FPGA-Based Urinalysis Using Principal Component Analysis

C. Llorente, J. Cualquiera, K. Loriaga, B. Macaspac, P. Roxas, K. Ybanez
Department of Electronics and Communications Engineering, De La Salle University, Manila, Philippines
cesar.llorente@dlsu.edu.ph

***Abstract*—Urinalysis is considered to be a common test performed in laboratory in order to diagnose Urinary Tract Infection (UTI). It undergoes three stages, which include macroscopic, dipstick, and microscopic analysis. This paper describes a way of performing urinalysis for UTI detection using the Principal Component Analysis (PCA) implemented using a Field Programmable Gate Array (FPGA). Input to the system is from five ion-selective sensors that measure five different components specifically sodium, nitrite, nitrate, potassium, and pH level of a urine sample. Tests show that the system obtained an accuracy of 94.13% for standard urinalysis showing the accuracy of sensors and measurements used. To be able to detect the presence of UTI in urines, an outlier detection method Principal Component Analysis (PCA), was used. PCA is a tool used in reducing multidimensional data to lesser dimensions while keeping all the information. An accuracy of 83.33% in detecting UTI infection was achieved. The accuracy of FPGA implementation of PCA was compared with MATLAB calculation results and an accuracy of 99.917% was achieved.**

***Index Terms*—DE0-Nano Development Board; PCA-FPGA; Urinalysis; UTI Detection.**

I. INTRODUCTION

Urinalysis is a common test performed in a laboratory to diagnose a Urinary Tract Infection (UTI) [1]. A urine sample is usually evaluated through macroscopic, dipstick, and microscopic analysis. In macroscopic analysis, the sample's color, clarity, and cloudiness are visually observed. Urine dipstick test uses a narrow plastic strip that contains small squares representing a component of the urine to be tested. The entire strip is dipped into the urine samples for a certain period of time and the color changes in each square, which may indicate, urine abnormalities, are noted. The dipstick test is an easy to interpret test. However, the result may be inaccurate because of the test is time-sensitive. Moreover, it only provides a qualitative result. Meaning, it does not provide precise measurements of that certain component being measured. Lastly, the microscopic analysis involves the use of a microscope that is responsible for examining the contents of the urine such as white blood cells and bacteria that are significant in diagnosing UTI.

In this paper, a new way of performing urinalysis and detecting UTI is developed. With the use of five different sensors that measure five different components of a sample, namely sodium, nitrite, nitrate, potassium, and pH level, urinalysis is performed. The sodium and potassium concentrations, as well as the pH level of the urine under test are recorded. In addition, the presence of nitrite and nitrate or if the urine sample is either positive or negative of nitrite and nitrate contents is noted.

To be able to detect UTI, Principal Component Analysis (PCA), which is an outlier detection method, was used. This method has been shown to detect fetal heartbeats [2]. The whole process of PCA was implemented on an EP4CE22F17C6 FPGA board using a combination of software running in the embedded softcore processor and in the FPGA fabric. Offline training phase and online phase are needed for the outlier detection. In the training phase, training data set is stored and it is when their principal components are calculated. In this study's case, this training data set contains only data coming from urine samples that are positive in UTI. On the other hand, online phase refers to when a random urine sample is desired to be tested if it is either UTI positive or negative by comparing its data to the offline training data. The outlier detection accuracy depends on the selection of the components to be measured. This is demonstrated in [3] and [4] where the selection of a principal component as axis where other parameters are to be projected can indicate network intrusion. Good selection of components to be measured will improve the accuracy of the detection.

Implementation of the PCA algorithm through hardware was demonstrated in [5] where an FPGA-based embedded system using the Xilinx ML605 FPGA Development Platform was used. Similarly, [6] used the Altium Nano Board to implement the PCA algorithm for face recognition. The eigenfaces of the face images were computed and stored in the database. Although only two faces were used in the database because of the hardware limitation in the development platform used, the authors demonstrated the functionality of the system. Problem due to hardware limitation in implementing PCA algorithms in embedded systems was addressed by [7] where a two dimensional PCA was used for face recognition. By using low precision in the representation of image feature vectors and network weights, the authors were able to implement a stochastic optimization method implemented in Xilinx Artix-7 XCA100T FPGA that is not constrained by image size.

II. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis is a statistical procedure used in analysis of data. When a very large quantity of observed data is gathered and these data do not have a specific value or a specific pattern, the PCA will make the dimensions of the data smaller until principal components are determined. This has been shown in [8] where PCA is used to reduce the dimensionality of hyper spectral data. In this work, Principal Component Analysis is used to reduce the dimension of the data sets, composed of five parameters namely, Sodium, Nitrite, Nitrate, Potassium and pH. The principal components

obtained are the basis in detecting Urinary Tract Infection. Figure 1 shows the block diagram of the system. The FPGA board is the DE0-Nano Development and Education Board [9], [10] and [11]. It contains the Cyclone IV FPGA hardwired to general purpose IO and 8-channel, 12-bit ADC. The AD8221 instrumentation amplifier [12] provides precision buffering to the ADC input and sensors. Output is displayed on the 2-line LCD display.

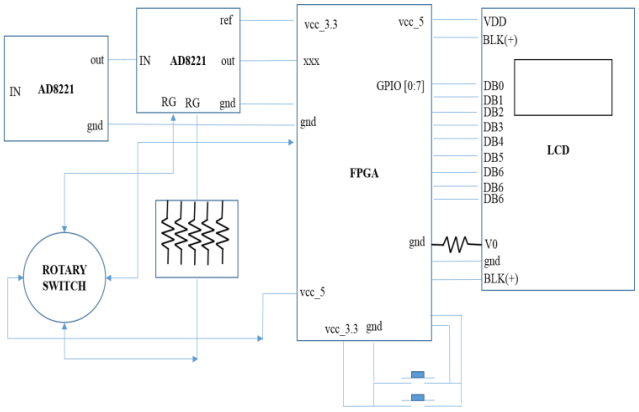

Figure 1: Block diagram of the FPGA-based urine analyzer.

A. *PCA Process in UTI Detection*

In performing PCA, the data set should first be determined. Performing principal component analysis involves the use of combining variance, covariance, covariance matrix, eigenvalues and eigenvectors. As shown in Figure 2, the first step is to get sensor readings from urine samples. From the sensor readings, the covariance matrix is formed, and the dominant eigenvalue and corresponding eigenvectors are extracted. Matrix multiplication is then carried out between the adjusted data and the eigenvectors. The result is a vector representing the principal component. The principal components are classified as UTI positive or UTI negative.

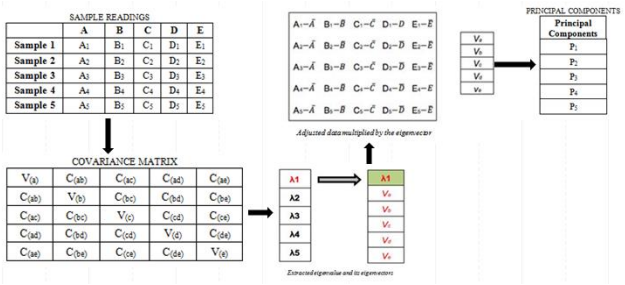

Figure 2: Process of determining the principal component of sensor data.

B. *PCA Framework and Outlier Detection*

The calculation of the principal component is carried out during what is referred to as the training or learning phase. It is an offline phase where the principal component of the data acquired through sensor readings is calculated. For this work, 10 training samples were used and there are 5 ion selective electrodes as sensors for nitrite, nitrate, sodium and potassium. pH electrode is also included as part of the sensors. Sensor readings are converted to 12-bit digital representation. Once the acquisition of all sensor data is complete, a 10 x 5 matrix is formed. This is then used to generate the covariance matrix. This process is illustrated in Figure 3.

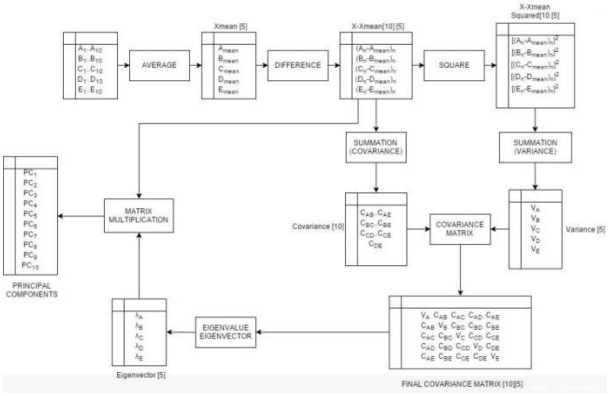

Figure 3: Offline phase. Training or learning mode

Once the principal component of the training set has been obtained from the offline operation, detection of UTI from an unknown urine sample can be carried out. This operation is referred to as the online phase. Here, sensors readings are made on the unknown urine sample. The eigenvector is calculated and this is used to classify whether it belongs to the principal components that are UTI positive or UTI negative. Meanwhile in the online state, the sensor readings of one sample and obtained eigenvectors from the offline state are used to obtain the element PC11. This element is used to classify whether the unknown sample coincides the values of the principal components and returns a value of either positive/negative. Figure 4 illustrates this process.

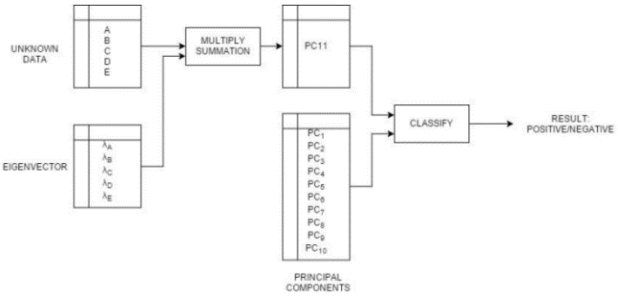

Figure 4: Online phase. UTI detection mode

III. HARDWARE DESIGN

Implementation of the PCA algorithm is facilitated with the use of Altera IP cores [10] that are pipelined to solve various operations in solving for the covariance matrix. The operations are executed synchronously after collecting the training samples. Due to limitation in logic elements and the size of matrices in solving for the principal components, software implementation is done to solve for the most dominant eigenvalue and its corresponding eigenvectors using power iteration algorithm. The disadvantage of implementing designs through software is that it consumes more power and also executes operations at a slower rate compared to hardware implemented designs especially when dealing with matrix operations [6].

A. *Signal Conditioning Circuit*

The signal conditioning circuit composed of the instrumentation amplifier, resistor bank to vary the gain and filter circuit to suppress the 60Hz line noise, is used to condition the sensor signal before applying it to the ADC. A selector switch allows selecting the right resistor combination

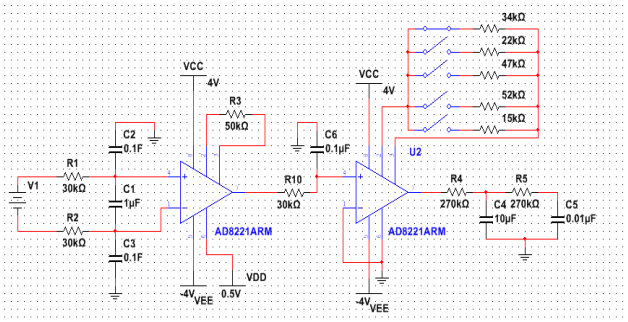

Figure 5: Signal conditioning circuit

to provide the right gain. Figure 5 shows the signal conditioning circuit.

B. *Computation Pipeline*

For computing the mean, variance and the covariance, the functions provided in the Qsys library were used. The resulting RTL schematic is shown in Figure 6.

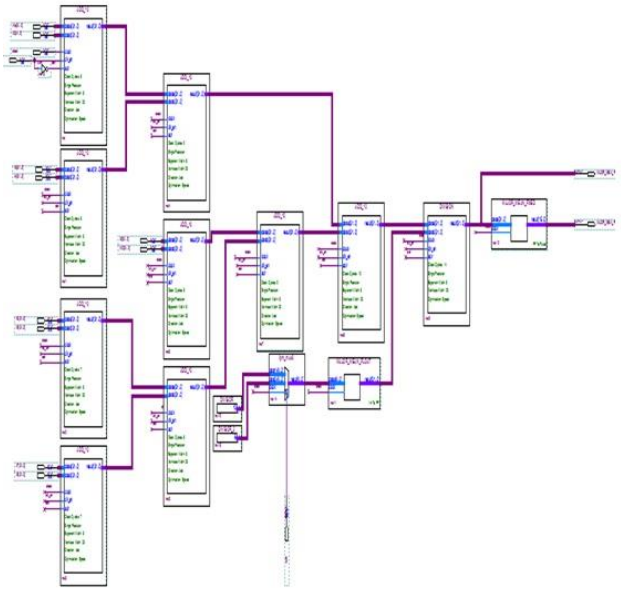

Figure 6: Circuit to compute for the mean, variance and co-variance

C. *NIOS Processor*

A C program that runs on the NIOS processor [5] is used to calculate the eigenvalue and eigenvector.

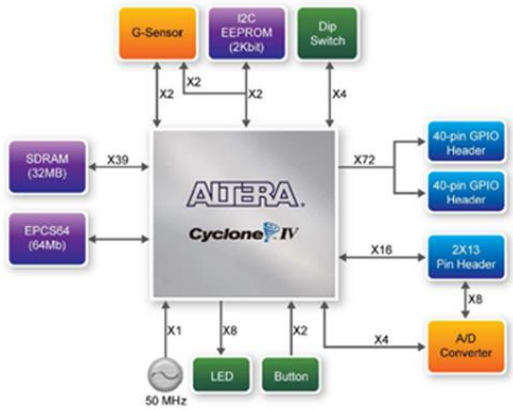

Figure 7: The FPGA platform that implements an embedded system containing a NIOS processor.

IV. Software Design

The computation of the 5x5 covariance matrix is carried out as a C program running in the NIOS processor. This was done because of the hardware limitations in implementing the desired operation purely on FPGA fabric.

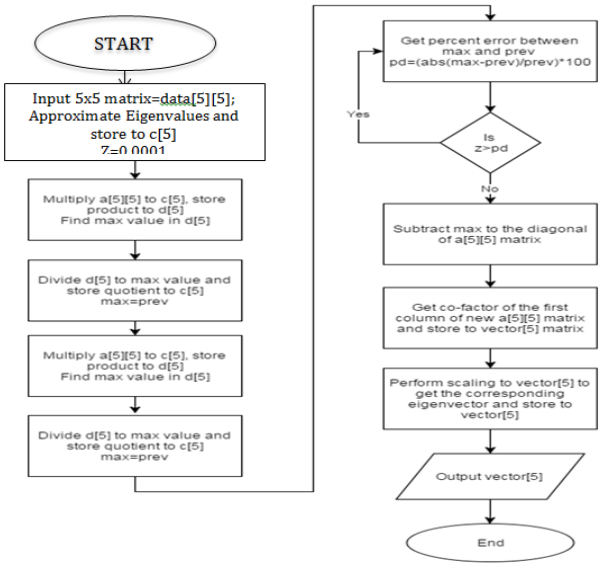

Figure 8: System flowchart for the C program that computes for the covariance matrix

V. Data and Results

A. *Matlab Comparison*

To verify the accuracy of the computation of the principal component done by the system, computations are performed using Matlab and then compared with computations carried out in FPGA. Table 1 shows the principal component calculation of Matlab and FPGA. The highest percent error of the principal component is 0.198 % and the average error of the calculation is 0.083%.

Table 1
Urinalysis Results Comparison Between Computation Done in Matlab and Computation Done Using The Proposed System

Sample	Matlab	FPGA	Percentage Error (%)
1	-0.1644	-0.1646	0.148
2	0.5823	0.5823	0.000
3	0.3718	0.3716	0.020
4	-05636	-0.5636	0.000
5	-0.5017	-05016	0.020
6	-0.2152	-0.2153	0.046
7	0.2511	0.2508	0.040
8	0.4929	0.4935	0.122
9	-0.1522	-01919	0.197
10	-0.1011	-0.1013	0.198
Average Percentage Error			0.083

B. *Urinalysis Comparison*

Urinalysis results obtained calculated by the system is compared with clinical results. Samples were taken from the subjects on the same day and subjected to clinical urinalysis using the facilities of a clinical laboratory. Urine sample taken from the same sample sent to the clinical laboratory was used where measurements of parameters were made and processed using to the proposed system. The results

summarized in Table 2 indicated that the proposed system can measure the presence of Sodium, Nitrate, Potassium and pH with a percentage average error of 5.87 %.

Table 2
Urinalysis Results Comparison Between Computation Done in Matlab and Computation Done Using The Proposed System

Sample	Parameter	Laboratory Results	Proposed System	Percent Error (%)
1	Sodium	4783.73	5375	12.40
	Nitrite	Negative	Negative	
	Potassium	1833.32	1730.48	5.61
	pH	6.00	6.28	4.67
2	Sodium	4253.09	0.5823	10.14
	Nitrite	Negative	Negative	
	Potassium	1630.40	1481.78	9.12
	pH	6.00	5.89	1.83
3	Sodium	0.3718	0.3716	1.39
	Nitrite	Negative	Negative	
	Potassium	758.50	722.99	4.68
	pH	6.00	6.38	6.33
4	Sodium	-05636	-0.5636	1.74
	Nitrite	Negative	Negative	
	Potassium	340.15	377.02	10.84
	pH	6.00	6.10	1.67
	Average Percentage Error			5.87

C. UTI Detection

UTI detection was carried out using 30 test samples and applied to 10, 15 and 20 training samples. The result is listed in Table 3. The highest accuracy achieved is 56.67 % with 20 training samples used to calculate the Principal Component.

Table 3
Accuracy of Detection with 10, 15 And 20 Samples with 5 Sensors

Training Samples	Accuracy (%)
10	53.33
15	50
20	56.67

It was observed that removing the pH sensor then calculating the eigenvalues generated the results listed in Table 4. The procedure in detecting UTI is again performed without the pH sensor yield the result listed in Table 5.

Table 4
Computed Eigenvalues with 5 Sensors and 4 Sensors (pH Sensor Removed)

Eigenvalues with pH sensor	Eigenvalues without pH sensor
0.2628	0.0720
0.0661	0.0485
0.0476	0.0159
0.0136	0.0123
0.0114	

As listed in Table 5, when 10 trainer samples were used, 18 out of 30 test samples matched the expected results. With 15 training samples, 22 out of 30 test samples matched, and 25 out of 30 samples matched for 20 training samples.

Table 5
Accuracy of Detection for 10, 15 and 20 Samples with pH Sensor Removed

Training Samples	Accuracy (%)
10	60
15	73.33
20	83.33

VI. CONCLUSION

This work is an attempt to create an electronic-based urinalysis and to use the Principal Component Analysis to detect UTI. The FPGA-based Urinalysis for UTI Detection Using Principal Component Analysis was successfully developed using a 5-parameter test urinalysis and implemented the PCA algorithm in a DE0-NANO board FPGA. Urine readings were taken by the five sensors namely, Sodium, Nitrite, Nitrate, Potassium and pH. The results of the urinalysis and the UTI detection are displayed on the LCD. Results displayed in the LCD are the concentration levels of Potassium, Sodium, Nitrate, Nitrite and pH level, also if it is nitrate, nitrite positive or negative. The system successfully computed for the principal components and compared to results obtained using Matlab. The percent error of 0.083 percent is indicative of high degree of precision achieved by the FPGA-based computation.

When the system is used to perform urinalysis on test urine samples, an average accuracy of 94.13% was achieved. When used to detect UTI, the system performed poorly achieving only 56.67 % with 5 sensors being used. Inspection of the resulting eigenvalues with 5 and four sensors, indicate 4 eigenvalues which are not distant in magnitude relative to each other while one eigenvalue with a relatively high value compared to the other four. With eigenvalues ranging from 0.0114 to 0.0661, the eigenvalues with a value of 0.2628 corresponding to the pH sensor is discarded. Re-computing the principal component and running the UTI detection test yields the detection accuracy of 83.33%. This lends well to the PCA methodology of analyzing data where selection of the principal components through the eigenvalues provides better results.

Increasing the number of samples used in training indicates that higher accuracy can be achieved. When the training samples were increased from 15 to 20 trainer samples the success rates increased from 73.33% to 83.33%. Future works include increasing the number of training samples and to use larger capacity FPGA Development boards to fully implement the algorithms on the FPGA fabric under the control of the softcore processor.

REFERENCES

[1] N. Siamak and N. Nabili, "Urinalysis: Get facts on this common examination, MedicineNet. Retrieved 25 March 2016, from http://www.medicinenet.com/urinalysis/article.htm.

[2] R. Petrolis, "Multi stage principal component analysis based method for detection of fetal heart beats," Computing in Cardiology 2013, pp-301-304.

[3] M. Elrawy, "Ids In Telecommunication Network Using PCA," International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.4, July 2013, pp. 147 – 157.

[4] A. Das, S. Misra, S. Joshi, J. Zambreno, G. Memik, and A. Choudhary, "An efficient FPGA implementation of principle component analysis based network intrusion detection system," Proceedings of the conference on Design, automation and test in Europe, 2008, pp. 1160-1165.

[5] D. G. Perera and K.F. Li, "Embedded Hardware Solution for Principal Component Analysis," Proceedings of 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, IEEE Conference Publications, pp. 730 - 735.

[6] R.Endluri, M.Kathait, and K. C. Ray, "Face recognition using PCA on FPGA based embedded platform," 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE), IEEE Conference Publications 2013.

[7] S. K. Dwivedi, S. P. Azad, and P. Ellervee, "Hardware implementation of face recognition using low precision representation," Electronics

Conference (BEC) 2016 15th Biennial Baltic, November, 2016, pp. 63 – 66.

[8] D. Fernandez, C. Gonzalez, D. Mozos, and Sebastian Lopez, "FPGA implementation of the principal component analysis algorithm for dimensionality reduction of hyperspectral images," Real-Time Image Proc (2016). doi:10.1007/s11554-016-0650-7.

[9] Altera.com, (2016). Nios II Processor - Design Tools. Retrieved 28 February 2016, from https://www.altera.com/products/processors/design-tools.html.

[10] Introduction to the Quartus II Software. (2016) (1st ed.). Retrieved from https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/manual/intro_to_quartus2.pdf.

[11] Nios II Gen2 Software Developer's Handbook. (2016) (1st ed.). Retrieved from https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/nios2/n2sw_nii5v2gen2.pdf.

[12] AD8221. (2016). Retrieved from http://www.analog.com/media/en/technical-documentation/data-sheets/AD8221.pdf.