System Technology, Vol. 2, pp.953 –958, 2000.
- [13] Y. Liu, S. K. Srivastava, D. A. Cartes, "A novel architecture for power quality monitoring based on multi-agent system theory." In: Reconfiguration and Survivability Symposium. Jacksonville, FLi 2005

- [14] A. Ferrero, "Measuring electric power quality: Problems and perspectives." *Measurement*, Vol. 41 (2), pp.121-129, 2008.
- [15] M. A. Eldery, E. F. El-Saadany, M. M. A. Salama, A. Vannelli, "A novel power quality monitoring allocation algorithm." *IEEE Transactions on Power Delivery*, Vol. 21 (2), pp.768 –777, 2006.
- [16] S. Srivastava, S. Suryanarayanan, P. Ribeiro, D. Cartes, M. Stcurer, "A conceptual power quality monitoring technique based on multi-agent systems". In: *37th Power Symposium. North American*, pp.358–363, 2005.
- [17] Available: http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/apiext.pdf
- [18] Available: http://en.wikipedia.org/wiki/Component_Object_Model
- [19] Available: http://en.wikipedia.org/wiki/Dynamic_Data_Exchange
- [20] Available: <http://www.freevbcode.com/ShowCode.asp?ID=5302>
- [21] Available: http://www.codersource.net/csharp_unsafe_code.html
- [22] Available: http://en.wikipedia.org/wiki/Java_Native_Interface
- [23] Available: http://www.mathworks.com/products/compiler/compiler_support.html
- [24] Available: http://www.mathworks.com/access/helpdesk/help/toolbox/compiler/compiler_product_page.html
- [25] Available: <http://www.mathworks.com/access/helpdesk/help/toolbox/webserver/webserver.shtml>
- [26] Available: <http://www.w3.org/CGI/>
- [27] Available: <http://www.w3schools.com/js/>
- [28] Available: <http://www.ll.mit.edu/HPEC/agends/proc03/pdfs/samsi.pdf#search=%2A%20Jva%20Based%Web%20Interface%20to%20MATLAB%22>
- [29] Available: <http://web.mit.edu/Kerberos/>
- [30] Available: <http://jmatlink.sf.net>
- [31] Available: <http://java.sun.com/products/jsp/>
- [32] Available: http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/JSPTags.html
- [33] Available: <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/JSPIntro7.html>
- [34] Available: <http://java.sun.com/products/jsp/jstl/>
- [35] Available: <http://java.sun.com/javaee/jaserverfaces/>
- [36] Available: <http://www.w3.org/TR/wsdl>
- [37] Available: http://developers.sun.com/prodtech/javatools/jsenterprise/nb_enterprise_pack/reference/techart/bpel.html
- [38] Available: <http://www.ws-i.org/>