BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI

Volumul 62 (66)

Numărul 1-2

AUTOMATICĂ şi CALCULATOARE

2016 **Editura POLITEHNIUM**

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
PUBLISHED BY
"GHEORGHE ASACHI" TECHNICAL UNIVERSITY OF IAŞI
Editorial Office: Bd. D. Mangeron 63, 700050, Iaşi, ROMANIA
Tel. 40-232-278683; Fax: 40-232-237666; e-mail: polytech@mail.tuiasi.ro

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
BULLETIN OF THE POLYTECHNIC INSTITUTE OF IAŞI
Volumul 62 (66), Numărul 1-2 2016

AUTOMATICĂ şi CALCULATOARE

SUMAR

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
BULLETIN OF THE POLYTECHNIC INSTITUTE OF IAŞI
Volume 62 (66), Number 1-2 2016

AUTOMATIC CONTROL and COMPUTER ENGINEERING

CONTENTS

Pp.

.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 1-2, 2016
Secţia
AUTOMATICĂ şi CALCULATOARE

A COMPARATIVE STUDY ON THE METHODS OF AUTOMATIC FABRIC INSPECTION

BY

DOINA CAŞCAVAL[1,*] and PETRU CAŞCAVAL[2]

“Gheorghe Asachi” Technical University of Iaşi, Romania,
[1]Faculty of Textiles, Leather and Industrial Management
[2]Faculty of Automatic Control and Computer Engineering

Received: January 5, 2016
Accepted for publication: February 8, 2016

Abstract. The aim of this work is to identify the most efficient algorithms for fabric defect detection, suitable for an on-line inspection process. The algorithms for the fabric inspection are based on the image processing techniques. More exactly, an image captured from the fabric surface (testing sample) is compared with the image from a fabric without defect (witness sample). Three different methods for fabric inspection have been considered in this overview: statistical methods, model based methods inspired from the formal recognition techniques, and spectral analysis based methods. This paper highlights that the efficiency of fabric inspection algorithms is strongly depended to the type of the defects we consider. For example, the algorithms that use spectral analysis techniques are very efficient in case of global defects, but are not so suitable in case of local ones. The most difficult problem of the fabric inspection is given by the fact that the number of fabric defects is very large. Consequently, to cover a large range of fabric defects, complex algorithms must be used. On the other hand, to increase the inspection speed, the algorithm must be as simple as possible. This is the challenge for the on-line fabric inspection process. Another conclusion of this analyse is that the system for automatic fabric inspection attached to a weaving machine or to a control ramp must have a high flexibility. Moreover, any algorithm for automatic inspection must allow

[*] Corresponding author; *e-mail*: cascaval@tex.tuiasi.ro

adjusting some work parameters, such as the window width, the overlap degree of the windows, the threshold values etc., depending on the two main conflicting requirements: the inspection speed and the accuracy of the defects detection. Another limitation for a fabric inspection derives from an essential feature of the classification methods. The problem is that once the defects detection rate increases, the rate of the false alarms also increases affecting the weaving process on the whole. In other words, the quality of the on-line fabric inspection cannot be increased however much because of the false alarms. The production engineer must keep this balance between these two contradictory trends, on the condition that the automatic inspection system ensures a high flexibility in this sense.

Keywords: woven fabric defects; automatic fabric inspection; image processing; spectral analysis.

2010 Mathematics Subject Classification: 68U10.

1. Introduction

The increasing quality requirements for manufactured products led to the development and continuous improvement of inspection and control. Textile materials are representative products with high quality requirements and which require high performance systems for automatic inspection and control.

Typically, the fabrics have the width between 1 and 3 meters, and are unrolled with a speed between 20 and 200 m/min. At a direct visual inspection, when the fabric is moving with more than 30 m/min, and its width it is bigger than 2 m, more than 60-70% of the defects can remain undetected. Moreover, the effectiveness of the visual inspection decreases with fatigue. As in other verification processes, the result depends largely on the experience and qualification of the worker. Note that an undetected defect could reduce the product price up to 45-60%. This is the reason why the development of integrated systems for automatic inspection is a major objective in textile quality control.

Most automatic inspection systems for textile products operate off-line and have a coating speed up to 100 m/min. In the last years, on-line automatic control devices installed on the loom have occurred. They consist of a set of video cameras and dedicated software able to detect fabric defects in real-time. In contrast to the human operator, the automatic system inspects the entire surface of the fabric, uninterruptedly and without variations due to fatigue. Nevertheless, such systems can detect only a limited set of defects. For such a system, we must take into account two conflicting requirements: (a) to detect a large range of fabric defects, and (b) to inspect the fabric with a high speed (*i.e.*, in real-time). In the last 20 years, many attempts to fulfil these requirements have been reported. The proposed approaches have been characterized into three categories: statistical, spectral and model-based. In the next section, a brief comparative analysis of these methods is presented. Note that, many authors suggest that the combination of statistical, spectral and model-based approaches can give better results than single any single approach. An excellent survey on fabric defect detection is given by Kumar (2008).

2. Fabric Inspection Techniques. A Comparative Analysis

For the fabric inspection, the image processing techniques are applied. More exactly, an image captured from the fabric surface called testing sample is compared with the image of a fabric sample without defect. The most difficult problem of the fabric inspection is given by the fact that the number of fabric defects is very large. Consequently, to cover a large range of fabric defects, complex algorithms must be used. On the other hand, to increase the speed of automatic inspection, the algorithm used must be as simple as possible. This is the challenge for the on-line fabric inspection process.

After a first classification, the fabric defects can be local or global. A local defect affects only a small part of the fabric area, whereas a global one causes a large distortion of the fabric structure. These two kinds of fabric defects require distinct approaches. For example, the global defects are detected easily by using a Fourier analysis of the image captured from the fabric surface. But, this technique of image processing is not at all recommended in case of a local defect.

As mentioned above, three different methods for fabric inspection have been reported in literature: statistical methods, model-based methods inspired from the formal recognition techniques, and spectral analysis based methods.

In a statistical approach, a detection algorithm is based on the sliding window technique in which a window is moved over the whole image area. In other words, the fabric image is divided into frames which are then compared each other. The presence of a defect over texture causes regular structure changes and consequently, statistical changes. Many techniques can be applied: comparing the average and the standard deviation; checking the correlation index; computing the Karhunen-Loeve (KL) transform; feature extraction on levels of gray etc. For example, the results presented by Tunák and Linka (2006) show that the statistical approaches are suitable for detection of directional defects or changes in regular structure of real fabrics.

The methods inspired from the formal recognition are based especially on the Markov models. For example, Cohen and his collaborators (1991) propose a Markov model of type “Random Field” (MRF). A parallel implementation of this model for a real-time fabric inspection is reported by Atalay (1995). Note that, the Markov models of the first order cover a reduce range of fabric defects. For a better detection of the fabric defects, the Markov models of second order or even of third order must be used. But, in these cases, the computing time increases exponentially. Another method applied to detect the defects in textured images is given by Chen and Jain (1988).

But, most methods for fabric inspection are based on the spectral analysis of the digital image captured from the fabric surface. A short enumeration of these methods is presented as follows.

Chan and Pang (2000) propose a method for global defect detection based on the Fourier analysis. As the Fourier transform has certain limitations which derive from the interval used for integration, this method is not recommended for local or isolated defects. To overcome this shortcoming, Kumar and Pang (2002a) propose another approach for fabric defect detection based on Gabor filters. But this method needs a large computation time. For this reason, they also propose a reduced method, much faster, which takes into account only the imaginary part of these filters. Later, Kumar and Pang (2002b) improve this method of spectral analysis by using some additional filters. Note that, most automatic inspection algorithms have in view other fabric without models, canvas fabric type. For the patterned fabric, more less algorithms have been reported. Of these, the algorithm given by Ngan and his collaborators (2003) must be mentioned. To cover the defects in patterned fabric they propose a method based on the wavelet transform.

Other techniques of image processing to improve the fabric inspection must be mentioned here. Thus, Kang and his collaborators (1999; 2001) propose an illumination technique of the fabric surface in order to identity the pattern. On the other hand, Tsai and Hu (1996) uses the Fourier tranform of some fabric images as entries into a neural network to improve the fabric inspection. He trained an artificial neural network to identify four types of fabric defects: lack of filling yarn, lack of warp yarn, oil stains and fabric break.

In other work reported by Hu and Tsai (2000), the authors use the wavelet facilities in connection with an artificial neural network. Other results related to the fabric analysis obtained based on wavelet transform is presented by Jasper (1995; 1996). Also, Escofet *et al*. (1998) use Gabor filters for fabric image segmentation for a large variety of fabric patterns.

Another algorithm for texture defect detection in uniform and grid-like structured fabrics is proposed by Bissi *et al*. (2013). This approach comprises a feature extraction phase, which relies on a symmetric Gabor filter bank, and on a defect identification phase, that is based on the Euclidean norm of features and on the comparison with fabric type specific parameters.

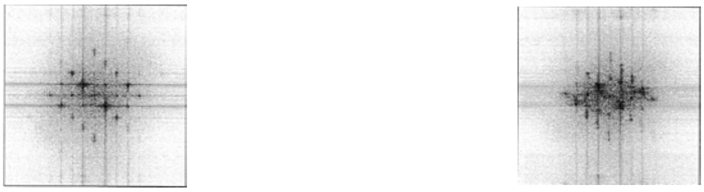
An important algorithm that addresses defect detection in textured surface by using an elliptical Gabor filter (EGF) is proposed by Hu (2015). In the inspection process, each sample under inspection is convoluted with a selected EGF, followed by a gray level thresholding process to generate a binary segmented result. Thus, a single filter is utilized during the inspection process and, consequently, the computational time required for each defective sample is quite limited, suitable for an on-line fabric inspection.

The efficiency of the techniques that use the spectral analysis in the process of fabric inspection is illustrated by an example related to a global defect. Fig. 1 presents two fabric samples (captured images): one from a fabric without defect, and another one from a fabric affected by a global defect.

(a) without defect (b) with a global defect

Fig. 1 – Fabric samples.

Escofet *et al.* (2001) have studied the point distribution in the power spectrum of the fabric image in order to identify the fabric pattern. The 2D Fourier transform has been evaluated by using a FFT algorithm and the result is presented in a logarithmic scale in Fig. 2.



(a) sample without defect (b) sample with global defect

Fig. 2 – Comparison regarding the power spectrum.

Fig. 2 highlights significant differences of power spectrum for the two fabric samples that means that the spectral analysis ensures a good dissociation in that case.

Other results that demonstrate the effectiveness of the spectral analysis techniques in detection of global fabric defects are presented by Millan and Escofet (1996), Ciamberlini *et al*. (1996), Zhang *et al*. (2011) etc. But, these techniques are not so suitable in case of local defects.

To address the local defects, Tsai and Hu (1996) have compared the larger values of the power spectrum in order to distinguish between the defective and the defect-free regions. They have trained an artificial neural network for a good classification of the samples. However, the method proposed by Tsai and Hu is too particular, applicable only to the fabric without model, canvas fabric type. This method needs a training phase for each type of fabric defect.

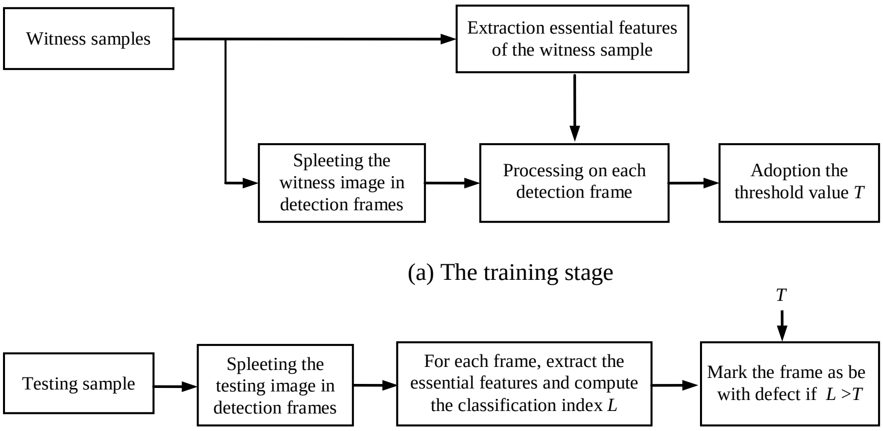
Another algorithm that addresses the local defects is presented by Çelik *et al*. (2014). The defect detection algorithm is based on wavelet transform, double thresholding binarization, and morphological operations. The defect classification approach relies on gray level co-occurrence matrix and feed forward neural network.

The problem of the fabric inspection becomes more complex if we try to locate a defect, not only to detect it. Note that, a Fourier analysis does not give sufficient information in order to locate a fabric defect. More suitable are the methods that use wavelets because these methods allow extracting the information both from the space and the frequency field. For example, Li *et al.* (2015) propose a complex fabric defect detection and location algorithm based on multi-scale wavelet transform. More exactly, the sample image is firstly tacked by a "pyramid" wavelet decomposition algorithm. Then, the obtained new images are segmented by an Expectation-Maximization (EM) algorithm.

All the considerations presented in this section demonstrate the effectiveness of the spectral analysis methods in the field of fabric inspection. Nevertheless, these methods need a large computation time.

3. Considerations Regarding the Fabric Inspection Process

Regardless the method used for defect detection, the fabric inspection process comprises two phases: a learning phase in which some samples of the fabric without defect are processed, and a checking phase in which the testing sample (an image captured from the fabric surface) is processed and compared with the samples without defect. The two stages of the fabric inspection process are illustrated in Fig. 3.


Witness samples
Extraction essential features of the witness sample
Spleeting the witness image in detection frames
Processing on each detection frame
Adoption the threshold value *T*
T
Testing sample
Spleeting the testing image in detection frames
For each frame, extract the essential features and compute the classification index *L*
Mark the frame as be with defect if *L* >*T*

(a) The training stage

(b) The testing stage

Fig. 3 – The stages of the defect detection process.

The method used for an automatic fabric inspection must be adopted taking into account many criteria, namely: if we want to cover a large variety of fabric defects or we want a higher speed of fabric inspection; the inspection process is carried out on-line or off-line; the fabric is canvas type or has a model; the model of the fabric has a regular structure or is a flower type etc. As a conclusion, the techniques for fabric inspection are very different; each of them has some advantages and disadvantages.

The system for automatic fabric inspection attached to a weaving machine or to a control ramp must be as flexible as possible. Such a system must allow to choice a suitable algorithm for inspection, from a large library of dedicated programs. Moreover, any algorithm for automatic inspection must allow adjusting some work parameters, such as the window width, the overlap degree of the windows, the threshold values etc., depending on the two main conflicting requirements: the inspection speed and the accuracy of the defects detection.

Another limitation for a fabric inspection derives from an essential feature of the classification methods. Such a method uses a classification index of which value is compared with a threshold value. Depending to this threshold value, the defect detection rate is bigger or smaller. The problem is that once the defect detection rate increases, the rate of the false alarms also increases, affecting the weaving process on the whole. In other words, the quality of the on-line fabric inspection cannot be increased however much, because the efficiency of the weaving process may decreases significantly as a result of the false alarms. The production engineer must keep this balance between these two contradictory trends, on the condition that the automatic inspection system ensures a high flexibility in this sense.

4. Conclusion

The techniques used for fabric inspection are very different and each of them has some advantages and disadvantages. As a rule, the most effectiveness algorithms with a higher defect detection rate are complex ones that need a large computation time. For this reason, the implementation in the on-line fabric inspection systems is quite difficult. As a compromise between the accuracy of the defect detection and the response time, we recommend the methods that combine the spectral analysis techniques with statistical ones. As regards the fabric with complex models, the current methods do not yield satisfactory results. Even though the Gabor filter based methods reach a high defect detection rate for canvas type fabric, in case a flower type fabric the results are not yet conclusive. Better results are expected from the methods that combine the spectral analysis techniques with artificial intelligent models.

Acknowledgments. Some of these results were achieved under the project PN II 71-142.

REFERENCES

Atalay A., *Automated Defect Inspection of Textile Fabrics Using Machine Vision Techniques*, M.S. Thesis, Bogazici University, Istanbul, Turkey, 1995.

Bissi L., Baruffa G., Placidi P., Ricci E., Scorzoni A., Valigi P., *Automated Defect Detection in Uniform and Structured Fabrics Using Gabor Filters and PCA*, Journal of Visual Communication and Image Representation, **24**, *7*, 838-845, 2013.

Ciamberlini C, Francini F., Longobardi G., Sanson P., Tiribilli B., *Defect Detection in Textured Materials by Optical Filtering with Structured Detectors and Self-Adaptable Masks*, Optical Engineering, **35**, *3*, 838-844, 1996.

Chan C.-H., Pang G.K.H., *Fabric Defect Detection by Fourier Analysis*, IEEE Transactions on Industry Applications, **36**, *5*, 1267-1276, 2000.

Chen J., Jain A.K., *A Structural Approach to Identify Defects in Textured Images*, Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC ’88), **1**, 29-32, Beijing, China, 1988.

Cohen F.S., Fan Z., Attali S., *Automated Inspection of Textile Fabrics Using Textural Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **13**, *8*, 803-808, 1991.

Çelik H.İ., Dülger L.C., Topalbekiroğlu M., *Development of a Machine Vision System: Real-Time Fabric Defect Detection and Classification with Neural Networks*, The Journal of The Textile Institute, **105**, *6*, 575-585, 2014.

Escofet X., Navarro R., Millan M.S., Pladeiiorens J., *Detection of Local Defects in Textile Webs Using Gabor Filters*, Optical Engineering, **37**, *8*, 2297-2307, 1998.

Escofet X., Millan M.S., Rallo M., *Modelling of Woven Fabric Structures Based on Fourier Image Analysis*, Applied Optics, **40**, *34*, 6170-6176, 2001.

Hu M.C., Tsai IS., *Fabric Inspection Based on best Wavelet Packet Bases*, Textile Research Journal, **70**, *8*, 662-670, 2000.

Hu G.H., *Automated Defect Detection in Textured Surfaces Using Optimal Elliptical Gabor Filters*, Optik - International Journal for Light and Electron Optics, **126**, *14*, 1331-134, 2015.

Jasper W.J., Potlapalli H., *Image Analysis of Mispicks in Woven Fabric*, Textile Research Journal, **65**, *1*, 683-692, 1995.

Jasper W.J., Gamier S.J., Potlapalli H., *Texture Characterization and Defect Detection Using Adaptive Wavelets*, Optical Engineering, **35**, *11*, 3140-3149, 1996.

Kang T.J. *et al.*, *Automatic Recongnition of Fabric Weave Patterns by Digital Image Analysis*, Textile Research Journal, **69**, *2*, 77-83, 1999.

Kang T.J. *et al.*, *Automatic Structure Analysis and Objective Evaluation of Woven Fabric Using Image Analysis*, Textile Research Journal, **71**, *3*, 261-270, 2001.

Kumar A., Pang G.K.H., *Defect Detection in Textured Materials Using Gabor Filters*, IEEE Transactions on Industry Applications, **38**, *2*, 425-440, 2002a.

Kumar A., Pang G.K.H., *Defect Detection in Textured Materials Using Optimized Filters*, IEEE Transactions on Systems, Man and Cybernetics, Part B, **32**, *5*, 553-570, 2002b.

Kumar A., *Computer-Vision-Based Fabric Defect Detection: a Survey*, IEEE Transactions on Industrial Electronics, **55**, *1*, 348-363, 2008.

Li P., Zhang H., Jing J., Li R., Zhao J., *Fabric Defect Detection Based on Multi-Scale Wavelet Transform and Gaussian Mixture Model Method*, The Journal of the Textile Institute, **106**, *6*, 587-592, 2015.

Millán M.S., Escofet J., *Fourier Domain Based Angular Correlation for Quasiperiodic Pattern Recognition, Applications to Web Inspection*. Applied Optics, **35**, *31*, 6253-6260, 1996.

Ngan H.Y.T., Pang G.K.H., Yung S.P., Ng M.K., *Defect Detection on Patterned Jacquard Fabric*, Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop (AIPR ’03), 163-168, Washington, DC, USA, 2003.

Tsai L.S., Hu M.C., *Automatic Inspection of Fabric Defects Using an Artificial Neural Network Technique*, Textile Research Journal, **66**, *7*, 474-482, 1996.

Tunák M., Linka A., *Simulation and Recognition of Common Fabric*, Proceedings of 13th International Conference on Structure and Structural Mechanics of Textiles, 363-370, Liberec, Czech Republic, 2006.

Zhang X., Pan R., Liu J., Gao W., Xu W., *Design Gabor Filters in the Frequency Domain for Unsupervised Fabric Defect Detection*, Industria Textila, **4**, 177-182, 2011.

STUDIU COMPARATIV PRIVIND METODELE DE VERIFICARE AUTOMATĂ A ŢESĂTURII

(Rezumat)

Scopul lucrării este identificarea celor mai eficienţi algoritmi de detectare a defectelor în ţesătură la o verificare on-line. Aceşti algoritmi se bazează pe tehnici de prelucrare a imaginilor captate de la ţesătura verificată şi compararea lor cu imagini ale unor mostre de ţesătură fără defect. Trei tipuri de metode sunt aplicate în prelucrarea acestor imagini: metode statistice, metode bazate pe modele inspirate din tehnicile de recunoaştere a formelor şi metode bazate pe analiza spectrală. Lucrarea evidenţiază faptul că eficienţa algoritmilor de verificare a ţesăturii depinde în mare măsură de tipurile de defecte urmărite. De exemplu, metodele de analiză spectrală sunt foarte eficiente în detectarea defectelor globale, dar sunt mai puţin potrivite în cazul defectelor locale, izolate. Dificultatea problemei de control automat derivă din faptul că numărul defectelor ce pot apărea în procesul de ţesere este foarte mare. Prin urmare, în problema inspectării automate a ţesăturii se impun două cerinţe contradictorii. Pe de o parte, pentru creşterea calităţii, se impune acoperirea unei game tot mai mari de defecte, ceea ce implică folosirea unor algoritmi sofisticaţi, iar pe de altă parte, pentru creşterea vitezei de inspectare automată, se impun algoritmi cât mai simpli.

O altă limitare a inspectării automate a ţesăturii derivă dintr-o caracteristică esenţială a metodelor de clasificare. Astfel, dacă prin valorile de prag se impune o rată mai ridicată de detectare a defectelor atunci şi rata alarmelor false creşte, ceea ce afectează randamentul maşinilor de ţesut. Cu alte cuvinte, calitatea procesului de inspectare automată nu poate creşte oricât de mult din cauza alarmelor false, care duc la oprirea abuzivă a procesului de ţesere. Inginerul de producţie trebuie să păstreze un echilibru între cele două tendinţe contradictorii.

O altă concluzie a acestui studiu este aceea că sistemul de verificare ataşat maşinii de ţesut sau de la rampa de control trebuie să fie cât mai flexibil. Mai mult, fiecare algoritm trebuie să permită ajustarea unor parametri de lucru, cum ar fi: lăţimea ferestrei de detecţie, gradul de suprapunere a ferestrelor de detecţie vecine, valorile de prag pentru clasificare etc. La stabilirea acestor parametri trebuie să se ţină cont de cele două cerinţe esenţiale, acurateţea controlului automat şi viteza de inspectare a ţesăturii.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 1-2, 2016
Secţia
AUTOMATICĂ şi CALCULATOARE

THE PERFORMANCE EVALUATION OF VARIOUS TECHNIQUES FOR TRANSPORTATION PROBLEM

BY

AMINUR RAHMAN KHAN[1,2,*], ADRIAN VILCU[2],
MD. SHARIF UDDIN[1] and CRISTIANA ISTRATE[2]

[1]Jahangirnagar University, Dhaka-1342, Bangladesh,
Department of Mathematics
[2]“Gheorghe Asachi” Technical University of Iaşi, Romania,
Department of Management Engineering

Received: March 22, 2016
Accepted for publication: April 28, 2016

Abstract. Umpteen approaches are available in the literature to determine the initial basic feasible solution to the transportation problem. They contain the intractability of carrying out enormous calculations. However it is difficult to presume which procedure will give optimum result in the solution of a given problem. Considering this we illustrate techniques using Matlab to reduce the enormous calculations. In addition to that it is possible to perform all procedures on any problem speedily. As a result researcher can select the best solution promptly. We also calculate the average performance of various techniques with respect to the optimal solution.

Keywords: Average Relative Deviation; Transportation Problem; MATLAB; Distribution Indicator; Optimum Solution.

2010 Mathematics Subject Classification: 90B50, 90C08.

[*]Corresponding author; *e-mail*: aminurju@yahoo.com

1. Introduction

The branch of Linear Programming Problem (LPP) in which a single homogeneous product is transported from several sources to numerous destinations subject to the supply and demand of the sources and destinations respectively, such that the total cost of transportation is minimized is Transportation Problem (TP). This problem was first presented by Hitchcock in 1941 and so it is often called in the literature the “Hitchcock problem”. Since then a number of procedures have been developed to solve the transportation problem, notably a simplified version of the Simplex method by Dantzig (1951), the stepping-stone method by Charnes *et al*. (1953) and the modified distribution method by Ferguson (1955).

General understanding is that better is an Initial Basic Feasible Solution (IBFS) lesser is the number of iterations in obtaining the minimal total cost solution to the TP. Motivated by this notion, a number of better methods of obtaining an efficient IBFS to the TP is developed by Abdur Rashid (2011; 2013), Aminur Rahman Khan (2011; 2012; 2015a; 2015b; 2015c), Hamdy (2007), Juman and Hoque (2015), Kasana and Kumar (2005), M. Sharif Uddin *et al*. (2011), Md. Amirul Islam *et al*. (2012a; 2012b), Md. Ashraful Babu *et al*. (2013; 2014a; 2014b), Md. Main Uddin *et al*. (2015), Mollah Mesbahuddin Ahmed *et al*. (2014a; 2014b; 2016), Nizam Uddin Ahmed *et al*. (2015), Sayedul Anam *et al*. (2012) and Utpal Kanti Das *et al*. (2014a; 2014b).

Object oriented programs using C++ had been developed for five methods (North West Corner Method, Least Cost Method, Row Minimum Cost Method, Column Minimum Cost Method and Vogel’s Approximation Method) initially by Taghrid Imam *et al*. (2009) and later by Nabendu Sen *et al*. (2010) to avoid enormous calculations of solving transportation problem. The code of VAM using C++ is corrected by Juman *et al*. (2013).

In this paper, we implement twelve methods in Matlab, a high level computer language and it is utilized via twenty randomly selected problems of different order from some reputed journals in order to prove the exactness of the code. We also calculate the Average Relative Deviation (ARD) to determine the Average Performance (AP) of various techniques with respect to the optimal solution.

2. Formulation of TP

The notations used to formulate the cost minimization transportation problem are as follows:

m – Total number of sources;
n – Total number of destinations;
S_i – Amount of supply from the ith source;

d_j – Amount of demand at *j*th destination;
c_{ij} – Unit transportation cost from *i*th sources to *j*th destination;
x_{ij} – Amount transported from *i*th sources to *j*th destination.

The following network in Fig. 1 represents the general cost minimization transportation problem.

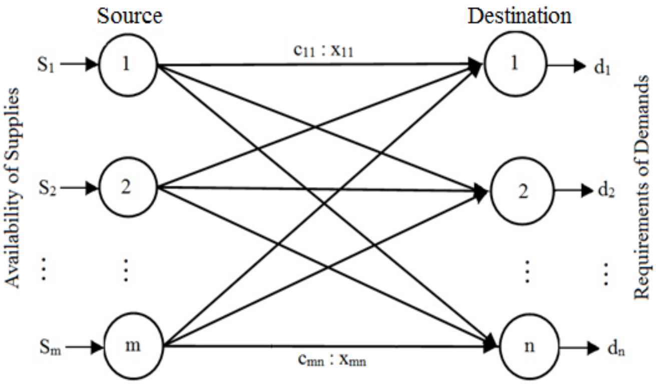


Fig. 1 – Network representation of Cost Minimization Transportation Problem.

Using the above notations, the LPP model of the balanced cost minimization transportation problem is given by

$$\text{Min.} \quad Z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\, x_{ij} \tag{1}$$

$$\text{s/t,} \quad \sum_{j=1}^{n} x_{ij} = S_i\,; \qquad i=1,2,\dots\ \dots,m \tag{1.1}$$

$$\sum_{i=1}^{m} x_{ij} = d_j\,; \qquad j=1,2,\dots\ \dots,n \tag{1.2}$$

$$x_{ij} \geq 0 \ \text{ for all } i, j. \tag{1.3}$$

3. Solution Procedures of TP

There are lots of methods available in the literature to obtain an initial basic feasible solution of balanced cost minimization transportation problem. Well known methods among them are summarized below.

3.1. North West Corner Method (NWCM)

NWCM starts with maximum possible allocation to the North West corner cell (*i.e.* top left corner) of the transportation table. It is then delete the row or column in accordance with supply or demand is satisfied and go to next row or column. The process is continued until all supply and demand values are exhausted.

3.2. Least Cost Method (LCM)

To begin with LCM, maximum possible allocation is made to the lowest cost cell of the transportation table. Ties are broken arbitrarily. It is then delete the row or column in accordance with supply or demand is satisfied and allocate to the next lowest cost cell. The procedure is continued until all supply and demand values are fulfilled.

3.3. Vogel's Approximation Method (VAM)

Reinfeld and Vogel (1958) proposed VAM by defining penalty as the difference of lowest and next to lowest cost in each row and column of a transportation table. Then they allocate as many units as possible to the least-cost cell corresponding to the highest penalty in the row or column. Ties are broken arbitrarily. The satisfied row or column is deleted. Penalties are revised and the procedure is continued until all the rim conditions are satisfied.

3.4. Extremum Difference Method (EDM)

Kasana and Kumar (2005) presented EDM by calculating the penalty as the difference between the highest and lowest unit transportation cost in each row and column of a transportation table. Then they allocate maximum possible amount to the minimum cost cell corresponding to the highest penalty. In case of ties, they choose arbitrarily. No further consideration is done for the row or column which is satisfied. Penalties are revised and the procedure repeated successively until all units are supplied.

3.5. Russel's Approximation Method (RAM)

Edward J. Russell (1969) introduced RAM and determines the penalty for each cell of the transportation table by subtracting corresponding row and column highest element of every cell from the respective element. Then the author makes maximum possible allocation to the cell having the smallest penalty. After breaking the ties randomly the satisfied row or column are deleted. Revised the penalties and repeats the method in succession until all units are completed.

3.6. Highest Cost Difference Method (HCDM)

Abdur Rashid (2011) suggested HCDM by introducing penalty as the difference between the highest and the next to highest costs for all rows and columns of a transportation table. Then allocate highest possible amount to the lowest cost cell in the row or column corresponding to maximum penalty. If a tie occurs, author breaks it randomly. Then the satisfied row or column has been deleted. Revised the penalties and continue the procedure until all rows and columns are satisfied.

3.7. TOCM-LCM Approach

Kirca and Satir (1990) first define the Total Opportunity Cost Matrix (TOCM) as the sum of Row Opportunity Cost Matrix (ROCM) and Column Opportunity Cost Matrix (COCM). Where, the ROCM is generated by subtracting the lowest cost of each row from the other cost elements in that row and, the COCM is generated by subtracting the lowest cost of each column from the other cost elements in that column. Kirca and Satir then essentially apply the Least Cost Method with some tie-breaking policies on the TOCM to determine the feasible solution of the transportation problem.

3.8. TOCM-VAM Approach

Mathirajan and Meenakshi (2004) extended TOCM of Kirca and Satir by using VAM procedure on the TOCM. They consider TOCM and allocate as like as VAM techniques. Finally, they calculate the total transportation cost for the feasible allocations as sum of the product of the original transportation cost matrix and corresponding allocated value of the transportation table.

3.9. TOCM-EDM Approach

Md. Amirul Islam *et al.* (2012a) extended TOCM of Kirca and Satir by applying EDM techniques on the TOCM to determine the initial basic feasible solution of transportation problem. Finally, they compute the total transportation cost as like as the TOCM-VAM procedure.

3.10. TOCM-HCDM Approach

Md. Amirul Islam *et al.* (2012b) again applied HCDM on TOCM to calculate the initial basic feasible solution of transportation problem. Finally, they compute the total transportation cost following the same procedure of TOCM-VAM.

3.11. TOCM-SUM Approach

Aminur Rahman Khan *et al.* (2015a) calculates the pointer cost as the sum of all entries in the respective row or column of the TOCM, proposed by Kirca and Satir to find the feasible solution of the transportation problem.

Finally, they compute the total transportation cost as like as the TOCM-VAM procedure.

3.12. TOCM-RAM Approach

Aminur Rahman Khan *et al.* (2015b) applied RAM on TOCM to compute the initial basic feasible solution of transportation problem.

4. Methods

Twenty sample transportation problem of different sizes, selected at random from Mhlanga *et al.* (2014), Aminur Rahman Khan *et al.* (2011; 2012; 2015a; 2015b, 2015c), Deshmukh (2012), Edward J. Russell (1969), Md. Amirul Islam *et al.* (2012), Md. Ashraful Babu *et al.* (2013; 2014a; 2014b), Mohammad Kamrul Hasan (2012), Mollah Mesbahuddin Ahmed *et al*. (2014; 2015; 2016), Wagener (1965) and Utpal Kanti Das *et al.* (2014a; 2014b) in order to test the exactness of our techniques. We also calculate the Average Relative Deviation (ARD) (Kulkarni, 2015), which indicates the average performance of various techniques with respect to the optimal solution is compared over the number of problem-instances. Following formulas are to be used for the comparison.

$$ARD(H)=\frac{1}{N}\sum_{i=1}^{N} RD(H,i) \tag{2}$$

and $$RD(H,i)=\frac{IBFS\,Cost-OBFS\,Cost}{OBFS\,Cost}\text{, } i=1, 2, \ldots\ \ldots, N \tag{2.1}$$

Here, *ARD*(*H*) – average relative deviation of the given heuristic method H; *RD*(*H,i*) – relative deviation of the ith problem for the given heuristic method H; N – the number of problem-instances over which the average is compared; IBFS – Initial Basic Feasible Solution; OBFS – Optimal Basic Feasible Solution.

5. Results Illustration

Table 1 and 1a shows the identical output obtained by both manually and using our Matlab code as well as comparison of different solution obtained by various methods by means of the above twenty sample examples. They also show the average relative deviation of various algorithms with respect to the optimal solution.

Table 1
List of Solutions Obtained by Both Manually and Using Matlab Code

Example			Different Solution				
No.	Source	Size	NWC	LCM	VA	ED	RAM
1	Md. A. Islam *et al.* (2012)	3×3	131	131	131	131	131
2	Mollah M. Ahmed *et al.*	3×3	1500	1450	1500	1390	1390
3	Mollah M. Ahmed *et al.*	3×3	545	433	425	439	425
4	Aminur R. Khan *et al.*	3×4	273	231	204	218	200
5	Aminur R. Khan *et al.*	3×4	1265	1165	1220	1165	1165
6	Aminur R. Khan (2012)	3×4	520	475	475	475	475
7	Deshmukh N.M. (2012)	3×5	363	295	290	295	295
8	Mollah M. Ahmed *et al.*	4×3	102	83	76	80	82
9	Md. Ashraful Babu *et al.*	4×4	540	435	470	415	420
10	Md. Ashraful Babu *et al.*	4×4	425	310	285	295	285
11	Md. Ashraful Babu *et al.*	4×5	635	510	510	510	510
12	Mohammad K. Hasan (2012)	4×5	670	420	420	420	420
13	Mhlanga A. (2014)	4×5	560	408	322	318	318
14	Aminur R. Khan (2011)	4×6	95	68	68	68	71
15	Deshmukh N.M. (2012)	4×6	139	112	112	112	115
16	Utpal Kanti Das *et al.* (2014)	5×5	1870	1685	1545	1685	1730
17	Edward J. Russell (1969)	5×5	1994	1123	1104	1102	1103
18	Wagener Ulrich A. (1965)	5×6	129	134	116	121	118
19	Utpal Kanti Das *et al.* (2014)	5×7	3180	1900	1920	2070	1930
20	Aminur R. Khan *et al.*	6×6	4285	2455	2310	2580	2700
Average Relative Deviation (ARD)			0.38	0.07	0.03	0.04	0.04

Table 1a
List of Solutions Obtained by Both Manually and Using Matlab Code

No.	Opt. Sol.	Different Solution						
		HCDM	TOCM-					
			LCM	VAM	EDM	HCDM	SUM	RAM
1	131	131	131	131	131	131	131	131
2	1390	1390	1450	1500	1390	1390	1440	1390
3	425	425	433	425	439	439	439	439
4	200	242	204	204	231	255	200	200
5	1160	1165	1165	1165	1165	1165	1280	1165
6	435	475	475	435	475	475	520	475
7	290	321	295	290	295	295	290	290
8	76	81	83	76	80	81	76	81
9	410	415	435	430	415	415	455	430
10	285	310	305	285	285	335	285	285
11	510	520	510	510	510	520	510	510

Table 1a

Continuation

No.	Opt. Sol.	Different Solution						
		HCDM	TOCM-					
			LCM	VAM	EDM	HCDM	SUM	RAM
12	420	440	420	420	420	440	420	420
13	316	372	408	322	322	372	364	322
14	68	70	70	68	72	100	79	71
15	105	114	114	112	116	144	123	115
16	1475	1850	1760	1515	1685	1850	1545	1600
17	1102	1215	1123	1104	1102	1433	1127	1103
18	116	131	134	116	125	128	123	118
19	1900	1960	1900	1900	2070	1930	2100	1900
20	2170	2620	2470	2170	2470	2470	2170	2800
ARD		0.08	0.07	0.01	0.05	0.13	0.06	0.04

Average Relative Deviation of various techniques for transportation problem obtained from Table 1 and 1a are shown in Chart 1 below:

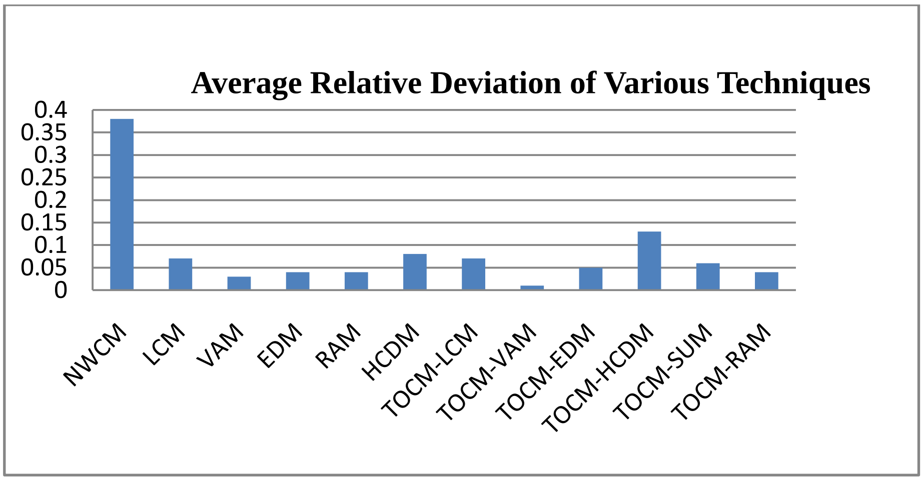


Chart 1 – Average Relative Deviation of various techniques.

According to the results shown in Table 1, 1a and Chart 1 we can conclude that the TOCM-VAM technique performs better than all other techniques (ARD is 0.01) over the number of problem-instances. VAM (ARD is 0.03), EDM (ARD is 0.04), RAM (ARD is 0.04) and TOCM-RAM (ARD is 0.04) perform after TOCM-VAM technique. NWCM (ARD is 0.38) gives lowest results among all techniques considered in this article.

6. Conclusion

Identical result with the manual solution proves the correctness of our techniques described in this study. But the results using different algorithms are different. So, using our proposed idea, researcher can easily perform all the techniques on any problem and then choose the best solution. The ARD indicates that the TOCM-VAM technique performs better than other techniques with respect to the optimal solution over the twenty randomly selected problems. The detailed procedure to the solution of the problems as well as the developed Matlab code are not shown in the works due to space considerations, and are available from the author.

Acknowledgements. The first author acknowledges the financial support provided by the EU Erasmus Mundus Project-cLINK, Grant Agreement No: 212-2645/001-001-EM, Action 2.

REFERENCES

Abdur Rashid, *An Effective Approach for Solving Transportation Problems*, Jahangirnagar Journal of Mathematics and Mathematical Sciences, **26**, 101-107, 2011.

Abdur Rashid, Syed Sabbir Ahmed, Md. Sharif Uddin, *Development of a New Heuristic for Improvement of Initial Basic feasible solution of a Balanced Transportation Problem*, Jahangirnagar University Journal of Mathematics and Mathematical Sciences, **28**, 105-112, 2013.

Aminur Rahman Khan, *A Re-solution of the Transportation Problem: An Algorithmic Approach*, Jahangirnagar University Journal of Science, **34**, 2, 49-62, 2011.

Aminur Rahman Khan, *Analysis and Re-solution of the Transportation Problem: A Linear Programming Approach,* M. Phil. Thesis, Dept. of Mathematics, Jahangirnagar University, 2012.

Aminur Rahman Khan, Adrian Vilcu, Nahid Sultana, Syed Sabbir Ahmed, *Determination of Initial Basic Feasible Solution of a Transportation Problem: A TOCM-SUM Approach*, Bul. Inst. Polit. Iași, Secţia Automatică și Calculatoare, **LXI (LXV)**, *1*, 39-49, 2015a.

Aminur Rahman Khan, Adrian Vilcu, Md. Sharif Uddin, Florina Ungureanu, *A Competent Algorithm to find the Initial Basic Feasible Solution of Cost Minimization Transportation Problem*, Bul. Inst. Polit. Iași, s. Automatică și Calculatoare, **LXI (LXV)**, 2, 71-83, 2015b.

Aminur Rahman Khan, Avishek Banerjee, Nahid Sultana, M. Nazrul Islam, *Solution Analysis of a Transportation Problem: A Comparative Study of Different Algorithms*', Accepted for Publication in the Bul. Inst. Polit. Iași, s. Textile. Leathership, in issue of 2015c.

Charnes A., Cooper W.W., Henderson A., *An Introduction to Linear Programming*, John Wiley & Sons, New York, 1953.

Dantzig G.B., *Application of the Simplex Method to a Transportation Problem*, Activity Analysis of Production and Allocation (T.C. Koopmans Ed.), New York: John Wiley and Sons, 359-373, 1951.

Deshmukh N.M., *An Innovative Method for Solving Transportation Problem*, International Journal of Physics and Mathematical Sciences, **2**, *3*, 86-91, 2012.

Edward J. Russell, *Extension of Dantzig's Algorithm to Finding an Initial Near-Optimal Basis for the Transportation Problem*, Operations Research, **17**, *1*, 187-191, 1969.

Ferguson R.O., *Linear Programming Principles and Application*, Proceedings of the National Capital Conference on New Horizons of Industrial Engineering, Washington, D.C., 1-11, 1955.

Hamdy A.T., *Operations Research: An Introduction*, 8th Edition, Pearson Prentice Hall, Upper Saddle River, New Jersey 07458, 2007.

Hitchcock F.L., *The Distribution of a Product from Several Sources to Numerous Localities*, Journal of Mathematics and Physics, **20**, 224-230, 1941.

Juman Z.A.M.S., Hoque M.A., Buhari M.I., *A Study of Transportation Problem and Use of Object Oriented Programming*, 3rd International Conference on Applied Mathematics and Pharmaceutical Sciences (ICAMPS'2013), Singapore, 353-354, 2013.

Juman Z.A.M.S., Hoque M.A., *An Efficient Heuristic to Obtain a Better Initial Feasible Solution to the Transportation Problem*, Applied Soft Computing, **34**, 813-826, 2015.

Kasana H.S., Kumar K.D., *Introductory Operations Research: Theory and Applications*, Springer International Edition, New Delhi, 2005.

Kirca Omer, Satir Ahmet, *A Heuristic for Obtaining an Initial Solution for the Transportation Problem*, Journal of the Operational Research Society, **41**, 865–871, 1990.

Koopmans T.C., *Optimum Utilization of the Transportation System*, Econometrica, **17**, 3-4, 1947.

Kulkarni S.S., *Empirical Analysis Based on Comparison Among Various Methods of Obtaining IBFS to Unbalanced Transportation Problem*, International Journal of Advanced Research in Management and Social Sciences, **4**, *12*, 1-7, 2015.

M. Sharif Uddin, Sayedul Anam, Abdur Rashid, Aminur R. Khan, *Minimization of Transportation Cost by Developing an Efficient Network* Model, Jahangirnagar Journal of Mathematics & Mathematical Sciences, **26**, 123-130, 2011.

Mathirajan M., Meenakshi B., *Experimental Analysis of Some Variants of Vogel's Approximation Method*, Asia-Pacific Journal of Operational Research, **21**, *4*, 447-462, 2004.

Md. Amirul Islam, Aminur Rahman Khan, M. Sharif Uddin, M. Abdul Malek, *Determination of Basic Feasible Solution of Transportation Problem: A New Approach*, Jahangirnagar University Journal of Science, **35**, *1*, 101-108, 2012a.

Md. Amirul Islam, Md. Masudul Haque, Md. Sharif Uddin, *Extremum Difference Formula on Total Opportunity Cost: A Transportation Cost Minimization Technique*, Prime University Journal of Multidisciplinary Quest, **6**, *1*, 125-130, 2012b.

Md. Ashraful Babu, Md. Abu Helal, Mohammad Sazzad Hasan, Utpal Kanti Das, *Lowest Allocation Method* (*LAM*)*: A New Approach to Obtain Feasible Solution of Transportation Model*, International Journal of Scientific and Engineering Research, **4**, *11*, 1344-1348, 2013.

Md. Ashraful Babu, Md. Abu Helal, Mohammad Sazzad Hasan, Utpal Kanti Das, *Implied Cost Method* (*ICM*)*: An Alternative Approach to Find the Feasible Solution of Transportation Problem*, Global Journal of Science Frontier Research-F: Mathematics and Decision Sciences, **14**, *1*, 5-13, 2014a.

Md. Ashraful Babu, Utpal Kanti Das, Aminur Rahman Khan, Md. Sharif Uddin, *A Simple Experimental Analysis on Transportation Problem: A New Approach to Allocate Zero Supply or Demand for All Transportation Algorithm*, International Journal of Engineering Research & Applications (IJERA), **4**, *1*, 418-422, 2014b.

Md. Main Uddin, Aminur Rahman Khan, Sushanta Kumer Roy, Md. Sharif Uddin, *A New Approach for Solving Unbalanced Transportation Problem due to Additional Supply*, Accepted for Publication in the Bul. Inst. Polit. Iași, s. Textile. Leathership, in issue of 2015.

Mhlanga Adwell, Nduna Immaculate S., Matarise Florence, Machisvo Albert, *Innovative Application of Dantzig's North–West Corner Rule to Solve a Transportation Problem*, International Journal of Education and Research, **2**, *2*, 1-12, 2014.

Mollah Mesbahuddin Ahmed, Abu Sadat Muhammad Tanvir, Shirin Sultana, Sultan Mahmud, Md. Sharif Uddin, *An Effective Modification to Solve Transportation Problems: A Cost Minimization Approach*, Annals of Pure and Applied Mathematics, **6**, *2*, 199-206, 2014a.

Mollah Mesbahuddin Ahmed, *Algorithmic Approach to Solve Transportation Problems: Minimization of Cost and Time*, M. Phil. Thesis, Dept. of Mathematics, Jahangirnagar University, 2014b.

Mollah Mesbahuddin Ahmed, Aminur Rahman Khan, Md. Sharif Uddin, Faruque Ahmed, *A New Approach to Solve Transportation Problems*, Open Journal of Optimization, **5**, *1*, 22-30, 2016.

Nabendu Sen, Tanmoy Som, Banashri Sinha, *A Study of Transportation Problem for an Essential Item of Southern Part of North Eastern Region of India as an OR Model and Use of Object Oriented Programming*, International Journal of Computer Science and Network Security, **10**, *4*, 78-86, 2010.

Nizam Uddin Ahmed, Aminur Rahman Khan, Md. Sharif Uddin, *Solution of Mixed Type Transportation Problem: A Fuzzy Approach*, Bul. Inst. Polit. Iași, s. Automatică și Calculatoare, **LXI (LXV)**, *2*, 19-31, 2015.

Reinfeld N.V., Vogel W.R., *Mathematical Programming*, Englewood Cliffs, NJ: Prentice-Hall, 1958.

Sayedul Anam, Aminur Rahman Khan, Md. Minarul Haque, Reza Shahbaz Hadi, *The Impact of Transportation Cost on Potato Price: A Case Study of Potato Distribution in Bangladesh,* The International Journal of Management, **1**, *3,* 1-12, 2012.

Taghrid Imam, Gaber Elsharawy, Mohamed Gomah, Iman Samy, *Solving Transportation Problem Using Object-Oriented Model*, International Journal of Computer Science and Network Security, **9**, *2*, 353-361, 2009.

Utpal Kanti Das, Md. Ashraful Babu, Aminur Rahman Khan, Md. Abu Helal, Md. Sharif Uddin, *Logical Development of Vogel's Approximation Method (LD-VAM): An Approach to Find Basic Feasible Solution of Transportation Problem*, International Journal of Scientific & Technology Research (IJSTR), **3**, *2*, 42-48, 2014a.

Utpal Kanti Das, Md. Ashraful Babu, Aminur Rahman Khan, Md. Sharif Uddin, *Advanced Vogel's Approximation Method (AVAM): A New Approach to Determine Penalty Cost for Better Feasible Solution of Transportation Problem*, International Journal of Engineering Research & Technology (IJERT), **3**, *1*, 182-187, 2014b.

Wagener Ulrich A., *A New Method of Solving the Transportation Problem*, Operational Research Society, **16**, *4*, 453-469, 1965.

EVALUAREA PERFORMANȚELOR UNOR TEHNICI DE REZOLVARE PENTRU O PROBLEMĂ DE TRANSPORT

(Rezumat)

În literatura de specialitate sunt disponibile multiple tehnici pentru determinarea unei soluții fezabile pentru problema clasică de transport. Pe de o parte, multe dintre aceste tehnici sunt greu de aplicat practic datorită algorimilor complecşi, iar pe de altă parte, este dificil de determinat procedura ce rezolvă optim o anumită instanță a problemei de transport. Având în vedere aceste aspecte vom compara rezultatele unor algoritmi pentru rezolvarea acestei probleme, specificați ca fiind dintre cei mai eficienți. Vom folosi mediul Matlab, creând astfel un instrument prin care un utlizator poate aplica, mai mulți algoritmi simultan pe o instanță numerică și poate alege cea mai bună soluție. Lucrarea determină și performanța medie a tehnicilor selectate față de soluția optimă.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 1-2, 2016
Secţia
AUTOMATICĂ şi CALCULATOARE

CUSTOMIZED VOGEL’S APPROXIMATION METHOD (CVAM) FOR SOLVING TRANSPORTATION PROBLEMS

BY

MOLLAH MESBAHUDDIN AHMED[1], AMINUR RAHMAN KHAN[1,2,*], FARUQUE AHMED[1] and MD. SHARIF UDDIN[1,3]

[1]Jahangirnagar University, Dhaka-1342, Bangladesh,
Department of Mathematics
[2]“Gheorghe Asachi” Technical University of Iaşi, Romania,
Department of Management Engineering
[3]Ruse University, Ruse, Bulgaria,
Department of Mathematics

Received: March 31, 2016
Accepted for publication: May 3, 2016

Abstract. Transportation cost helps a company to keep the product cost low and also to make profit. A business owner must understand the effect of transportation cost on the production. In this research, we propose a customized Vogel’s Approximation Method (CVAM) for finding an initial basic feasible solution (IBFS) for cost minimizing transportation problems. We observe that customized procedure is computationally easier and yielding comparatively better solution.

Keywords: Transportation Cost; Transportation Problem; VAM; IBFS.

2010 Mathematics Subject Classification: 90B50, 90C08.

[*]Corresponding author; *e-mail*: aminurju@yahoo.com

1. Introduction

Transportation cost is one of the main factors of a business to maximize profit. So, reducing transportation cost is to be targeted to maximize profit. Because, in today's highly competitive market, reducing costs of materials, equipment and workers are generally very difficult to maintain the quality of a product. Industries should way out the transportation route to have commodities shipped to them or to customers with minimum transportation cost.

The transportation problem (TP) is a special brand of linear programming problem. In the general form, a TP has a number of origins and a number of destinations. The origin of a TP is the place from which shipments are dispatched. The destination of a TP is the place to which shipments are transported. A certain amount of a particular consignment is available in each origin. Similarly, each destination has a certain requirement. The unit transportation cost is the cost of transporting one unit of the consignment from an origin to a destination. The TP indicates the amount of consignment to be transported from various origins to different destinations so that the total transportation cost is minimized without violating the availability constraints and the requirement constraints.

The TP was first introduced in 1941 when F.L. Hitchcock (Hitchcock, 1941) presented a study entitled "The Distribution of a Product from Several Sources to Numerous Localities". The presentation is regarded as the first contribution to the solution of TPs. In 1947, T.C. Koopmans (Koopmans, 1947) also presented a study called "Optimum Utilization of the Transportation System". These two contributions are the fundamentals for the progress of TPs.

TPs can be solved by using general simplex based integer programming methods (Hamdy, 2007); however it involves time-consuming computational procedure. There are specialized algorithms for solving TPs those are much more efficient than the simplex algorithm (Dantzig, 1951). The basic steps to solve TPs are:

Step 1. Determination of the initial feasible solution,

Step 2. Determination of optimal solution using the IBFS.

In this study, basic idea is to get better IBFS for the TP.

The first systematic procedure in solving the TP was developed by Hitchcock (1941) and referred as Least Cost Method (LCM), which consists in allocating as much as possible in the lowest cost cell of the TP in making allocation in every stage. Then the North West Corner Method (NWCM) by Charnes *et al*. (1953), which consider the north-west corner cell cost at every stage of allocation. Vogel's Approximation Method (VAM) (Reinfeld and Vogel, 1958) also developed on allocating in cost cells and comparatively provides better IBFS. These classical methods are well discussed in all the operation research books (Kasana and Kumar, 2005; Hamdy, 2007).

Again, several models were formulated also there were many trials to minimize the transportation cost; Kirca and Satir (1990) developed a heuristic to obtain efficient initial basic feasible solutions, called Total Opportunity Cost Method (TOCM). Balakrishnan (1990) proposed a modified version of VAM for unbalanced transportation problems. Adlakha and Kowalski (2003) presented a simple heuristic algorithm for the solution of small fixed-charge transportation problems. Mathirajan and Meenakshi (2004) extended TOCM using the VAM procedure. They coupled VAM with TOCM and achieved very efficient initial solutions. Aminur Rahman Khan *et al*. (2015a; 2015b; 2015c) have used TOCM and define the pointer cost as the sum of all entries in the respective row or column of the TOCM and allocate to the minimum cost cell corresponding to the highest pointer cost. Md. Amirul Islam *et al.* (2012a; 2012b) applied Extremeum Difference Method on TOCM, and allocate to the minimum cost cell corresponding to the highest distribution indicator and again HCDM on TOCM. Aminur Rahman Khan (2011; 2012) presented HCDM by defining pointer cost as the difference of highest and next to highest cost in each row and column of a transportation table and allocate to the minimum cost cell corresponding to the highest three pointer cost. Md. Ashraful Babu *et al.* (2013) applied first allocation on lowest cost cell which appears along lowest demand/supply, Utpal Kanti Das *et al*. (2014) brought out a logical development in the solution procedure of VAM, Mollah Mesbahuddin Ahmed *et al*. (2016) applied the first allocation on lowest odd valued cost cell and proposed to form an allocation table to allocate rest of the allocation for finding an IBFS.

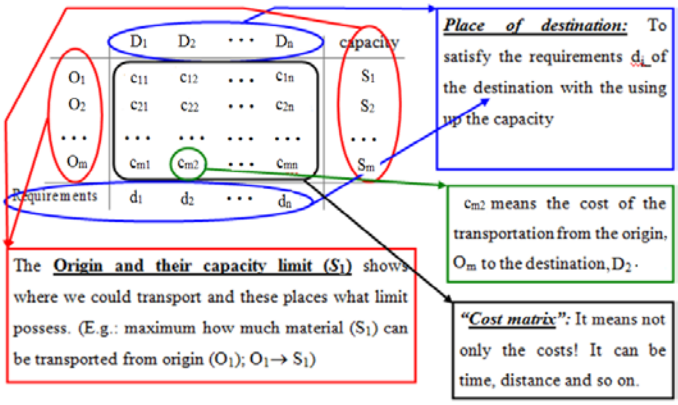
In this research, we studied TPs and its solution procedure in details. We also studied various methods for finding an IBFS for TPs, and observed that there is no unique method which can be claimed as the best method for finding an IBFS. Again, it is very difficult to measure the quality of a solution procedure. We also observed that some of the methods are yielding optimal or near optimal solution in most of the cases, but the computational procedure is time consuming and complicated. In this study we try to bring some change in the solution procedure of the existing methods with an aim to make the solution procedure computationally easier. We work on VAM and observe that computing penalty cost in each and every allocation is a computationally complicated and time consuming procedure. So, we try to avoid computing penalty cost for each and every allocation with an aim to simplify the solution procedure. Al last we develop a method by customizing existing VAM. In the proposed method, we need to compute penalty cost only for the first allocation. And for rest of the allocations, we find out alternative allocation procedure.

In this work we also solve several numbers of numerical problems using existing methods and proposed method. We also carry out comparative study to measure the effectiveness of the proposed procedure.

2. Representation and Mathematical Formulation of TP

To describe the transportation problem, following notations are to be used:
m – Total number of sources/origins;
n – Total number of destinations;
S_i – Amount of supply at source i;
d_j – Amount of demand at destination j;
c_{ij} – Unit transportation cost from source i to destination j;
x_{ij} – Amount to be shipped from source i to destination j.

The most general and accepted form of the transportation problem is presented by the following Fig. 1:



	D₁	D₂	• • •	Dₙ	capacity
O₁	c_{11}	c_{12}	• • •	c_{1n}	S_1
O₂	c_{21}	c_{22}	• • •	c_{2n}	S_2
• • •	• • •	• • •	• • •	• • •	• • •
Oₘ	c_{m1}	c_{m2}	• • •	c_{mn}	S_m
Requirements	d_1	d_2	• • •	d_n	

Place of destination: To satisfy the requirements d_j of the destination with the using up the capacity

c_{m2} means the cost of the transportation from the origin, O_m to the destination, D_2.

The **Origin and their capacity limit (*S*₁)** shows where we could transport and these places what limit possess. (E.g.: maximum how much material (S_1) can be transported from origin (O_1); $O_1 \rightarrow S_1$)

"Cost matrix": It means not only the costs! It can be time, distance and so on.

Fig. 1 – Transportation Problem Scheme.

Network representation of transportation model is represented by the following network in Fig. 2:

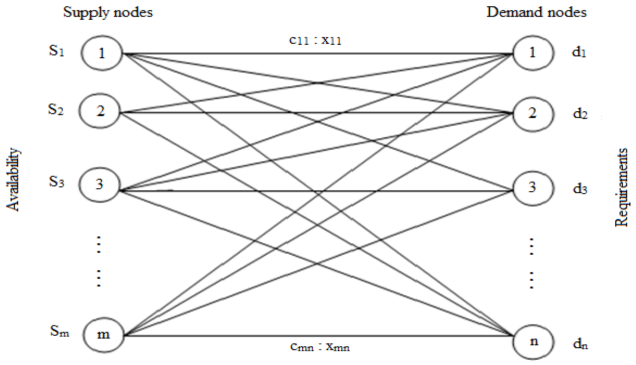


Fig. 2 – Network Diagram for Transportation Problem.

A specially designed table is constructed to solve a TP systematically, which is called transportation table, is shown in Table 1.

Table 1
Tabular form of Transportation Table

Factory /Origin	Destination / Warehouse					Available products/ Supply
	1	2	3	………..	n	
1	c_{11}: x_{11}	c_{12}: x_{12}	c_{13}: x_{13}	………..	c_{1n}: x_{1n}	S_1
2	c_{21}: x_{21}	c_{22}: x_{22}	c_{23}: x_{23}	………..	c_{2n}: x_{2n}	S_2
3	c_{31}: x_{31}	c_{32}: x_{32}	c_{33}: x_{33}	………..	c_{3n}: x_{3n}	S_3
⋮	⋮	⋮	⋮	⋮	⋮	⋮
m	c_{m1}: x_{m1}	c_{m2}: x_{m2}	c_{m3}: x_{m3}	………..	c_{mn}: x_{mn}	S_m
Demand	d_1	d_2	d_3	………..	d_n	

The objective of the model is to determine the unknowns' x_{ij} that will minimize the total transportation cost while satisfying the supply and demand restrictions. Basing on this objective The Hitchcock-Koopman's transportation problem is expressed as a linear transportation model as follows:

$$\text{Minimize:} \quad z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

$$\text{subject to :} \quad \sum_{j=1}^{n} x_{ij} \le S_i \quad ; \quad i = 1, 2, \dots \dots, m$$

$$\sum_{i=1}^{m} x_{ij} \ge d_j \quad ; \quad j = 1, 2, \dots \dots, n$$

$$x_{ij} \ge 0, \text{ for all } i \text{ and } j \qquad (1)$$

3. Proposed Procedure

In this proposed method first allocation is made by using the solution procedure of VAM. For this, we name the proposed method "Customized Vogel's Approximation Method (**CVAM**)". The projected procedure is illustrated below:

▪ Step-1: Balance the transportation problem.

▪ Step-2: Find the difference between the smallest and second smallest elements along every row and column. This difference is known as penalty. In the transportation table, enter the column penalties below the corresponding columns and row penalties to the right of the corresponding rows.

▪ Step-3: Select the lowest cost cell corresponding to the highest penalty and allocate maximum possible amount to this cell. If tie occurs in the maximum penalty, choose the penalty along which minimum cost cell appears. In case of ties for minimum cost cells, select the cell where maximum allocation can be made. If a tie occurs again, select the cell for which sum of demand and supply is maximum in the original transportation table. Finally if all these are same, choose the cell arbitrarily.

▪ Step-4: Adjust the supply and demand requirements in the respective rows and columns. Then following cases arise:

First Case: If the allocation equals the supply, cross out the row. Now complete the column allocation along the basic cell by making the allocation(s) in the smallest cost cell(s) continuously. Consider that the column is exhausted for the allocation x_{pq} in the cell (p, q). Now follow the same procedure to complete the allocation along p-th row and continue the process until all the rim conditions are satisfied.

Second Case: If the allocation equals the demand, cross out the column. Now complete the row allocation along the basic cell by making the allocation(s) in the smallest cost cell(s) continuously. Finally follow the procedure explained in first case to complete the solution.

Third Case: If the allocation equals both the demand and the supply, assign a zero in the next smallest cost cell which are along the row or column of the allocated cell and cross out both the row and column. Consider that the zero allocation x_{st} is made in the cell (s, t) which is along the exhausted row. Now complete the allocation along t-th column following the process described in First Case and Second Case to complete the allocations.

Fourth case: For any allocation, other than first allocation made along the row/column satisfies both of them, in such case find the smallest cost cell which is along the column/row and assign a zero in that cell and continue the process described in first and second cases to complete the allocations.

▪ Step-5: Finally calculate the total transportation cost.

4. Numerical Examples

Only the theoretical explanation cannot ensures the acceptance of a model in the practical field. For this we solve ten problems from some reputed journals and books. These problems along with their IBFS obtained by CVAM are tabulated in the following Table 2.

Table 2
Numerical Examples

Example No	Data of the problems and Sources	Total transportation cost
Ex-1	$[c_{ij}]_{3x3}$=[6 4 1; 3 8 7; 4 4 2] $[s_i]_{3x1}$=[50, 40, 60] $[d_j]_{1x3}$=[20, 95, 35]; (Desmukh, 2012)	**555**
Ex-2	$[c_{ij}]_{3x3}$=[4 3 5; 6 5 4; 8 10 7] $[s_i]_{3x1}$=[90, 80, 100] $[d_j]_{1x3}$=[70, 120, 80]; (Mollah Mesbahuddin Ahmed *et al*., 2016)	**1390**
Ex-3	$[c_{ij}]_{3x4}$=[3 6 8 4; 6 1 2 5; 7 8 3 9] $[s_i]_{3x1}$=[20, 28, 17] $[d_j]_{1x4}$=[15, 19, 13, 18]; (Aminur Rahman Khan *et al*., 2015a, 2015b, 2015c)	**200**
Ex-4	$[c_{ij}]_{3x4}$=[3 1 7 4; 2 6 5 9; 8 3 3 2] $[s_i]_{3x1}$=[300, 400, 500] $[d_j]_{1x4}$=[250,350,400,200]; (Mollah Mesbahuddin Ahmed *et al*., 2016)	**2850**
Ex-5	$[c_{ij}]_{4x4}$=[5 3 6 10; 6 8 10 7; 3 1 6 7; 8 2 10 12] $[s_i]_{4x1}$=[30, 10, 20, 10] $[d_j]_{1x4}$=[20, 25, 15, 10]; (Md. Ashraful Babu *et al*., 2014)	**285**
Ex-6	$[c_{ij}]_{3x5}$=[5 7 10 5 3; 8 6 9 12 14; 10 9 8 10 15] $[s_i]_{3x1}$=[5, 10, 10] $[d_j]_{1x5}$=[3, 3, 10, 5, 4]; (Ray and Hossain, 2007)	**183**
Ex-7	$[c_{ij}]_{3x5}$=[4 1 2 4 4; 2 3 2 2 2; 3 5 2 4 4] $[s_i]_{3x1}$=[60, 35, 40] $[d_j]_{1x5}$=[22, 45, 20, 18, 30]; (Desmukh, 2012)	**290**
Ex-8	$[c_{ij}]_{4x3}$=[2 7 4; 3 3 1 ; 5 4 7; 1 6 2] $[s_i]_{4x1}$=[5, 8, 7, 14] $[d_j]_{1x3}$=[7, 9, 18]; (Mollah Mesbahuddin Ahmed *et al*., 2014a, 2014b)	**80**
Ex-9	$[c_{ij}]_{5x5}$=[68 35 4 74 15; 57 88 91 3 8; 91 60 75 45 60; 52 53 24 7 82; 51 18 82 13 7] $[s_i]_{5x1}$=[18, 17, 19,13, 15] $[d_j]_{1x5}$=[16, 18, 20, 14, 14]; (Shweta Sing *et al*., 2012)	**2224**
Ex-10	$[c_{ij}]_{5x7}$=[12 7 3 8 10 6 6; 6 9 7 12 8 12 4; 10 12 8 4 9 9 3; 8 5 11 6 7 9 3; 7 6 8 11 9 5 6] $[s_i]_{5x1}$=[60, 80, 70, 100, 90] $[d_j]_{1x7}$=[20, 30, 40, 70, 60, 80, 100]; (Utpal Kanti Das *et al*., 2014)	**1920**

5. Illustrative Solution of an Example

Solution of a numerical example is illustrated with an aim to make the readers acquainted with CVAM. Step by step solution of **Ex-4** is given in Table 3 below:

Table 3
IBFS of Ex-4 Using CVAM

	1	2	3	4	Supply	Row Penalty
1	3	300 1	7	4	300	2
2	250 2	6	150 5	9	400	3
3	8	50 3	250 3	200 2	500	1
Demand	250	350	400	200		
Column Penalty	1	2	2	2		

• According to Step-1: It is found that the given problem is balanced. Because of the sum of supplies = sum of the demands =1200.

• As per Step-2: Column and Row penalty costs 1, 2, 2, 2 and 2, 3, 1 are respectively entered below the column and after the row.

• Now 3 is the highest penalty cost appears along the second row. As per step-3, along this highest penalty cost, 2 is the smallest cost cell appears in the cell (2, 1). Allocate 250= min (400, 250) in the cell (2, 1) and Cross out the 1st column as this allocation satisfies this column and along 2nd row supply is reduced to 150=(400-250).

• Now as per the other steps, Now the smallest cost cell along 2nd row is 5 appears in (2, 3) and allocate the reduced supply 150 in this cell. In this case 2nd row is exhausted and the demand along 2nd column is reduced to 250=(400-250).

• By following the same procedure other allocations are made. These allocations are 250 in the cell (3, 3); 200 in the cell (3, 4); 50 in the cell (3, 2) and 300 in the cell (1, 2) respectively.

• Finally total transportation cost is,

$$300x1+250x2+150x5+50x3+250x3+200x2=2850.$$

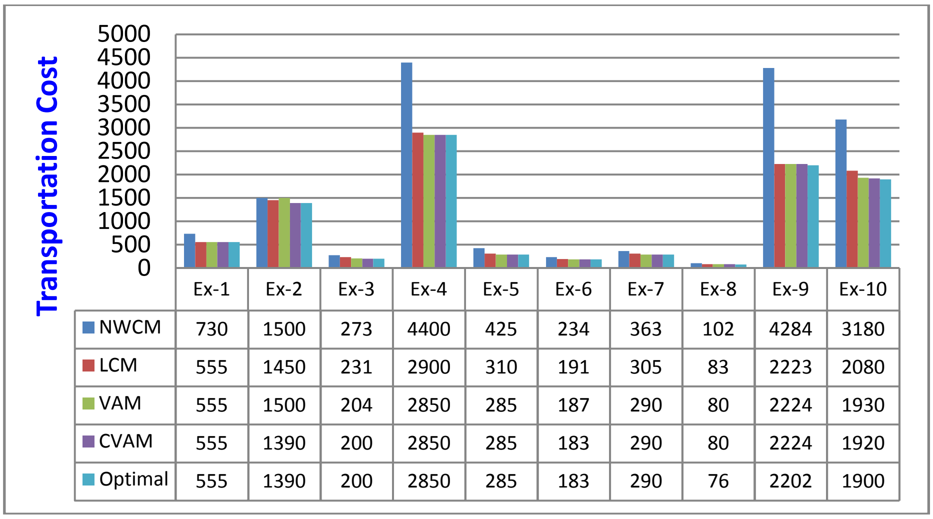
6. Comparative Study and Analysis

We study the IBFS obtained various methods including the proposed method. We also carry out a comparative study in different ways to analyze the perfectness and to justify the usefulness of the proposed method. These comparisons are shown in the following tables and charts:

Table 4
A Comparative Study of Initial Basic Feasible Solution

Example	Initial Basic Feasible Solution (I_R)				Optimal Result (F_R)	Performance of the Result			
	NWCM	LCM	VAM	CVAM		NWCM	LCM	VAM	CVAM
Ex-1	730	555	555	555	555	Uopt	Eopt	Eopt	Eopt
Ex-2	1500	1450	1500	1390	1390	Uopt	Uopt	Uopt	Uopt
Ex-3	273	231	204	200	200	Uopt	Uopt	Uopt	Eopt
Ex-4	4400	2900	2850	2850	2850	Uopt	Uopt	Eopt	Eopt
Ex-5	425	310	285	285	285	Uopt	Uopt	Eopt	Eopt
Ex-6	234	191	187	183	183	Uopt	Uopt	Uopt	Eopt
Ex-7	363	305	290	290	290	Uopt	Uopt	Eopt	Eopt
Ex-8	102	83	80	80	76	Uopt	Uopt	Uopt	Uopt
Ex-9	4284	2223	2224	2224	2202	Uopt	Uopt	Uopt	Uopt
Ex-10	3180	2080	1930	1920	1900	Uopt	Uopt	Uopt	Uopt

Table 4, represents the IBFS obtained by various methods. Performance of the IBFS is also shown in this table. It is observed that in 70%, 40% and 10% cases IBFS obtained respectively by proposed method, VAM and LCM is numerically equal to optimal result. NWCM performs nil in this case. This data speaks the better performance of the proposed method. The graphical representation of IBFS is the mirror of this performance, displayed in Graph 1.



	Ex-1	Ex-2	Ex-3	Ex-4	Ex-5	Ex-6	Ex-7	Ex-8	Ex-9	Ex-10
■ NWCM	730	1500	273	4400	425	234	363	102	4284	3180
■ LCM	555	1450	231	2900	310	191	305	83	2223	2080
■ VAM	555	1500	204	2850	285	187	290	80	2224	1930
■ CVAM	555	1390	200	2850	285	183	290	80	2224	1920
■ Optimal	555	1390	200	2850	285	183	290	76	2202	1900

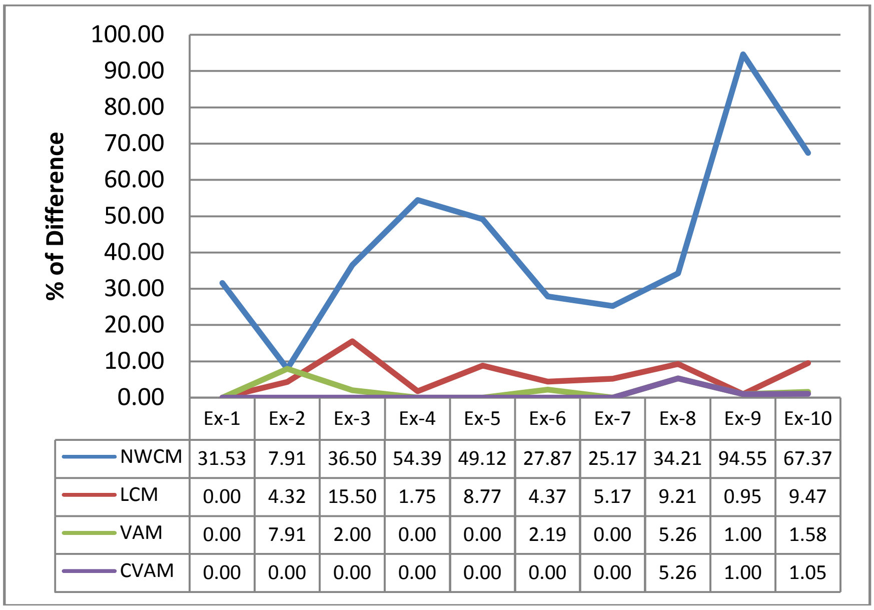
Graph 1 – Graphical representation of IBFS.

Table 5

A Comparative Study of Performance of IBFS

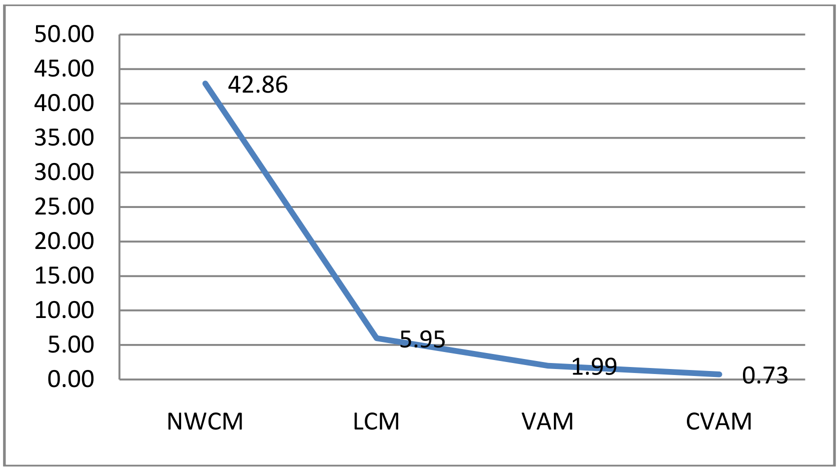
Example	Difference between optimal result and IBFS				Percentage of Difference (POD)			
	NWCM	LCM	VAM	CVAM	NWCM	LCM	VAM	CVAM
Ex-1	175	0	0	0	31.53	0.00	0.00	0.00
Ex-2	110	60	110	0	7.91	4.32	7.91	0.00
Ex-3	73	31	4	0	36.50	15.50	2.00	0.00
Ex-4	1550	50	0	0	54.39	1.75	0.00	0.00
Ex-5	140	25	0	0	49.12	8.77	0.00	0.00
Ex-6	51	8	4	0	27.87	4.37	2.19	0.00
Ex-7	73	15	0	0	25.17	5.17	0.00	0.00
Ex-8	26	7	4	4	34.21	9.21	5.26	5.26
Ex-9	2082	21	22	22	94.55	0.95	1.00	1.00
Ex-10	1280	180	30	20	67.37	9.47	1.58	1.05
				Average of POD	**42.86**	**5.95**	**1.99**	**0.73**

Table 5, represents the difference between IBFS and the Optimal Result. Percentage of Difference (POD) and the Average of POD are also shown in this table. This study is carried out to justify the usual performance of the method. During this process, it is observed that the performance of the proposed method is superior to other methods discussed in this table, which is shown in Graph 2 and Graph 3 respectively.



	Ex-1	Ex-2	Ex-3	Ex-4	Ex-5	Ex-6	Ex-7	Ex-8	Ex-9	Ex-10
NWCM	31.53	7.91	36.50	54.39	49.12	27.87	25.17	34.21	94.55	67.37
LCM	0.00	4.32	15.50	1.75	8.77	4.37	5.17	9.21	0.95	9.47
VAM	0.00	7.91	2.00	0.00	0.00	2.19	0.00	5.26	1.00	1.58
CVAM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.26	1.00	1.05

Graph 2 – Graphical Representation of Performance of IBFS.



Graph 3 – Graphical Representation of Average of POD.

Above discussion ensures the feasibility of the proposed method in solving transportation problems. The customized solution procedure is also computationally easier.

7. Conclusion

Quality of products ensures the reputation of an industry. In the present competitive global market, maintaining this quality has become challenging for earning profit. As transportation cost is one of the many factors to maximize profit. Transportation model provides a powerful framework to industries to determine the better ways to procure and deliver goods with minimum transportation cost. In this study we propose a method named CVAM for finding an IBFS for the minimization of transportation cost. We also illustrate the proposed method in our study and solve few numbers of problems using proposed method. It is observed that proposed method is computationally easier and provides comparatively a better IBFS. We would finally wrap up that our developed algorithm provides a remarkable IBFS by ensuring minimum transportation cost which may be an attractive option to the conventional transportation problem solution methods.

REFERENCES

Adlakha Veena, Kowalski Krzysztof, *A Simple Heuristic for Solving Small Fixed Charge Transportation Problems*, OMEGA: The International Journal of Management Science, **31**, *3*, 205–211, 2003.

Aminur Rahman Khan, *A Re-solution of the Transportation Problem: An Algorithmic Approach*, Jahangirnagar University Journal of Science, **34**, *2*, 49-62, 2011.

Aminur Rahman Khan, *Analysis and Re-solution of the Transportation Problem: A Linear Programming Approach*, M. Phil. Thesis, Dept. of Mathematics, Jahangirnagar University, 2012.

Aminur Rahman Khan, Adrian Vilcu, Nahid Sultana, Syed Sabbir Ahmed, *Determination of Initial Basic Feasible Solution of a Transportation Problem: A TOCM-SUM Approach*, Bul. Inst. Polit. Iași, s. Automatică și Calculatoare, **LXI (LXV)**, *1*, 39-49, 2015a.

Aminur Rahman Khan, Adrian Vilcu, Md. Sharif Uddin, Florina Ungureanu, *A Competent Algorithm to find the Initial Basic Feasible Solution of Cost Minimization Transportation Problem*, Bul. Inst. Polit. Iași, s. Automatică și Calculatoare, **LXI (LXV)**, *2*, 71-83, 2015b.

Aminur Rahman Khan, Avishek Banerjee, Nahid Sultana, M Nazrul Islam, *Solution Analysis of a Transportation Problem: A Comparative Study of Different Algorithms*', Accepted for Publication in the Bul. Inst. Polit. Iași, s. Textile. Leathership, in issue of 2015c.

Balakrishnan N., *Modified Vogel's Approximation Method for the Unbalanced Transportation Problem*, Applied Mathematics Letters, **3**, *2*, 9–10, 1990.

Charnes A., Cooper W.W, Henderson A., *An Introduction to Linear Programming*, John Wiley & Sons, New York, 1953.

Deshmukh N.M., *An Innovative Method for Solving Transportation Problem*, International Journal of Physics and Mathematical Sciences, **2**, 3, 86-91, 2012.

Dantzig G.B., *Application of the Simplex Method to a Transportation Problem*, Activity Analysis of Production and Allocation (T.C. Koopmans Ed.), New York: John Wiley and Sons, 359-373, 1951.

Hamdy A.T., *Operations Research: An Introduction*, 8th Edition, Pearson Prentice Hall, Upper Saddle River, New Jersey 07458, 2007.

Hitchcock F.L., *The Distribution of a Product from Several Sources to Numerous Localities*, Journal of Mathematics and Physics, **20**, 224-230, 1941.

Kasana H.S., Kumar K.D., *Introductory Operations Research: Theory and Applications*, Springer International Edition, New Delhi, 2005.

Kirca Omer, Satir Ahmet, *A Heuristic for Obtaining an Initial Solution for the Transportation Problem*, Journal of the Operational Research Society, **41**, 865–871, 1990.

Koopmans T.C., *Optimum Utilization of the Transportation System*, Econometrica, **17**, 3-4, 1947.

Mathirajan M., Meenakshi B., *Experimental Analysis of Some Variants of Vogel's Approximation Method*, Asia-Pacific Journal of Operational Research, **21**, *4*, 447-462, 2004.

Md. Amirul Islam, Aminur Rahman Khan, M. Sharif Uddin, M. Abdul Malek, *Determination of Basic Feasible Solution of Transportation Problem: A New Approach*, Jahangirnagar University Journal of Science, **35**, *1*, 101–108, 2012a.

Md. Amirul Islam, Md. Masudul Haque, Md. Sharif Uddin, *Extremum Difference Formula on Total Opportunity Cost: A Transportation Cost Minimization Technique*, Prime University Journal of Multidisciplinary Quest, **6**, *1*, 125-130, 2012b.

Md. Ashraful Babu, Md. Abu Helal, Mohammad Sazzad Hasan, Utpal Kanti Das, *Lowest Allocation Method* (*LAM*): *A New Approach to Obtain Feasible Solution of Transportation Model*, International Journal of Scientific and Engineering Research, **4**, *11*, 1344-1348, 2013.

Md. Ashraful Babu, Utpal Kanti Das, Aminur Rahman Khan, Md. Sharif Uddin, *A Simple Experimental Analysis on Transportation Problem: A New Approach to Allocate Zero Supply or Demand for All Transportation Algorith*, International Journal of Engineering Research & Applications (IJERA), **4**, *1*, 418-422, 2014.

Mollah Mesbahuddin Ahmed, Abu Sadat Muhammad Tanvir, Shirin Sultana, Sultan Mahmud, Md. Sharif Uddin, *An Effective Modification to Solve Transportation Problems: A Cost Minimization Approach*, Annals of Pure and Applied Mathematics, **6**, *2*, 199-206, 2014a.

Mollah Mesbahuddin Ahmed, *Algorithmic Approach to Solve Transportation Problems: Minimization of Cost and Time*, M. Phil. Thesis, Dept. of Mathematics, Jahangirnagar University, 2014b.

Mollah Mesbahuddin Ahmed, Aminur Rahman Khan, Md. Sharif Uddin, Faruque Ahmed, *A New Approach to Solve Transportation Problems*, Open Journal of Optimization, **5**, *1*, 22-30, 2016.

Ray G.C., Hossain M.E., *Operation Research*, First Edition, Bangladesh, 237, 2007.

Reinfeld N.V., Vogel W.R., *Mathematical Programming*, Englewood Cliffs, NJ: Prentice-Hall, 1958.

Shweta Sing, G.C. Dubey, Rajesh Shrivastava, *Optimization and Analysis of Some Variants Through Vogel's Approximation Method* (*VAM*), IOSR Journal of Engineering (IOSRJEN), **2**, *9*, 20-30, 2012.

Utpal Kanti Das, Md. Ashraful Babu, Aminur Rahman Khan, Md. Abu Helal, Md. Sharif Uddin, *Logical Development of Vogel's Approximation Method* (*LD-VAM*): *An Approach to Find Basic Feasible Solution of Transportation Problem*, International Journal of Scientific & Technology Research (IJSTR), **3**, *2*, 42-48, 2014.

Utpal Kanti Das, Md. Ashraful Babu, Aminur Rahman Khan, Md. Sharif Uddin, *Advanced Vogel's Approximation Method* (*AVAM*): *A New Approach to Determine Penalty Cost for Better Feasible Solution of Transportation Problem*, International Journal of Engineering Research & Technology (IJERT), **3**, *1*, 182-187, 2014.

METODA DE APROXIMARE VOGEL (CVAM) ADAPTATĂ PENTRU REZOLVAREA PROBLEMELOR DE TRANSPORT

(Rezumat)

Costurile de transport pot ajuta o companie să mențină costurile de producție la un nivel scăzut și evident să realizeze profit. Un manager trebuie să înțeleagă efectele costurilor de transport asupra producției. În această lucrare, propunem o variantă modificată a metodei de aproximare a lui Vogel (CVAM) pentru a găsi o soluție inițială fezabilă (IBFS) cu scopul de a minimiza problemele de transport. Concluzia desprinsă din această lucrare este aceea că metoda adaptată este mai simplă din punct de vedere computațional și furnizează soluții mai bune.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 1-2, 2016
Secţia
AUTOMATICĂ şi CALCULATOARE

LEAP MOTION CONTROLLED CAR

BY

ALEXANDRU-GABRIEL TUDORACHE[*] and FLORINA UNGUREANU

“Gheorghe Asachi” Technical University of Iaşi, Romania,
Faculty of Automatic Control and Computer Engineering

Received: April 14, 2016
Accepted for publication: May 16, 2016

Abstract. This paper describes the implementation of a Smart Car Architecture, using a Leap Motion Device Controller, both from the hardware perspective and the software one. The main components used are the Raspberry Pi Model B+, FRDM-KL25Z, two DC Gear engines (N20), two Bluetooth Modules (master HC-05 and slave HC-06) and the L298N Dual H-Bridge Motor Controller. The software relies on WebSocket technology – the client uses this communication in the JavaScript code (embedded in the web page the server offers), the server itself is a Python program, and the connection to the C++ code running on the FRDM (connected to the motor driver) is done using the Bluetooth modules. The position of the hand (X and Z inclinations) controls the engines’ speed and therefore the movement of the car.

Keywords: Raspberry Pi; FRDM; Smart Car; WebSocket; L298N Motor Controller.

2010 Mathematics Subject Classification: 74M05, 68M11.

1. Introduction

The described project fits the world of Internet of Things, as it uses multiple development platforms, various communication techniques, giving the user control over the movement of a Smart Car (its engines’ rpm).

[*] Corresponding author; *e-mail*: alexandru.tudorache93@gmail.com

One main customer grade product used in the project is a Leap Motion Controller, a device used to monitor the movement of the hands and fingers and other objects with similar shape. The Leap Motion controller (Fig. 1) uses optical sensors and infrared light. The sensors are directed along the y-axis – upward when the controller is in its standard operating position – and have a field of view of about 150 degrees. The effective range of the Leap Motion Controller extends from approximately 25 to 600 millimeters above the device (1 inch to 2 feet).

Fig. 1 – The Leap Motion controller’s view of your hands (Leap Motion Developer API Overview, 2015).

The system architecture diagram is presented in Fig. 2:

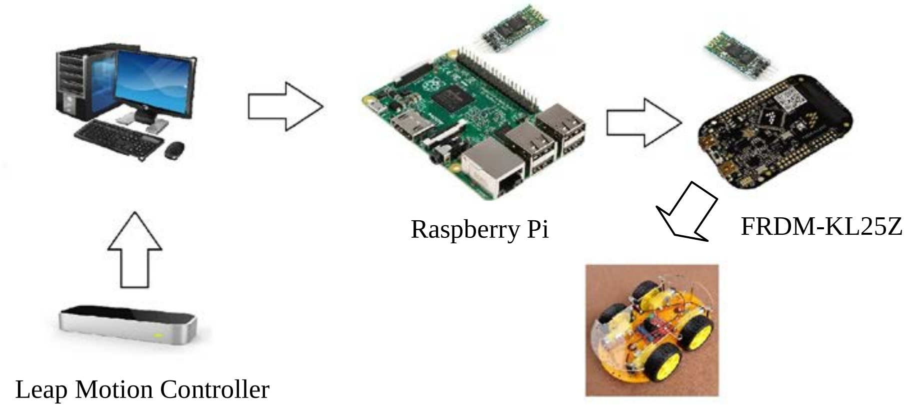


Fig. 2 – The system architecture.

The Leap Motion Controller is connected to a PC/laptop (playing the role of the WebSocket client) using an USB cable; the web server which runs on the Raspberry Pi (located in the same local network and running the WebSocket server) offers the client a web page which can be accessed by using the IP address of the Raspberry and the port of the WebSocket connection and typing them in a web browser; the data from the Leap device is processed by calling certain functions belonging to the JavaScript API, offering information such as hand inclination on the 3 axis, number of fingers, along with other data regarding gestures and movement (Leap Motion Developer Frames, 2015). Any of these pieces of information can be used for the car control - the X and Z hand inclination have been chosen for this project; they are sent through the WebSocket, after they are encoded in a JSON object with 2 fields, {“x_coords”, “z_coords”} – the inclinations on the 2 axis; the instructions are written in JavaScript and belong to the web page (the code written in this programming language runs on the client device). The JSON object is only sent is there is at least a 5-degree hand modification in the inclination on one of the selected axis. Once the object gets to the server, it is unpacked, and depending on the received data, a value is sent using the Bluetooth module – one byte is enough for 4 speed gears and 1 bit for the direction, for each engine; the FRDM receives the byte and sends the signals to the L298N Motor Controller, which controls the 2 engines.

Another view of the device, showing the placement of the infrared cameras is presented in Fig. 3.

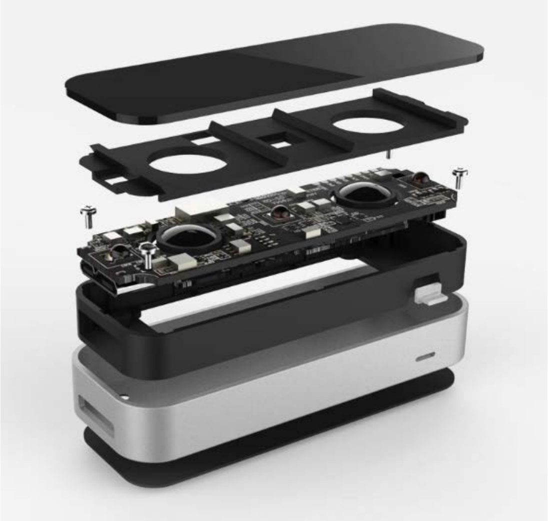


Fig. 3 – Expanded view of the Leap Motion Controller (Leap Motion Developer API Overview, 2015).

2. Software (API) and Hardware Presentation

The Leap Motion API (in JavaScript) and the hardware part of the project are presented in the first part of this section. The second part reveals the way all the components have been programmed to communicate and transmit the hand movement to the engines (JavaScript, 2015).

The reason for which the JavaScript language has been chosen is the necessity of a program, that although included on the server side, in the web page (written in Python, using the Tornado framework, running on the Raspberry Pi), can run on the client side, where the Leap Motion device is connected. The project can be extended so that the server can record (act as a logging system) and, if required, modify the command range to which the client has access to, changing neither the client configuration, nor the code on the FRDM, that gives the commands to the engines' driver.

JavaScript is a high-level, dynamic, untyped and interpreted programming language. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

The Leap Motion JavaScript API offers tracking data as a series of snapshots called frames. Each frame contains the measured positions and other information on each entity detected in the current snapshot.

The classes that are part of the Leap Motion API are: Bone, InteractionBox, Controller, Pointable, Finger, Matrix math, Frame, Vector math and Hand. Their purposes are briefly described as follows:

• Frame – the Frame class is the core of the data model, giving access to all the tracked entities. A new frame is created at each update interval. It contains the lists of hands and fingers tracked in the current frame (Data about fingers can also be obtained from a Hand object).

• Hand – the Hand object describes the position and the orientation of the hand, tracks its movement between frames and contains the list of corresponding fingers.

• Hand.arm – describes the position, direction and orientation of the arm of the corresponding hand. Data about arms can only be accessed using the Hand object.

• Pointable, Finger – Pointable objects define basic finger attributes. The Finger class extends the Pointable class by adding specific details to the tracked entities.

• Bone – the Bone object contains the position and orientation of a bone. The tracking of the bones includes the metacarpals and phalanges of the fingers.

• Image – the Image objects provide the raw sensor data and calibration grid for the Leap Motion cameras.

The class diagram used in the hand tracking model is represented in Fig. 4.

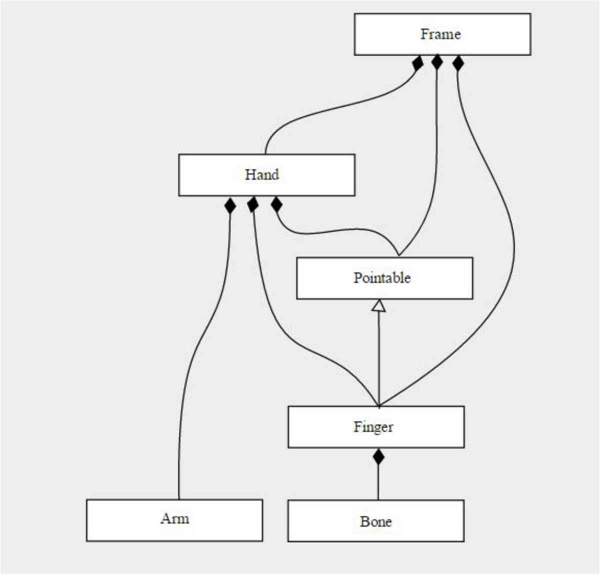


Fig. 4 – Classes in the hand tracking model (*Leap Motion Tracking Model*, 2015).

Each Frame object contains a snapshot of the recorded environment from the Leap Motion. The WebSocket Leap Motion server sends frames to the application as they are obtained; these messages are received by the JavaScript API (in the leap.js file) and are available from the Controller object as Frame objects (WebSocket Protocol, 2015). There are two ways of connecting the controller and acquiring information from frames, the simplest being the use of the *Leap.loop()* function, used in the present project, but also by creating a new object that extends Controller.

The *Leap.loop()* function creates an infinite loop that updates every frame, creates a Controller object and connects to the WebSocket server. After each update interval, the callback function is called, with the most actual frame as parameter.

```
      var controller = Leap.loop({enableGestures:true},
function(frame){
            var currentFrame = frame;
            var previousFrame = controller.frame(1);
            var tenFramesBack = controller.frame(10);
      });
```

The default setting of the *Leap.loop()* function are:

```
{enableGestures: false, frameEventName: "animationFrame"}.
```

The hardware components selected for the project are:

• Raspberry Pi Model B+ (Fig. 5), for its capacity to run a WebSocket server (using the Python programming language, on a Linux distribution – Raspbian Jessie) and the possibility to configure it so that it belongs to the same local area network with the client to which the Leap Motion is connected;

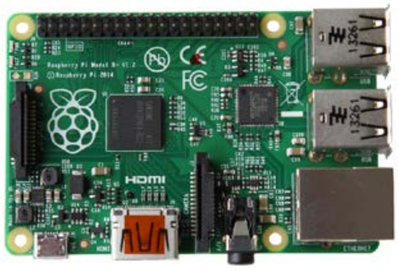


Fig. 5 – Raspberry Pi Model B+ (Rasberrypi, 2015).

• FRDM-KL25Z development platform (Fig. 6), chosen for its connectivity with other devices such as Bluetooth modules and motor drivers, and the ease of writing the code in the C++ language, with countless libraries for virtually every device, inside the online programming environment of mbed IoT Device Platform; a 9V battery is used to power the FRDM (ARMmbed, 2015);

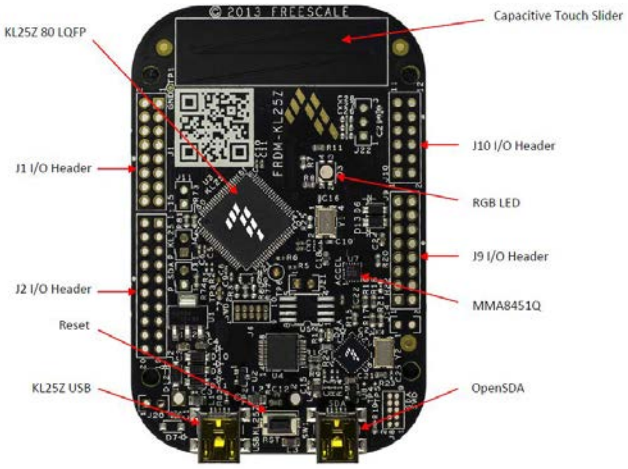


Fig. 6 – FRDM-KL25Z.

The two DC engines (Fig. 7), along with a Smart Robot Car chassis, for their affordable price and possibility to be controlled with the help of a L298N Motor Controller (Fig. 8), allowing the rotation of the wheels in both directions; four 1.5V batteries are used to power the L298N;

Fig. 7 – N20 DC Engine.

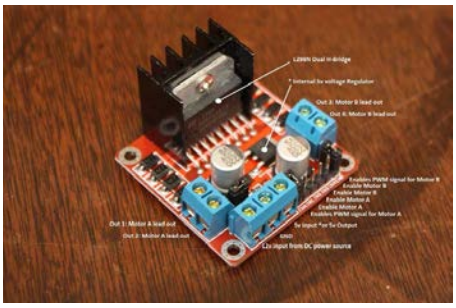


Fig. 8 – L298N Motor Controller.

• The Bluetooth modules (master HC-05 and slave HC-06 presented in Fig. 9), for their ability to interact with the Raspberry Pi and FRDM, assuring the communication of speed and direction, data that can be compressed in a single byte.

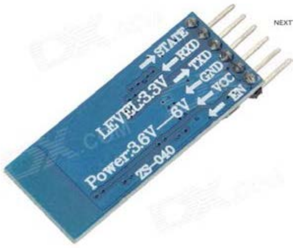


Fig. 9 – Bluetooth module HC-05.

3. Implementation

The Leap Motion controller is connected via USB to a computer that acts as a client and sends the hand inclination for the engines’ rpm control on the X axis – movement forward/backward and on the Z axis (rotating the car) to the Raspberry Pi, which acts as a server. The selected communication protocol is the WebSocket, that assures a full-duplex communication over a TCP connection; the HTTP servers interpret the handshake as an upgrade request. It has 2 parts: the handshake and the data transfer.

The handshake from the client has the following format:

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

The handshake from the server has the following format:

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: chat
```

A web server (using Tornado – a Python web framework and asynchronous networking library) and a WebSocket server run on the Raspberry (Tornado Web Server, 2015). In this way the client has access to a web page through which he will send the hand coordinates, packed in JSON objects with 2 fields {“x_coords”, “z_coords”}. Once it gets to the server, the data will be sent using the Bluetooth master module HC-05, connected to the Raspberry, to the FRDM development platform, that also has a Bluetooth module – HC-06 slave.

Key elements in the web page

The web page served to the client was developed starting from the sample one on the Leap Motion Developer website, to which visual modifications and the WebSocket client code (in JavaScript) have been added.

The webpage WebSocket server is accessed by typing into a browser the IP address of the Raspberry and the port (2000 for this project); the computer and Raspberry are in the same local network. The IP address of the server is read from the url and then used in the WebSocket constructor:

```
var addr = location.href;
addr = addr.substr(7); // remove http://
var webSocket = new WebSocket("ws://" + addr + "ws");
// for example, a client can type in his browser:
// 192.168.1.5:2000
```

The specific WebSocket methods for the most important events have been implemented – *onopen()*, *onmessage()* and *onclose()*:

```
webSocket.onopen = function() {
      alert("OK [WebSocket] Connected...");
};

webSocket.onmessage = function (evt) {
      var received_msg = JSON.parse(evt.data);
```

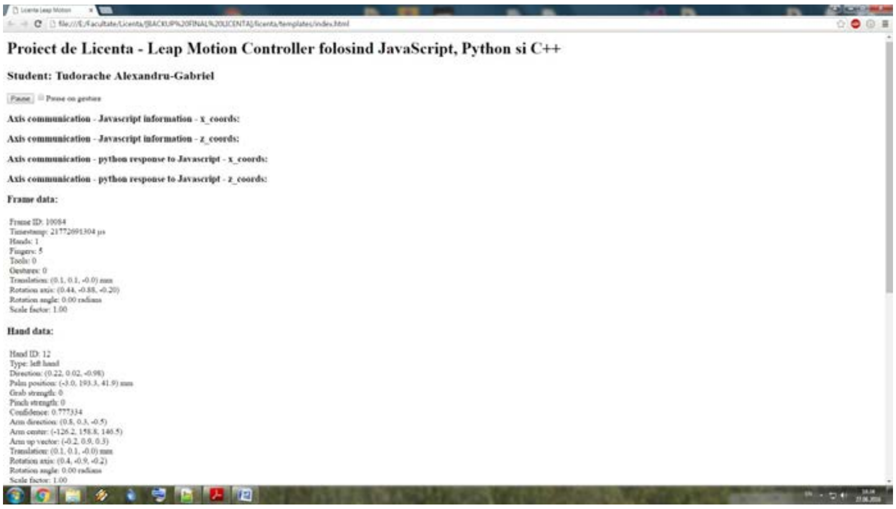
```
            document.getElementById("python_response_
x_coords").innerHTML = received_msg.x_coords;
            document.getElementById("python_response
_z_coords").innerHTML = received_msg.z_coords;
      };

      webSocket.onclose = function() {
            alert("Connection is closed...");
      };
```

A function named *poll_connection_and_send(rotationAxis)* has been created and is called every valid frame, using the information from the rotation axis when a change of at least 5 degrees occurs. This is the code that takes the X and Z inclinations and sends them as a JSON object:

```
      var msg = new Object();
      msg.x_coords = rotationAxis[0].toFixed(2);
      msg.z_coords = rotationAxis[2].toFixed(2);
      webSocket.send(JSON.stringify(msg));
```

A print screen of the web page can be seen in the Fig. 10.


Licenta Leap Motion
file:///E:/Facultate/Licenta/[BACKUP%20FINAL%20LICENTA]/licenta/templates/index.html
Pause
□ Pause on gesture

Proiect de Licenta - Leap Motion Controller folosind JavaScript, Python si C++

Student: Tudorache Alexandru-Gabriel

Axis communication - Javascript information - x_coords:

Axis communication - Javascript information - z_coords:

Axis communication - python response to Javascript - x_coords:

Axis communication - python response to Javascript - z_coords:

Frame data:

Frame ID: 10084
Timestamp: 21772691304 µs
Hands: 1
Fingers: 5
Tools: 0
Gestures: 0
Translation: (0.1, 0.1, -0.0) mm
Rotation axis: (0.44, -0.88, -0.20)
Rotation angle: 0.00 radians
Scale factor: 1.00

Hand data:

Hand ID: 12
Type: left hand
Direction: (0.22, 0.02, -0.98)
Palm position: (-3.0, 193.3, 41.9) mm
Grab strength: 0
Pinch strength: 0
Confidence: 0.777334
Arm direction: (0.8, 0.3, -0.5)
Arm center: (-126.2, 158.8, 140.5)
Arm up vector: (-0.2, 0.9, 0.3)
Translation: (0.1, 0.1, -0.0) mm
Rotation axis: (0.4, -0.9, -0.2)
Rotation angle: 0.00 radians
Scale factor: 1.00

Fig. 10 – Print screen of the web page.

The server (written in Python) loads the web page:

```
      self.render("templates/index.html");
```

and decodes the message:

```
      parsed_json = json.loads(message);
```

The two coordinates have rational values in [-1, 1] and are multiplied by 100 in order to only work with integers.

Once the server receives the values from the client, the code divides the values in 4 speed gears forward and 4 backward (25%, 50%, 75% and 100% of the maximum speed, using PWM – Pulse Width Modulation). Changing the car’s direction is done by moving the engines in opposite directions.

The byte that is sent has the following meaning:

$$\underbrace{b_7b_6b_5b_4}\,\underbrace{b_3b_2b_1b_0}$$

Data for the right and left engines

- b_7, b_3 – right / left engine direction: 0 – forward, 1 – backward;
- $b_6b_5b_4$, $b_2b_1b_0$ – the speed of the right / left engine as follows:
    - 000 – engine stopped;
    - 001 – 25% of top speed (1st gear);
    - 010 – 50% of top speed (2nd gear);
    - 011 – 75% of top speed (3rd gear);
    - 100 – 100% of top speed (4th gear).

The following conventions are made to convert the hand inclinations from [-100, 100] to the values described above:

- for the X axis:
    - [-30, 30]: deadzone / engine stopped;
    - 1st gear forward: [30, 45];
    - 2nd gear forward: [45, 60];
    - 3rd gear forward: [60, 75];
    - 4th gear forward: [75, 100];
    - Similar for the other direction (negative values).
- for the Z axis (only applied when in deadzone on X):
    - [-50, 50]: deadzone;
    - A value lower than -50 – rotation to the left (left gear backward, right gear forward);
    - A value higher than 50 – rotation to the right (left gear forward, right gear backward).

The FRDM runs a code written in C++ that receives the byte using the Bluetooth module, then finds the direction and speed with the help of the following bitwise operations:

```
left_engine_direction  = (received_int >> 3) & 1;
right_engine_direction = (received_int >> 7) & 1;
left_engine_speed      = received_int & 0x07;
right_engine_speed     = (received_int >> 4) & 0x07;
```

The value is written on the corresponding PwmOut pins:

```
engine1_1 = 0.25f * right_engine_speed;
engine1_2 = 0;
// for the other direction:
engine1_1 = 0;
engine1_2 = 0.25f * right_engine_speed;
```

The same goes for the other engine. The car is presented Fig. 11.

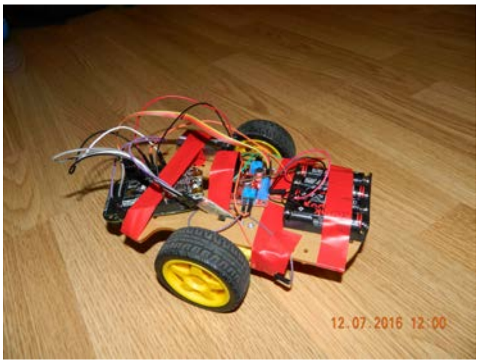


Fig. 11 – The Smart Car.

4. Conclusion

The project described in this paper shows the way in which multiple components of the Internet of Things can be connected to control a Smart Robot Car. Multiple development platforms and devices, like Raspberry Pi, FRDM-KL25Z and Leap Motion controller have been set up using different programming languages such as JavaScript, Python and C++.

Acknowledgments. This work is supported by the Romanian National Authority for Scientific Research (UEFISCDI), Project 1/2014 Virtual Therapist with Augmented Feedback for Neuromotor Recovery (TRAVEE).

REFERENCES

ARMmbed, https://developer.mbed.org/platforms/KL25Z/, 2015.

JavaScript, https://en.wikipedia.org/wiki/JavaScript, 2015.

Leap Motion Developer – API Overview, https://developer.leapmotion.com/documentation/javascript/devguide/Leap_Overview.html?proglang=javascript, 2015.

Leap Motion Developer Frames, https://developer.leapmotion.com/documentation/javascript/devguide/Leap_Frames.html, 2015.

Leap Motion Tracking Model, https://developer.leapmotion.com/documentation/javascript/devguide/Leap_Tracking.html, 2015.

Rasberrypi, https://www.raspberrypi.org/products/model-b-plus/, 2015.

WebSocket Protocol, https://tools.ietf.org/html/rfc6455, 2015.

Tornado Web Server, http://www.tornadoweb.org/en/stable/, 2015.

CONTROLUL UNEI MAŞINI CU AJUTORUL DISPOZITIVULUI LEAP MOTION

(Rezumat)

Această lucrare descrie implementarea unei arhitecturi Smart Car, folosind un dispozitiv Leap Motion, atât din perspectivă hardware, cât și software. Componentele principale folosite sunt Raspberry Pi Model B+, FRDM-KL25Z, două motoare de curent continuu N20, două module Bluetooth (master HC-05 și slave HC-06) și driver-ul de motor L298N. Componenta software are la bază protocolul de comunicație WebSocket – clientul folosește acest tip de comunicație în codul JavaScript (încorporat în pagina web deservită de server), server-ul este scris în Python, iar conexiunea dintre acesta și codul în C++ ce rulează pe FRDM (conectat la driver-ul de motor) este realizată prin intermediul modulelor Bluetooth. Poziția mâinii (înclinațiile pe axele X și Z) determină viteza motoarelor și astfel mișcarea mașinii.

BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 1-2, 2016
Secţia
AUTOMATICĂ şi CALCULATOARE

EMBEDDED RBF NEURAL NETWORK USING GRADIENT-BASED TRAINING WITH MULTIPLE ADAPTIVE LEARNING RATES

BY

LAVINIA FERARIU[*] and GEORGE GALAI

“Gheorghe Asachi” Technical University of Iaşi, Romania,
Faculty of Automatic Control and Computer Engineering

Received: May 20, 2016
Accepted for publication: June 29, 2015

Abstract. This paper discusses the development of a neuronal embedded system meant for solving function approximation and identification problems. The inductive learning capabilities of neural networks recommend their utilisation in a variety of applications. However, this formalism is rarely adopted in embedded systems – which are generally limited to reduced computational resources. This paper analyses how the neural techniques can be accommodated to the embedded architectures. The suggested approach considers a neural topology with radial basis functions (RBF). Given the local nature of hidden neurons and the linearity of output neurons, RBF brings some important advantages, *e.g.* the possibility of obtaining a low memory print, as well as the availability of fast constructive design algorithms. Necessary re-training of the model is performed without changing the initially selected neural topology. To this end, a gradient-based procedure with variable learning rates is used. Every epoch, the learning rate values are tuned via a fractional experiment. The approach can be easily ported on embedded architectures with limited hardware resources, as demonstrated for a FreeRTOS multitasking application developed for dsPIC33.

Keywords: neural networks; radial basis functions; function approximation; fractional experiments; gradient-based method.

2010 Mathematics Subject Classification: 49M30, 62K15, 62M45, 68N99, 68T99.

[*] Corresponding author; *e-mail*: lferaru@ac.tuiasi.ro

1. Introduction

Neural networks with **r**adial **b**asis **f**unction (RBF) are commonly used in a large variety of applications, such as data classification and nonlinear function approximations (Haykin, 2009). Mainly, the success of RBF results from its structural characteristics. Typically, an RBF admits a two-layer architecture including hidden neurons with radial activation functions (*e.g.* Gaussian) and linear output neurons. This structure exemplifies a functional decomposition providing universal approximation capabilities w.r.t. bounded continuous functions. Furthermore, because each hidden neuron is activated in a restraint area around its centre, the behaviour of the model could be more intuitively interpreted by analysing the locations of input patterns relative to the locations of centres. This is helpful for preventing over fitting and also for improving the configuration of the hidden layer during the training process. After the hidden layer is configured, the parameters of the output neurons can be simply computed by solving a set of algebraic linear equations.

One of the most popular learning approaches is the constructive algorithm which iteratively inserts the necessary hidden neurons - each new neuron correcting the worst error produced by a training sample (Haykin, 2009; Han *et al.*, 2011). Later improvements of the neural model can accept adding and deleting some hidden neurons. The elimination is generally approved for the hidden neurons having the least contribution; after elimination, the centres closer to the centres of the deleted neurons are slightly modified, in order to ensure that the activation areas of the remained neurons cover the whole input neural space (Han *et al.*, 2011).

The above mentioned characteristics recommend the use of RBF formalism in the embedded applications, too. However, according to our knowledge, no such application has been reported. Generally, very few embedded systems involving soft computing techniques have been developed and most of them considered **f**ield - **p**rogrammable **g**ate **a**rray architectures (FPGA) with extended computational capabilities (Baturone *et al.*, 2014; Eddahech *et al.*, 2013). Unlike these researches, this paper aims at finding solutions for a wider spectrum of embedded architectures, including the microcontroller-based ones. The goal is very challenging, especially because these embedded applications should ensure the compatibility with hard real-time constraints and predictability, while being shaped for reduced computational resources (Ibrahim, 2014; Laplante and Ovaska, 2012). The suggested approaches consist in several versions of the inertial back propagation algorithm with adaptive learning parameters, applied for an iterative improvement of the neural model.

It is worth mentioning that the constructive algorithm can support a scalable design of the RBF architecture within the context of embedded systems. The insertion of each new neuron ensures a monotonic improvement of the resulted squared output error w.r.t. the training data set; consequently, this permits balancing between accuracy and complexity in better accordance with the available computational resources. Although the hidden layer can be reconfigured by using only some recent training samples, the calculation of the output neural parameters requests accessing a large relevant data set including also the training samples previously learned. This could involve an unreasonable large computational load of the embedded applications, whenever only small modifications of the neural model are needed.

This explains our preference for the gradient-based learning. Generally, when applied from scratch, the back propagation algorithm involves higher computational effort than the constructive learning. However, as argued later, when only minor changes of the neural model should be handled, the gradient-based method permits working with fewer training samples, at reduced computational costs. Also, for the above mentioned working scenarios, this learning algorithm can allow a more flexible synchronisation with the real time acquisitions.

The procedures adopted for the adaptation of the learning parameters aim at reducing the risks of instability and slow convergence, commonly induced by the gradient-based techniques. In order to provide improved time performances, the adaptation is carried out during fractional experiments designed according to Taguchi's method. This paper discusses the computational performances for different adaptations schemes and exemplifies their development for FreeRTOS – a very popular open source, real time operating system (Barry, 2007). However, as long as the suggested embedded implementation considers only basic multitasking facilities, the presented solution can be also ported toot her real-time operating systems.

This paper is organized as follows. Section 2 describes the Taguchi's method used for the **d**esign **o**f the fractional **e**xperiments (DoE), while Section 3 presents how DoE is integrated into the gradient-based training algorithm, in order to allow the adaptation of the learning parameters. The design of the embedded application is discussed in Section 4 and some preliminary experimental results are analysed in Section 5. Last section summarises the conclusions.

2. The Method of Designing Experiments

DoE provides an effective design of partial factorial experiments related to multivariate problems involving factors with finite set of levels. One considers a quality function, F, depending on the independent factorsf_i ($i = \overline{1,a}$), each one with v distinct allowed levels, $l_{1,i},..l_{v,i} \in R$: $F: \{l_{1,1},..l_{v,1}\} \times..\times \{l_{1,a},..l_{v,a}\} \to R$, $F = F(f_1, f_2, \dots, f_a)$. Generally, the goal of DoE is to determine through few, yet relevant experiments what combination of factor levels is needed for the best value of F. The reduced set of trials can be also used for other refined statistical analyses (*e.g.* the **an**alysis **of** **va**riance - ANOVA). The fractional experiment is useful especially in problems with numerous factors and/or numerous levels per factor, for which the full factorial set of v^a trials requests unreasonable computational resources.

Table 1
L$_4$ *Orthogonal Array:* (*Layout* 2^3)

Experiment	Factors		
	f_1	f_2	f_3
#P.1	$l_{1,1}$	$l_{1,2}$	$l_{1,3}$
#P.2	$l_{1,1}$	$l_{2,2}$	$l_{2,3}$
#P.3	$l_{2,1}$	$l_{1,2}$	$l_{2,3}$
#P.4	$l_{2,1}$	$l_{2,2}$	$l_{1,3}$

Table 2
The Factorial Analysis – the Full Experiment (*Layout* 2^3)

Experiment	Factors			F
	f_1	f_2	f_3	
#F.1	2 ($l_{1,1}$)	10 ($l_{1,2}$)	100 ($l_{1,3}$)	112
#F.2	2 ($l_{1,1}$)	10 ($l_{1,2}$)	150 ($l_{2,3}$)	158
#F.3	2 ($l_{1,1}$)	20 ($l_{2,2}$)	100 ($l_{1,3}$)	122
#F.4	2 ($l_{1,1}$)	20 ($l_{2,2}$)	150 ($l_{2,3}$)	172
#F.5	4 ($l_{2,1}$)	10 ($l_{1,2}$)	100 ($l_{1,3}$)	114
#F.6	4 ($l_{2,1}$)	10 ($l_{1,2}$)	150 ($l_{2,3}$)	164
#F.7	4 ($l_{2,1}$)	20 ($l_{2,2}$)	100 ($l_{1,3}$)	124
#F.8	4 ($l_{2,1}$)	20 ($l_{2,2}$)	150 ($l_{2,3}$)	170
$\bar{F}_{\mid[l_{1,i}]}$, $i = 1,2,3$	**141** $[l_{1,1}]$(in #F.1,#F.2,#F.3,#F.4)	**137** $[l_{1,2}]$(in #F.1,#F.2,#F.5,#F.6)	**118** $[l_{1,3}]$(in #F.1,#F.3,#F.5,#F.7)	
$\bar{F}_{\mid[l_{2,i}]}$ $i = 1,2,3$	143 $[l_{2,1}]$(in #F.5,#F.6,#F.7,#F.8)	147 $[l_{2,2}]$(in #F.3,#F.4,#F.7,#F.8)	166 $[l_{2,3}]$(in #F.2,#F.4,#F.6,#F.8)	

Most fractional approaches are derived from Fisher's technique, for factors with separable effects. An example is the Taguchi method which offers precomputed orthogonal arrays, directly indicating the subset of experiments (Taguchi *et al.*, 2005). An orthogonal array corresponds to a specific layout, v^a, such that a column is associated to a factor and a row describes a trial. Within this context, the orthogonality means that the levels of a factor are evenly used and, for any pair of factors, the combinations of their levels are also evenly used. Table 1 exemplifies the array (**L**$_4$) built with only 4 trials for the layout 2^3. For instance, the experiment #P.1 sets all the factors on their first level. The full corresponding experiment (with 8 trials) is shown in Table 2 – here, the trials marked with red are those selected for the partial experiment. Obviously, $\mathbf{L}_4$ follows the two balancing rules indicated above.

Table 3

The Factorial Analysis – the Partial Experiment Designed with $\mathbf{L}_4$ (*Layout* 2^3)

Experiment	Factors			F
	f_1	f_2	f_3	
#P.1 (#E.1)	2 ($l_{1,1}$)	10 ($l_{1,2}$)	100 ($l_{1,3}$)	112
#P.2 (#E.4)	2 ($l_{1,1}$)	20 ($l_{2,2}$)	150 ($l_{2,3}$)	172
#P.3 (#E.6)	4 ($l_{2,1}$)	10 ($l_{1,2}$)	150 ($l_{2,3}$)	164
#P.4 (#E.7)	4 ($l_{2,1}$)	20 ($l_{2,2}$)	100 ($l_{1,3}$)	124
$\bar{F}_{\|[l_{1,i}]}$ $i = 1,2,3$	**142** $[l_{1,1}]$ (in #P.1,#P.2)	**138** $[l_{1,1}]$ (in #P.1, #P.3)	**118** $[l_{1,3}]$ (in #P.1, #P.4)	
$\bar{F}_{\|[l_{2,i}]}$ $i = 1,2,3$	144 $[l_{2,1}]$ (in #P.3,#P.4)	148 $[l_{2,2}]$ (in #P.2, #P.4)	168 $[l_{2,3}]$ (in #P.2, #P.3)	

Table 4

The $\mathbf{L}_{16}$ *Array Showing the Fractional Experiment Used for Training* (*Layout* 4^5)

Experiment	$f_1 = \mu_w$	$f_2 = \mu_c$	$f_3 = \mu_\sigma$	$f_4 = \mu_b$	$f_5 = \mu_m$
#P.1	$l_{1,1}$	$l_{1,2}$	$l_{1,3}$	$l_{1,4}$	$l_{1,5}$
#P.2	$l_{1,1}$	$l_{2,2}$	$l_{2,3}$	$l_{2,4}$	$l_{2,5}$
#P.3	$l_{1,1}$	$l_{3,2}$	$l_{3,3}$	$l_{3,4}$	$l_{3,5}$
#P.4	$l_{1,1}$	$l_{4,2}$	$l_{4,3}$	$l_{4,4}$	$l_{4,5}$
#P.5	$l_{2,1}$	$l_{1,2}$	$l_{2,3}$	$l_{3,4}$	$l_{4,5}$
#P.6	$l_{2,1}$	$l_{2,2}$	$l_{1,3}$	$l_{4,4}$	$l_{3,5}$
#P.7	$l_{2,1}$	$l_{3,2}$	$l_{4,3}$	$l_{1,4}$	$l_{2,5}$
#P.8	$l_{2,1}$	$l_{4,2}$	$l_{3,3}$	$l_{2,4}$	$l_{1,5}$
#P.9	$l_{3,1}$	$l_{1,2}$	$l_{3,3}$	$l_{4,4}$	$l_{2,5}$
#P.10	$l_{3,1}$	$l_{2,2}$	$l_{4,3}$	$l_{3,4}$	$l_{1,5}$
#P.11	$l_{3,1}$	$l_{3,2}$	$l_{1,3}$	$l_{2,4}$	$l_{4,5}$
#P.12	$l_{3,1}$	$l_{4,2}$	$l_{2,3}$	$l_{1,4}$	$l_{3,5}$
#P.13	$l_{4,1}$	$l_{1,2}$	$l_{4,3}$	$l_{2,4}$	$l_{3,5}$
#P.14	$l_{4,1}$	$l_{2,2}$	$l_{3,3}$	$l_{1,4}$	$l_{4,5}$
#P.15	$l_{4,1}$	$l_{3,2}$	$l_{2,3}$	$l_{4,4}$	$l_{1,5}$
#P.16	$l_{4,1}$	$l_{4,2}$	$l_{1,3}$	$l_{3,4}$	$l_{2,5}$

The list provided by Taguchi includes the orthogonal arrays necessary for the most frequent layouts (up to 31 two-level factors, 13 three level factors and 9 four-level factors). The smallest possible array compliant with the problem should be taken from this list and its unnecessary columns should be ignored. Generally, a partial factorial experiment modifies the key measures used by the factorial analysis. An example is given in Tables 2 and 3. Here, the best configuration of levels is set with respect to the minimization of F. For each independent factor, the decision is based on the average quality value, $\overline{F}$, resulted for the subset of trials involving only one level of the factor. According to Tables 2 and 3, the partial testing leads to slightly different averages, but eventually both sets of experiments recommend the same configuration, *i.e.* $l_{1,1}$, $l_{1,2}$ and $l_{1,3}$. Unfortunately, the relevance of the partial factorial subset cannot be guaranteed for any F, regardless of the interactions between the factors. This limitation can be easily exemplified by changing the result of #F.2 from 158 to 180, as after this modification, the full experiment selects $l_{2,1}$ instead of $l_{1,1}$.

This paper employs the Taguchi method for the dynamic configuration of the momentum constant and learning rates used in the inertial back propagation algorithm. As explained later, the approach involves a maximal layout with 5 four-level factors, corresponding to a full experiment of 1024 trials. This layout is compatible with the $\mathbf{L}_{16}$ orthogonal array (Table 4) which allows a significant reduction of the full experiment. Details regarding the significance of the factors and the way in which the Taguchi method is integrated into the learning procedure are given in Section 3.

3. The Back Propagation Algorithm with Fractional Experiments

The embedded neural system is developed according to the RBF topology, which implicitly offers increased scalability. The goal is to support an effective re-training of the neural network, such that to correct the errors without ignoring the knowledge already stored in the neural system. To fit this end, the current neural parameters are iteratively improved via the inertial back propagation algorithm, while keeping the neural topology unchanged. Small environmental changes are expected to demand only few epochs of training, thus making the evolution of the neural model easier to manage in real-time applications. Furthermore, the approach employs refined policies for configuring the parameters of the training procedure, as explained below.

One considers a standard RBF structure having N inputs, a hidden layer with K Gaussian neurons and an output layer with P linear neurons. The architecture is fully connected. The neural parameters which should be modified during learning are the centers ($\boldsymbol{c}_k = [c_{kn}]$) and the dispersions ($\sigma_k$) of the hidden neurons, as well as the weights (w_{pk}) and the biases (b_p) of the output neurons - with $n = \overline{1, N}$, $k = \overline{1, K}$, $p = \overline{1, P}$. The available training data set, S, contains M relevant patterns, each one specifying the input vector ($\boldsymbol{x}_m = [x_{mn}]$)) and the corresponding targeted output vector ($\boldsymbol{t}_m = [t_{mp}]$) - with $m = \overline{1, M}, n = \overline{1, N},\ p = \overline{1, P}$. During run-time training, this data set stores the most recent M samples acquired from the environment; additionally, in the case of identification problems, the input patterns are reshaped according to the adopted serial – parallel scheme of external delays.

As mentioned before, the inertial back propagation algorithm is applied for the iterative adaptation of the neural parameters. The batch algorithm aim sat minimizing the mean squared output error corresponding to *S*:

$$J = \sum_{m=1}^{M} \sum_{p=1}^{P} (t_{mp} - y_{mp})^2, \tag{1}$$

where $\boldsymbol{y}_m = [y_{mp}]$ indicates the neural output vector obtained for the input pattern $\boldsymbol{x}_m$.The common adaption rule demands the appropriate configuration of two parameters, namely the learning rate and the momentum constant:

$$z(i+1) = z(i) + \mu_z \Delta z_J(i) + \mu_\alpha \Delta z(i-1), \tag{2}$$

$$\Delta z_J(i) = -\frac{\partial J}{\partial z}(i), \Delta z(i-1) = z(\mathrm{i}) - z(i-1), \tag{3}$$

where i specifies the iteration, z denotes a generic parameter, μ_z and μ_α are the learning rate and the momentum constant, respectively. The learning rate influences the learning speed and the stability of the algorithm, as well as its ability to escape from the neighborhood of local optima. Additionally, the momentum constant sets the confidence in the previously operated changes, which are used in (2) for softening the successive variations produced along opposite directions. Obviously, the adaptation of these parameters could be advantageous for accommodating their values to the demands of the training process.

Furthermore, in this layer-based neural architecture, the local gradients of the hidden layer are expected to be significantly smaller than those of the output layer. Therefore, if a single learning rate is employed for all the neural units, the updating rule should accommodate neurons with very different learning potential. Moreover, the learning capacity of a neural unit depends on the number of its incoming connections, those with more input links being exposed to larger behavioral variations during a single iteration. In a full connected neural network, all the neurons of a layer have the same number of input connections.

With this in mind, distinct learning rates are employed for each category of parameters, *i.e.* μ_w and μ_b for the weights and the biases of the output layer, as well as μ_c and μ_σ for the centers and the spread of hidden neurons, respectively. Also, these learning rates, together with the momentum constant are adaptively set, by extending the idea presented by Lin *et al.* (2010). More precisely, these learning parameters are considered factors influencing the values of the quality function *J*. Assuming that each factor accepts *v* potential levels ($v \in \{2, 3, 4\}$), a partial experiment is carried out in order to select the best configuration of the learning parameters recommended for the current iteration.

The levels of the factors are distributed, as follows: $\{\mu_z(i-1), \mu_z(i-1)+q\}$, for $v=2$; $\{\mu_z(i-1)-q, \mu_z(i-1), \mu_z(i-1)+q\}$, for $v=3$; $\{\mu_z(i-1)-q, \mu_z(i-1), \mu_z(i-1)+q, \mu_z(i-1)+2q\}$, for $v=4$. Here, $q=b\cdot\mu_z(i)$ with $b\in(-1,1)$, and the resulted levels are saturated between 0 and 2. By allowing both negative and positive values of q, the algorithm can flexibly balance the number of trials increasing/decreasing the learning parameters, when $v=2$ or $v=4$. According to Taguchi's method, the orthogonal arrays associated to the resulted v^5 layouts are, correspondingly: $\mathbf{L}_8$ without its two last columns (for $v=2$), $\mathbf{L}_{18}$ without the first and the last two columns (for $v=3$) and $\mathbf{L}_{16}$ (for $v=4$). The last configuration is displayed in Table 4. Being the most time consuming, the case $v=3$ is the least preferred. Although, any $v\in\{2,3,4\}$ anticipates reasonable time performances even for the rudimentary embedded architectures.

Aiming at improving the time performances, the fractional experiment is optionally allowed to work on a reduced RBF architecture ignoring the less significant hidden neurons. This option is motivated by the local character of the Gaussian neurons; for a specific training sample, only few hidden neurons are activated, namely those having the center close to the input vector. Solely these active hidden neurons could trigger significant changes for the output weights; therefore, the hidden neurons having all their output weights with very small variations are invalidated in the reduced neural structure. For a sequential learning, the reduced RBF could include very few hidden neurons without significantly altering the accuracy of the factorial analysis. However, if the sequential learning starts with random initial centers and spreads, the reduction is not operational during the first iterations. Also, this advantage cannot be provided for batch training, if each hidden neuron is activated by at least one training sample.

The summary of the learning algorithm is presented in Fig. 1. Based on this outline, other faster simplified version scan be derived. For instance, one can consider that the training procedure is solely applied for the weights and the biases of the output layer. This can still offer good adaptation capabilities, if the Gaussian bells describing the hidden layer are well distributed throughout the neural input space. In this case, the factorial analysis involves fewer (three) factors, thus leading to partial fractional experiments defined according to $\mathbf{L}_4$ (for $v=2$), $\mathbf{L}_9$ without the last column (for $v=3$) and $\mathbf{L}_{16}$ without the last two columns (for $v=4$). The same layout of fractional experiments can be also used if the back propagation algorithm works on all the neural parameters, but μ_w=μ_b and μ_α is kept constant.

Table 5 characterizes the computational complexity of the algorithm required for a single iteration. The sequential version could be derived for $M=1$. Here, $\beta^*\leq\beta$. Smaller values of $\beta^*$are anticipated for the sequential learning and also for the batch learning using few training samples describing only a subset of potential behaviors. The embedded implementation should take into account that the algorithm is time consuming whenever a large number of epochs are required, *i.e.* for the initial training and for any training providing significant changes of the model; for these cases the classic constructive learning algorithm could to be faster.

```
Update the training data set $S$ according to recently measured samples.
For each iteration $i$:
    If sequential training is applied, fill $S^*$ with a single sample, randomly selected from $S$.
    Otherwise use $S^* = S$.
    For each training sample, $m$, available in $S^*$:
        Compute the response of the hidden layer, the output of the RBF and the squared
        error corresponding to this sample, $J_m = \sum_{p=1}^{P}(t_{mp} - y_{mp})^2$.
        For each parameter $z$, compute the variation $\partial J_m/\partial z$ , according to the
        backpropagation algorithm.
    If batch training is applied,
        For each parameter, compute the average variation resulted on $S^*$.
    If the reduced RBF structure is designed:
        Fill the set $W^*$ with the weights having variations higher than $\Delta$. If $W^*$ is empty,
        put in $W^*$ the weight with the biggest absolute value of the variation.
        Find all the $K^*$ hidden neurons having at least one output link characterized by a
        weight from $W^*$.
        Build a reduced RBF* with the $K^*$ hidden neurons found at the previous step; keep
        only their input/output connections.
    For each trial of the fractional experiment:
        Generate the temporary values of $\mu_w$, $\mu_b$, $\mu_c$, $\mu_\sigma$, $\mu_\alpha$.
        Apply (2) for computing the new value of each parameter included in reduced
        RBF. Use whole RBF if reduced RBF is not available.
        Compute the quality value according to (1), w.r.t the reduced RBF. Use whole
        RBF if reduced RBF is not available
    Select the best configuration for $\mu_w$, $\mu_b$, $\mu_c$, $\mu_\sigma$, $\mu_\alpha$.
    Apply (2) w.r.t. the whole RBF and store the resulted neural parameters.
```

Fig. 1 – The training algorithm with fractional experiment.

However, the constructive algorithm is less flexible w.r.t. the synchronization with the real-time acquisitions. Firstly, a learning procedure which adds or deletes some hidden neurons requests also updating the parameters of the output layer after the configuration of the hidden layer is decided. Usually, this is done by solving an algebraic system of linear equations (or equivalently), *e.g.* by means of QR decomposition (according to the computational complexity $O(M(K+1)^2)$). This calculation cannot be stopped in an intermediate stage; it must be fully run in order to subsequently use the resulted weights. It is worth mentioning that the configuration of the hidden layer can be solved according to $O(MNK + MKP)$. Secondly, because the structure of the hidden layer is reconfigured from scratch with respect to the available training data set, the classic constructive algorithm cannot be applied in sequential version. The training data set should store huge number of relevant samples, including also the relevant samples learned at previous trainings, even if only minor variations of the neural model are needed. Obviously, this translates to the need of using numerous algebraic equations.

On the contrary, the back propagation training accepts versions with different computational complexity, making its integration more flexible for various real-time applications. With the risk of reduced accuracy, this training process can be stopped after any iteration, whenever higher-priority processes become ready for execution. Also, because the parameters are not computed from scratch, small training data sets (including $M = 1$) can be also considered. It is worth mentioning that during the re-training process the risk of instable sequential learning is quite reduced. However, this risk is supplementary diminished by the adaptive run-time configuration of the learning parameters.

Table 5

The Computational Complexity of the Inertial Back Propagation Training – per Iteration

Algorithm configuration	With $RBF*$	Without the fractional experiment	The fractional experiment		
			$v = 2$	$v = 3$	$v = 4$
Improve all the neural parameters: $\beta = P(K+1) + K(N+1)$, $\beta^* = P(K^*+1) + K^*(N+1)$.	no	$O(M\beta)$	$O(8M\beta)$	$O(18M\beta)$	$O(16M\beta)$
	Yes		$O(8M\beta^*)$	$O(18M\beta^*)$	$O(16M\beta^*)$
Improve only the parameters of the output layer: $\beta = P(K+1)$, $\beta^* = P(K^*+1)$.	No	$O(M\beta)$	$O(4M\beta)$	$O(9M\beta)$	$O(16M\beta)$
	Yes		$O(4M\beta^*)$	$O(9M\beta^*)$	$O(16M\beta^*)$

4. The Design of the Embedded Application

The development of the embedded real time application is exemplified for FreeRTOS (Barry, 2007) and dsPIC33 (Bates, 2011). As long as only common multitasking facilities and computational resources were assumed, the solution can be simply ported to other real-time operating systems and hardware architectures.

Before going into the details of real-time application design, the basic necessary characteristics of FreeRTOS are reviewed. FreeRTOS accepts ISRs and preemptive/ non-preemptive tasks. The tasks are implemented as infinite loops and accept four different states: ready for execution, running, blocked and suspended. The transitions towards the suspended state are triggered by `vTaskSuspend` called from any process. Also, any process can force a suspended task to become ready, via `vTaskResume`. The blocking state corresponds to the tasks which are waiting for a specific time interval or which are waiting to receive the reading/writing access to a queue of messages which is currently unavailable (*i.e.* an empty queue accessed for reading or a full queue accessed for writing). FreeRTOS allows the online deletion/creation of tasks, as well the online modification of their priorities. The dynamic memory allocations are supported by `heap_3` memory scheme. The scheduler is clock-driven, with static priorities. Whenever called, it assigns the processor to the highest priority task which is not suspended or blocked; the tasks with equal priority are managed according to round robin policy. In preemptive mode, the implicit rescheduling occurs periodically, at the beginning of each frame. Rescheduling is also demanded by different API services in preemptive and non-preemptive modes. FreeRTOS offers also facilities for communicating through queues of messages, as well as for sharing passive resources via semaphores implemented with or without priority inference (Barry, 2007).

The execution of FreeRTOS application could be monitored via several available hooks and trace functions. However, debugging the embedded applications is generally time consuming and difficult (Ibrahim, 2014; Laplante and Ovaska, 2012). This is why the development process was organized in three main phases. The first stage was focused on analyzing the functionalities of the algorithm and the influence of its parameters, in comparison with the classic constructive training procedure. This step was solved in MATLAB, which offers extensive facilities for matrix-based computations, as well as libraries for neural network design and graphical visualizations. Then, the application was ported to C, by considering a friendly integrated development environment which can support fast compilations (Visual Studio). Aiming at providing a good control on the memory consumption and execution times, the automatic translation from MATLAB to C was not preferred. Although, by using the MATLAB implementation as reference, the C code was very easy to debug. The C implementation was limited to the libraries available for the embedded application. Lastly, the C functions built for the design of the neural system were integrated in the real-time multi-tasking application developed for FreeRTOS.

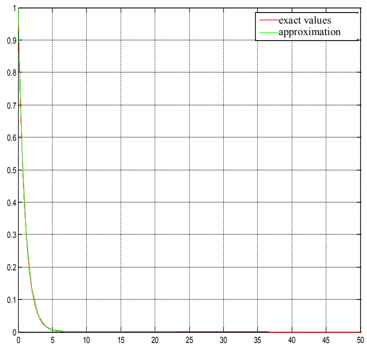


Fig. 2 – The approximation of the exponential function (red/light grey) based on a piecewise linearization (green/darker grey).

C application development considered the key requirements of the embedded systems related to memory and time resources. Therefore, in order to avoid memory segmentation, the preference was for static allocations. Only few dynamic memory reservations were accepted, applied for the initialization of the neural network, in a controlled, non-preemptive sequence, with the only goal of easily verifying the availability of the necessary amount of memory. Furthermore, useful data structures were prototyped for storing the training data set, the neural parameters, the responses of neurons, the variations triggered by the training procedure for each neural parameter, as well as the configurations of fractional experiments. These structures allowed a simple reference-wise management of data via reduced stack sizes. All necessary large data blocks were accepted with far allocations. Also, in order to obtain a small memory print, the inclusion of mathematical libraries was avoided. To this end, the exponential function $f(x) = \exp(-x)$ needed for the hidden layer was implemented by considering an accurate piecewise linearization with 11 vertices, as shown in Fig. 2.

```
TASK_INIT{
while(1){
    Configure PIT1, ATD1 and COM1
    (without enabled interrupts);
    Initialize the data structures required
    for monitoring and training;
    If the neural parameters are read via
    COM1 from the PC:
        Enable COM1 interrupt, read all
        the parameters and disable
        COM1 interrupt;
        Set the number of epochs used
        in regular trainings;
        Create TASK_TRAIN and
        TASK_MONITOR;
        Create queue Q1 with depth 1
        (shared later between ISR_AD
        and TASK_MONITOR);
        Enable ATD1 interrupt ;
    Otherwise
        Enables ATD1 interrupt until
        ISR_AD fills the initial training
        data set;
        Disable ATD1;
        Set the number of epochs for the
        initial training;
        Create TASK_TRAIN;
Suspend itself.}}

ISR_AD{
    Update the index of the training sample;
    Read data from ATD1 and write the new
    training sample;
    If Q1 available,
        Write in Q1 with infinite timeout.}

TASK_TRAIN{
while(1){
    Disable ATD1 interrupt;
    Loop for the available number of
    epochs{
        Modify the parameters according to
        the adopted version of inertial back
        propagation algorithm;
    If TASK_MONITOR is not created:
        Set the number of epochs for
        regular trainings;
        Create TASK_MONITOR;
        Create queue Q1 with depth 1;
    Enable ATD1 interrupt;
    Suspend itself.}}

TASK_MONITOR{
while(1){
    Read Q1 with infinite timeout;
    Monitor the error ;
    If training is necessary
        Resume TASK_TRAIN.}}
```

Fig. 3 – The configuration of the embedded application.

The real-time embedded application includes: an **i**nterrupt **s**ervice **r**outine (ISR) collecting the results of the AD convertor (ISR_AD), an ISR for sending/receiving data via a serial interface to/from a PC, a task performing the necessary initial configurations (TASK_INIT), a task for training (TASK_TRAIN) and a task for monitoring the error of the RBF model (TASK_MONITOR). This architecture can be extended with any other ISRs and tasks according to available processing time and time constraints, *e.g.* tasks for diagnosis and control. For the dsPIC33 architecture, the acquisitions are set to be periodically triggered by the timer PIT1; an equivalent configuration can be provided for any microcontroller, with I/O devices commonly available on chip.

The functionality is briefly described in Fig. 3. As explained in Section 3, the main limitations of the back propagation procedure are related to the cases when long trainings (with numerous iterations) are required. About this, the embedded application performs the first training without allowing interruptions and preemptions. Also, the application accepts to directly parse an available set of initial neural parameters via COM1. These options are controlled by TASK_INIT. Additionally, TASK_INIT makes all necessary initial hardware configurations. Once the neural model is configured, the monitoring is started. Every sampling period, ISR_AD updates a new sample in the training data and then TASK_MONITOR evaluates the error corresponding to the newly acquired sample. If the output squared error exceeds the preset threshold for several consecutive sampling periods, TASK_MONITOR triggers the re-training. The synchronization between ISR_AD and TASK_MONITOR is solved via a queue with depth 1, such that TASK_MONITOR is permitted to evaluate the model only after the execution of ISR_AD is finished; to this end, the queue is accessed for writing by ISR_AD and for reading by TASK_MONITOR, with infinite accepted timeout. The ATD1 interrupt could remain enabled during training, only if the training process ends before the next sampling time. This functionality is preferred, but it requests previous experimental verifications performed for the adopted algorithm parameters and embedded architecture. TASK_TRAIN suspends itself after training and it is resumed by TASK_MONITOR whenever necessary. For a predictable execution, TASK_TRAIN has higher priority than TASK_MONITOR and lower than TASK_INIT.

5. Experimental Results

Firstly, the experimental investigations analysed the accuracy performances of the batch inertial back propagation algorithms involving different fractional experiments. The testing scenario uses two nonlinear function approximations, as indicated in Table 6. The results were compared with those provided by the classic constructive training algorithm implemented in MATLAB by `newbr`. The constructive algorithm considered the same number of hidden neurons as the back propagation learning. For all these experiments, the gradient-based training algorithm is applied from scratch, starting with random initial parameters. Although this is not the standard case targeted by the real-time re-training, this configuration was preferred during testing because it can generate more diverse behaviours of the learning algorithm. In order to allow a fair comparison between different versions of the gradient-based training corresponding to the same testing set, the total number of epochs, N_{ep}, was set such that to ensure a quite similar computational load of the fractional experiments, regardless of the number of trials they involve.

The configurations listed in Table 6 aims at illustrating the role of the factorial experiment applied for different number of factors, subject to the same set of neural parameters (namely the parameters of the hidden and output layers), without any reduction of the RBF. Therefore, the following configurations are used: (C1) a factorial experiment with 16 trials updating 5 four-level factors, *i.e.* μ_w,μ_b, $\mu_c\mu_\sigma$ and μ_α; (C2) a factorial experiment with4 trials, updating 3 two-level factors, *i.e.* μ_w=μ_b, μ_c andμ_σ; (C3) no factorial experiment, *i.e.* the learning parameters are kept constant. The training was applied for $M = 16$ uniformly distributed input samples, while the validation data set considered 1000 samples. Given the complexity of the approximated functions, working with $K = 4$ was enough for achieving accurate approximations. In all the testing cases the learning parameters were initialised with 0.4 and modified later, according to $b = \pm 0.3$. The results obtained for the validation data set are listed in Table 7 and plotted in Fig. 4. All the training algorithms were able to provide accurate approximations; for enough large number of epochs, the gradient-based algorithm can offer improved accuracy. The results presented in Table 7 indicate that the adaptation of more learning parameters if beneficial for the accuracy of the neural model, w.r.t to the validation data set (#1 vs. #2 and #2 vs. #3 in T1, T2 and T3). Moreover, these preliminary experimental results indicate that (C1) can favour the achievement of a better accuracy even after intermediary epochs (T3 vs. T1).

If the re-training needs to accommodate minor changes of the neural model, adequate accuracy performances are expected to be obtained after very few iterations. These circumstances are investigated via the second testing scenario. The algorithm continues the testing case#T1.1, by re-training the resulted RBF w.r.t. a variation of 20% of the targeted output, as indicated by the new training sample $[\frac{\pi}{2}, 0.8 * sin^2\left(\frac{\pi}{2}\right)$. In only 3 epochs, the output squared error results insignificant: 1.618 →0.005→1.9e-06. This fast re-training was supported by substantial variations of the adapted learning parameters, from $\mu_b = \mu_\sigma = \mu_c = \mu_\alpha = 0.272, \mu_w = 0.582$ to $\mu_b = \mu_\alpha = 0.184$, $\mu_w = 0.865$, $\mu_\sigma = \mu_c == 0.446$. During these iterations, a single hidden neuron is activated and only the weight corresponding to its output connection has significant variations (-0.139, -0.00118, -0.00017), all the other output weights being changed with less than 10^{-6}. This is a good motivation for building a reduced structure RBF* including only this hidden neuron (*e.g*. according to $\Delta \approx 10^{-4}$). In this case, the resulted RBF* allows performing a faster fractional experiment, without disturbing the conclusions of the factorial analysis.

Table 6

The Experimental Configurations Included in the Testing Scenario

Test set	Test case	Test configuration		N_{ep}
T1	#1	$y: [0, 2\pi] \to Ry(u) = \sin^2 u$	C1	400
	#2	$y: [0, 2\pi] \to Ry(u) = \sin^2 u$	C2	1600
	#3	$y: [0, 2\pi] \to Ry(u) = \sin^2 u$	C3	6400
T2	#1	$y: [0, 1] \to R, y(u) = 1 + e^{-u}\sin(2\pi u - \pi/2).$	C1	400
	#2	$y: [0, 1] \to R, y(u) = 1 + e^{-u}\sin(2\pi u - \pi/2).$	C2	1600
	#3	$y: [0, 1] \to R, y(u) = 1 + e^{-u}\sin(2\pi u - \pi/2).$	C3	6400
T3	#1	$y: [0, 1] \to R, y(u) = 1 + e^{-u}\sin(2\pi u - \pi/2).$	C1	100
	#2	$y: [0, 1] \to R, y(u) = 1 + e^{-u}\sin(2\pi u - \pi/2).$	C2	400
	#3	$y: [0, 1] \to R, y(u) = 1 + e^{-u}\sin(2\pi u - \pi/2).$	C3	1600

Table 7

The Experimental Results – the Mean Squared Errors for the Validation Data Set

Test set	Constructive algorithm	Gradient-based learning		
		#1	#2	#3
T1	0.00117	0.00112	0.00131	0.00153
T2	0.00066	0.00014	0.00102	0.00159
T3	0.00066	0.00030	0.00075	0.00092

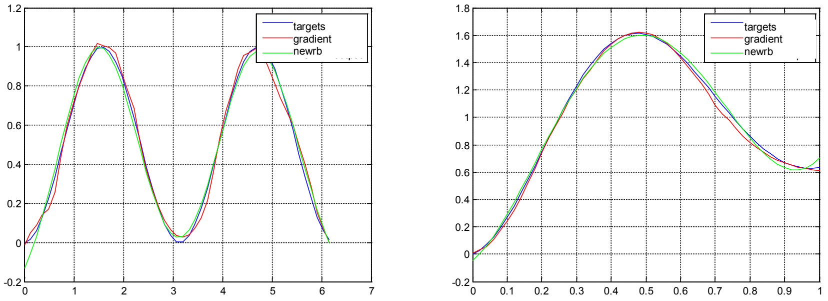


Fig. 4 – The results obtained on the validation data set: left figure for #T1.1, right figure for#T3.1.

Lastly, the algorithm was tested on the embedded architecture, in order to illustrate the memory consumption and the execution times for different working configurations. For #T1.1 working with maximum 51 training samples, the approach needed only 12312B (4% of the available amount). The time performances were monitored via `traceTASK_SWITCHED_OUT` and `traceTASK_SWITCHED_OUT` for frames equal to 1msec; in the case of (C1), the training time resulted smaller than 1msec per epoch, without reducing the RBF's structure during the fractional experiments. Given the fact that dsPIC33 is only a cheap microcontroller with less than average performances, these results show an appropriate scalability of the suggested embedded solution.

6. Conclusions

The paper presents an embedded application developed for RBF neural networks. The training algorithm considers an inertial back propagation scheme with multiple adaptive learning rates. In order to allow a faster adaptation of learning parameters, the factorial analysis is solved for a fractional experiment designed according to Taguchi's orthogonal arrays.

The paper discusses different versions of the training algorithms and their impact on the performances of the embedded application. The main targeted working scenario is the one when the neural model needs run-time retraining. Unlike the constructive algorithms, this learning algorithm is able to improve the accuracy of the model without starting from scratch, using also few training samples. This leads to convenient memory consumption and small execution times, whenever minor changes of the neural model should be learned.

Future research will take into account additional experimental verification of the real-time application illustrating also the use of this neural embedded system in different control structures.

REFERENCES

Barry R., *FreeRTOS - User Manual*, 2007.

Bates M., *PIC Microcontrollers* (3rd Edition), Elsevier, 2011.

Baturone I., Gersnoviez A., Barriga A., *Neuro-Fuzzy Techniques to Optimize an FPGA Embedded Controller for Robot Navigation*, Applied Soft Computing, **21**, 95-106, 2014.

Eddahech A., Chtourou S., Chtourou M., *Hierarchical Neural Networks Based Prediction and Control of Dynamic Reconfiguration for Multilevel Embedded Systems*, Journal of Systems Architecture, **59**, *1*, 48-59, 2013.

Ibrahim D., *Designing Embedded Systems with 32-Bit PIC Microcontrollers and Mikro C*, Elsevier, 193-258, 2014.

Haykin S., *Neural Networks and Learning Machines*, 3rd Edition, Prentice Hall, USA, 2009.

Han H.G., Chen Q.L., Qiao J.F., *An Efficient Self-Organizing RBF Neural Network for Water Quality Prediction*, Neural Networks, **24**, 717-725, 2011.

Laplante P., Ovaska S., *Real-Time Systems Design and Analysis*, 4th Edition, IEEE Press, 2012.

Lin W.M., Gowa H.J., Tsai M.T., *An Enhanced Radial Basis Function Network for Short-Term Electricity Price Forecasting*, Applied Energy, **87**, 3226-3234, 2010.

Taguchi G., Chowdhury S., Wu Y., *Taguchi's Quality Engineering Handbook*, John Wiley and Sons, 2005.

SISTEM ÎNCORPORAT NEURONAL DE TIP RBF FOLOSIND ALGORITM DE ANTRENARE BAZAT PE GRADIENT CU MUTLIPLE RATE DE ÎNVĂŢARE ADAPTIVE

(Rezumat)

Lucrarea analizează dezvoltarea unui sistem neuronal încorporat dedicat problemelor de aproximare de funcţii şi identificare. Capacitatea reţelelor neuronale artificiale de învăţare inductivă le recomandă a fi utilizate într-o largă varietate de aplicaţii. Totuşi, formalismul nu este frecvent folosit în sisteme încorporate, care oferă în general doar resurse de calcul limitate. Această lucrare analizează modul în care tehnicile neuronale pot fi utilizate pe arhitecturi încorporate. Abordarea propusă consideră reţele neuronale cu funcţii de bază radiale (RBF). Dat fiind caracterul local al neuronilor ascunşi şi liniaritatea stratului de ieşire, RBF aduce câteva avantaje importante, de exemplu consumul scăzut de memorie şi posibilitatea de antrenare rapidă folosind un algoritm constructiv. Soluţia propusă consideră că re-antrenarea modelului se va efectua fără modificarea structurii neuronale, folosind un algoritm de tip gradient cu moment, cu rate adaptive de învăţare. La fiecare epocă, ratele de învăţare şi constanta moment sunt modificate pe baza unui experiment parţial. Abordarea propusă este posibil de utilizat pe arhitecturi încorporate cu resurse hardware limitate, aşa cum este ilustrat pentru aplicaţia dezvoltată sub FreeRTOS pentru dsPIC33.